ТРУДЫ

ИНСТИТУТА СИСТЕМНОГО ПРОГРАММИРОВАНИЯ РАН

PROCEEDINGS OF THE INSTITUTE FOR SYSTEM PROGRAMMING OF THE RAS

ISSN Print 2079-8156
Том 35 Выпуск 1

ISSN Online 2220-6426
Volume 35 Issue 1

Институт системного
программирования
им. В.П. Иванникова РАН

Москва, 2023

ИСП РАН

ISSN 2079-8156 (Print), ISSN 2220-6426 (Online)

Труды Института системного программирования РАН
Proceedings of the Institute for System Programming of the RAS

Труды ИСП РАН – это издание с двойной анонимной системой рецензирования, публикующее научные статьи, относящиеся ко всем областях системного программирования, технологий программирования и вычислительной техники. Целью издания является формирование научно-информационной среды в этих областях путем публикации высококачественных статей в открытом доступе.
Издание предназначено для исследователей, студентов и аспирантов, а также практиков. Оно охватывает широкий спектр тем, включая, в частности, следующие:

- операционные системы;
- компиляторные технологии;
- базы данных и информационные системы;
- параллельные и распределенные системы;
- автоматизированная разработка программ;
- верификация, валидация и тестирование;
- статический и динамический анализ;
- защита и обеспечение безопасности ПО;
- компьютерные алгоритмы;
- искусственный интеллект.

Журнал издается по одному тому в год, шесть выпусков в каждом томе.
Поддерживается открытый доступ к содержанию издания, обеспечивая доступность результатов исследований для общественности и поддерживая глобальный обмен знаниями.
Труды ИСП РАН реферируются и/или индексируются в:

Proceedings of ISP RAS are a double-blind peer-reviewed journal publishing scientific articles in the areas of system programming, software engineering, and computer science. The journal's goal is to develop a respected network of knowledge in the mentioned above areas by publishing high quality articles on open access.
The journal is intended for researchers, students, and practitioners. It covers a wide variety of topics including (but not limited to):

- Operating Systems.
- Compiler Technology.
- Databases and Information Systems.
- Parallel and Distributed Systems.
- Software Engineering.
- Software Modeling and Design Tools.
- Verification, Validation, and Testing.
- Static and Dynamic Analysis.
- Software Safety and Security.
- Computer Algorithms.
- Artificial Intelligence.

The journal is published one volume per year, six issues in each volume.
Open access to the journal content allows to provide public access to the research results and to support global exchange of knowledge. **Proceedings of ISP RAS** is abstracted and/or indexed in:

УДК004.45

Адрес: 109004, г. Москва, ул. А. Солженицына, дом 25.
Телефон: +7(495) 912-44-25
E-mail: info-isp@ispras.ru
Сайт: http://www.ispras.ru/proceedings/

Address: 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.
Tel: +7(495) 912-44-25
E-mail: info-isp@ispras.ru
Web: http://www.ispras.ru/en/proceedings

Содержание

Table of Contents

DOI: 10.15514/ISPRAS-2023-35(1)-1

Design of an adaptable dashboard for smart cities

V. Contreras-Figueroa, ORCID: 0000-0003-2650-9748 <zs20000684@uv.mx>
L.G. Montané-Jiménez, ORCID: 0000-0003-2732-5430 <lmontane@uv.mx>
M. Cepero-García, ORCID: 0000-0002-0255-4256 <marite_cepero@live.com.mx>
E. Benítez-Guerrero, ORCID: 0000-0002-5386-107X <edbenitez@uv.mx>
C. Mezura-Godoy, ORCID: 0000-0001-5844-4198 <cmezura@uv.mx>

University of Veracruz,
Xalapa, Veracruz, 91020, Mexico

Abstract. Today there are smart cities that, through the use of information technologies, sensors, and specialized infrastructure, focus their efforts on improving the quality of life of their inhabitants. From these efforts arose the need to analyze and represent data within a system to make it useful and understandable to people, for which dashboards emerge. The objective of these systems is to provide users with information to support decision-making, so it is essential to adapt the visualization of the information provided to their needs and preferences. However, the analysis of adaptability through user interaction and its benefits is a topic still under exploration. This paper analyzes the literature on information visualization in adaptable dashboards for smart cities. Based on the elements of adaptable dashboards identified in the literature review, we propose an adaptable dashboard architecture, identify the main characteristics of the users of a smart city dashboard, and build an adaptable dashboard prototype using user-centered techniques.

Keywords: Dashboard; Information visualization; Smart Cities; Adaptable system

For citation: Contreras-Figueroa V., Montané-Jiménez L.G., Cepero-García T., Benítez-Guerrero E., Mezura-Godoy C. Design of an adaptable dashboard for smart cities. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023, pp. 7-24. DOI: 10.15514/ISPRAS-2023-35(1)-1

Разработка адаптируемой информационной панели для умных городов

В. Контрерас-Фигероа, ORCID: 0000-0003-2650-9748 <zs20000684@uv.mx>
Л.Г. Монтане-Хименес, ORCID: 0000-0003-2732-5430 <lmontane@uv.mx>
М. Сеперо-Гарсия, ORCID: 0000-0002-0255-4256 <marite_cepero@live.com.mx>
Э. Бенитес-Герреро, ORCID: 0000-0002-5386-107X <edbenitez@uv.mx>
К. Мезура-Годой, ORCID: 0000-0001-5844-4198 <cmezura@uv.mx>

Университет Веракруса,
91020, Мексика, Веракрус, Халапа

Аннотация. Сегодня существуют умные города, в которых за счет использования информационных технологий, датчиков и специализированной инфраструктуры повышается качество жизни жителей. При этом возникла потребность в анализе и представлении данных в некоторой системе, чтобы сделать их полезными и понятными для людей, для чего применяются информационные панели. Целью этих систем является предоставление пользователям информации для поддержки принятия решений, поэтому важно адаптировать визуализацию предоставляемой информации к их потребностям и предпочтениям. Однако анализ возможностей и преимуществ адаптивности посредством взаимодействия с пользователями – это тема, находящаяся на стадии изучения. В данной статье анализируется литература по визуализации информации в адаптируемых информационных панелях для

умных городов. На основе элементов адаптируемых информационных панелей, выявленных в обзоре литературы, мы предлагаем архитектуру адаптируемой информационной панели, определяем основные характеристики пользователей информационной панели умного города и создаем прототип адаптируемой информационной панели с использованием методов, ориентированных на пользователей.

Ключевые слова: информационная панель; визуализация информации; умные города; адаптируемая система

Для цитирования: Контрерас-Фигероа В., Монтане-Хименес Л.Г., Сеперо-Гарсия М., Бенитес-Герреро Э., Мезура-Годой К. Разработка адаптируемой информационной панели для умных городов. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 7-24. DOI: 10.15514/ISPRAS–2023–35(1)–1

1. Introduction

Some cities are now called smart because of their ability to use information and communication technologies (ICT) to improve the quality of life of citizens and the overall operation of the city [1, 2]. The smart city approach incorporates ICT in almost all aspects of daily life in an urban space. Some areas that can take advantage of the integration of services and platforms are economy, government, environment, housing, society, and mobility, to mention a few [3].

A key smart city technology is the dashboard. According to Few [4], dashboards are a visual display of the most important information to support users to achieve one or more objectives in their daily lives.

Visualizations in dashboards show in a visual way – through charts and maps – the information of key performance indicators (KPIs). To be useful for decision-making, the visualizations of these KPIs must be carefully designed and then selected for deployment [5].

To meet the demand for information, it is important to consider the characteristics of the users. New forms of interaction can improve existing systems and create new platforms that contribute to the development of different areas of a city. Some recent research on the development of smart city-centric systems shows the importance of integrating the user into the development and deployment process of these systems [6-8].

In the current development of dashboards for smart cities there is a tendency to focus on a single user or a group with specific characteristics [9], which affects the deployment in cases where a user who is outside the specific context seeks to use the dashboard for their benefit. Some authors [10-12] have explored the different information needs of dashboard users and tried to approach the problem from a user-centric perspective, which has led to the development of different dashboards for different users.

Sharifiq [13], through an analysis and description of information visualization using a flexible dashboard, shows how users can create their own configurations focused on what they want to visualize. This approach is a first attempt to make the interface adaptable to the user.

This research analyzes adaptable dashboards in the context of smart cities, identifies city dashboards' users, and proposes an adaptable dashboard and its architecture. This paper is an extension of the paper originally presented at the 9th International Conference in Software Engineering Research and Innovation (CONISOFT 2021) [14]. The paper is organized as follows. Section 2 presents the methodology followed in carrying out the systematic review. Section 3 presents the quantitative and qualitative results of the review. Section 4 shows the proposed component specification as a result of the previous work. Section 5 shows the development of the interface and the user model. Section 6 shows the process carried out in evaluating the adaptable dashboard prototype. Finally, Section 7 discusses conclusions and future work.

2. Research process

We have conducted a systematic literature review following the methodology for systematic reviews developed by Kitchenham [15]. The research questions addressed by this study are:

Q1. How is key performance indicator information represented within the visual components of a dashboard?

Q2. What are the methodologies currently applied in the construction of dashboards?

Q3. What are the benefits of adaptability applied to dashboards?

Q4. What elements can be adapted within a smart city dashboard?

Based on the research questions and considering other terms obtained from previous research [16-18, 9], we identified the following key words that are consistent with what is proposed to be found: ”Smart city”, ”Adaptable dashboard”, ”Information visuali*ation”, and ”Key performance indicator”. Once the terms had been defined, they were used to create the search string (S1):

*S1. (”Smart city” OR ”adaptable dashboard”) AND (”Information visuali*ation” OR ”key performance indicator”).*

To carry out an orderly review and to find valuable results for the research questions, we used the following selection criteria with which the works will have to comply to be considered:

- Inclusion:
  - A research carried out in retrospective of no more than five years.
  - A research carried out in the area of computer science.
- Exclusion:
  - Technical and programmatic platform development work.
  - Papers published in a language other than English.

In order to search the resources to answer the research questions, we used the queries in four research databases: ACM Digital Library, IEEE Xplore, ScienceDirect, and SpringerLink. To extract the relevant papers for the systematic literature review, we followed a four-stage procedure:

1) Application of inclusion and exclusion criteria within search engines.
2) Synthesis of initial results and organization of metadata.
3) The initial selection process of articles through their title and keywords.
4) Selection process of papers by analysis of their abstract and contributions.

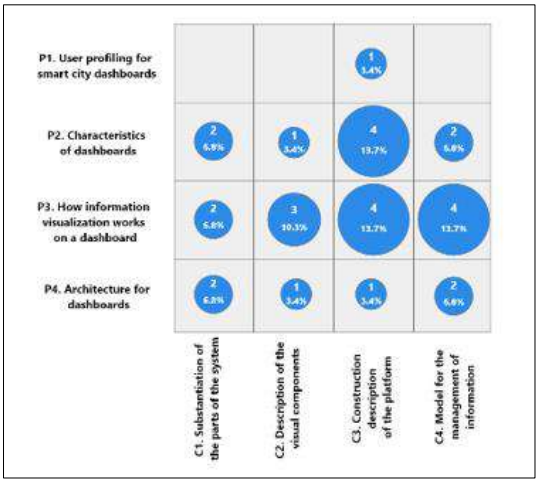

Fig. 1. Quantitative analysis of adaptability in dashboards

3. Review results

3.1 Quantitative analysis

The results of the search in the databases gave total of 542 articles, of which 518 were excluded, and 24 were selected for analysis. Based on the 24 selected articles, we performed a quantitative analysis of the problems and contributions identified in each paper through a bubble chart.

The quantitative analysis of the contributions on adaptability in dashboards for smart cities is presented in Fig. 1. In this analysis, we identified that the greatest concentration of work is on how information is visualized on a dashboard.

3.2 Qualitative analysis

Within the visualization field, dashboards are commonly applied to monitor what is happening in a specific context so that people can interpret the visualization results and relate them to particular goals [19].

Tong and Wu [20] summarize the six characteristics of city dashboards: recording, connectivity, sensing, interaction, adaptation, and integration. Recording refers to saving city data. Adaptation refers to the ability to customize data products and services based on needs.

Decision-makers can contrast the data through visual components (either static or real-time KPIs) to make comparisons and inferences to improve city operations [21] [22].

One of the biggest challenges of dashboards for smart cities is to satisfy the different information visualization needs of users to ensure that they are useful in their decision making [9].

In the current development of dashboards for smart cities, the aim is to provide the user with visual representations of data that are part of several screens [23].

To learn how to integrate adaptability into dashboards, we will analyze the literature in the following subsections. Each of the subsections corresponds to the answers to the research questions posed in Section 2.

3.2.1 Data visualization for city dashboard

City dashboards collect data from the urban environment for its analysis and display. Today there are multiple schemes (tools, frameworks, indices, indicators, and rankings) of urban data formed with a hierarchical structure of urban data analysis, where each level is described by the results of the previous level [24], [25].

Zdraveski [12] proposes a model with three scales of resolution or level of detail of the indicators: temporal (annual, quarterly, monthly, weekly, daily, hourly, and real-time values); spatial (values based on city, district, street, or GPS location); and human or population (values based on the city, region, municipality, neighborhood, household, or person).

The dashboards technology tries to solve the information overload for users by using visual components such as charts and tables to effectively communicate the city's current state and historical data to help identify patterns.

A recent survey [26] mentioned that it is vital for users of dashboards and smart city systems to manipulate the information for their benefit. According to this study, there are three main types of users: citizens, authorities, and communities.

A starting point to integrate the user in the information visualization process is to analyze how the user will obtain information from the system. There are several ways to represent and classify the information, Protopsaltis [19], and Peddoju [27] made a description of the most used charts in information visualization based on the analysis and pattern of data of interest. Their work concentrates on the main charts used when making decisions with a dashboard, as defined in Table 1, with univariate, bivariate, or multivariate variables. The charts are currently used in dashboards to represent multiple data and provide additional information that allows users to interpret the data.

Table. 1 Graphics used for the construction of dashboards

Data type	Name	Characteristics
Univariate	Columns and bars [27] [19]	Measurement of a variable according to a metric.
	Pie [27]	Illustrate the proportion of the elements that make up a whole.
	Area [27] [19]	To identify patterns between measurements of the variable and make comparisons.
Bivariate	Scatter plot [27]	Visualization of information in 2 and 3D for multidimensional analysis.
	Heat map [27] [19]	To show the spatial distribution of the variable on a map.
	Line [27] [19]	To explain functional dependencies between variables.
Multivariate	Circular area (radar) [19]	Comparison of multiple variables and their behavior among them
	Stacked bars [19]	Composition of data and categories that change over time.
	Bubbles [27] [19]	Specify large dimensions of variables within the same graphic.
	Timelines [19]	To understand the evolution of variables in relation to time.

Maps are another dashboard visualization component that presents geographic information through a digital representation of space, displaying location-related information [28, 29].

In addition to traditional charts and maps, specific visualization methods have been developed for certain scenarios. Purahoo [30] used a way of representing the information employing a gauge chart (speedometer-like chart) to represent the decibel level of the environmental sounds. Moustaka et al. [31] proposed a display to show the relationships between the various dimensions of a smart city through a model based on DNA structure. Another form in which the information can be represented is a 3D model [32].

The use of different ways of representing information allows developers to adapt these components according to what a user needs to achieve their goals. This representation also brings us to the challenge of building a dashboard with adaptable features that consider the user to select relevant information for their goals when viewing the dashboard and which tools it will display to promote a good experience.

3.2.2 Construction of dashboards

The construction of the scorecard of the city of Trieste in Italy developed by Brunetto [33], detailed analysis of its users and context. The dashboard was built considering the characteristics of the people who use it and the impact it would have within the government in which they work.

Habibzadeh et al. [34] showed the characteristics for the construction of a smart city system in detail. As a starting point, [34] considers seeing these systems as a set of five different planes: I) of application, II) detection, III) communication, IV) data, and V) security. The model, despite being robust, does not consider the user in the construction.

One way in which user characteristics are considered to improve a system is the process applied by [28]. By using a three-dimensional design of the city and through data analysis with the Internet of Things (IoT) data, these systems provide crucial information to smart city dashboard end-users.

Vicuna [29] applied a visualization process focused on the improvement of transportation within the city with transportation performance metrics (TPM), which consisted of applying steps for the analysis of information and the visualization process.

Rolim et al. [35] proposed an architecture used to build dashboards focused on how the user can comfortably interact with them. They use two architectures; the first is focused on constructing a dashboard for the visualization of information. The second is a dashboard for developers where they can modify and control the information on the visualization. Related to this, Chrysantina and Ivar

[36] in their work on user-focused dashboard design, mentioned some of the main issues to consider when developing a user-focused (or user-designed) dashboard.

An important contribution identified from the review of the literature was the Dashboard Design Guide developed by Few [16], in which he identified and described analogies to common problems in dashboards. Few also present guidelines for positioning and arranging elements to avoid overloading the user. The authors [36], [37] and [9], developed their research based on the characteristics of this guide.

Regarding the construction of adaptable dashboards, it was identified in the works of Ergasheva et al. [9], and Elshehaly et al. [37] similar steps in the process of building adaptable dashboards, so the following procedure is proposed for their construction, it can be seen in the Fig. 2.

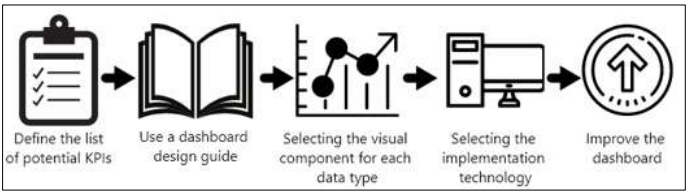

Fig. 2. Procedure for dashboard construction

A fundamental part of applying methodologies to the development of dashboards is the possibility of validation through testing. Dostal et al. [38] developed a method of evaluation for smart city environments through models based on current and future information.

The literature review commonly mentions the process of building dashboards based on their technology and the steps to achieve a final product that meets the characteristics of information visualization. However, this only provides applied processes and patterns that, even though they follow a standardization, are not studied with the end users.

3.2.3 Adaptability in dashboards

One feature of dashboard design that has been little explored and implemented is adaptability.

As mentioned in [9], the focus of current dashboards is a single person or a group of people who will be using the system. Making it clear the need to incorporate adaptability into the dashboard design to meet users' needs, and in turn, improve their decision-making based on the KPIs displayed on their screen. For example, Ergasheva et al. [9] built its InnoMetrics platform focused on energy efficiency, which adapts what is displayed on the screen according to the importance of an event that requires a solution.

The most crucial feature identified in the literature review is integrating user needs and preferences into the development of platforms related to information visualization, particularly dashboards for smart city. It must meet their expectations and objectives to be helpful as a tool to improve their quality of life and their interaction with their environment [39].

As mentioned by Han et al. [22], the creation of dashboards through user assistance tools to personalize the components which will display all the information requires a classification of the information to be subsequently displayed in a useful way to the user in its context. An efficient dashboard is one that considers the needs of users to ensure that the viewing process is completed smoothly [40].

Elshehaly et al. [37] developed a dashboard that actively involves the user during the whole process. Since the tool allows the user to adapt the information displayed within the cards, users feel comfortable visualizing what is necessary and can expand the information communicated through their interaction with the system.

The use of Nielsen heuristics [41] to develop a dashboard type system makes it possible to add features that make it attractive for users in their context of use. Within smart cities, understanding the needs of both the city and the users who are going to make decisions is crucial.

One of the primary benefits of adaptability in dashboard is to improve the quality of the information to make decisions in a smart city. Information visualization helps to provide the information and facilitates the work of local authorities and people interested in monitoring the city. As a result of using visualization to achieve this, it is possible to build better opportunities for citizens and visitors [42].

3.2.4 Adaptable dashboard elements

The first type of adaptability identified in the literature analyzed was the operation of a visualization system. An example of this type of adaptability is shown in the document of Chan et al. [43], where graphical elements display information from sensors in an organized manner, first using a data analysis algorithm that provides an open standard for users to create compelling visualizations.

The creation of dashboards through user assistance tools to customize the components that will display all the information requires a classification of the information to later display it in a useful way to the user in context [22]. Smart city dashboards function as a constant monitoring system that shows users information about their environment to help them make decisions [44].

According to Alves et al. [45], one way to adapt visual information is with zooming in and out on different sets of time-based details, which allows the user to specify the period in which they want to review the information.

In contrast, a particular way identified for adapting dashboards according to users and the context of smart cities is the use of Domain Specific Language (DSL), which focuses on using specific words and phrases with a syntax to make changes to the content displayed [46].

Silva et al. [47] identified that the number of KPIs used for smart city systems is relevant to modify the way in which information is displayed in a logical and orderly manner, thus improving the understanding of the information for the user. The organization of the KPIs information within a smart city system is an important feature that can be adaptable [47]. Silva et al. [47] proposed a structure that organizes the system elements involved in the information organization, from the document's structure. Following this same idea, Limon et al. [42], through an analysis based on the construction of smart cities, identified the critical parts when designing a platform focused on information visualization. This model seeks to improve the construction of web platforms by using a software engineering model [48]. Other factors identified previously in the literature as relevant components of an adaptable dashboard are: dashboard design guidelines [16] and the importance of integrating users and their information within the process [9] [35] [34]. Based on identified adaptable dashboard elements, we present the specification of components and an architectural proposal for an adaptable dashboard in the following section.

4. Structure of adaptable dashboard

Adaptable dashboards can be customized to present data according to the user knowledge to tailor what is being communicated. Therefore, adaptability characteristics in dashboards can enhance the user's data understanding and use of available information [18]. Based on the literature review and the model of [42], we identified elements that had not been previously considered such as design guidelines, user information for the construction of the dashboard, and a personalized selection of key performance indicators.

Smart city features are another component that is considered when designing adaptable dashboards. This information can be collected through sensors and databases. Subsequently, the selection of the KPIs used to display the information through visual components is made. We suggest building the visual components using the recommendations of the dashboard design guide developed by Few

[16]. And finally, to build the dashboard, we will use user preferences and interaction to adapt the dashboard interface.

We develop a functional architecture to show the internal structure of an adaptable dashboard (see Fig 3). Its parts will be described in the following sections to understand the adaptable dashboard proposal.

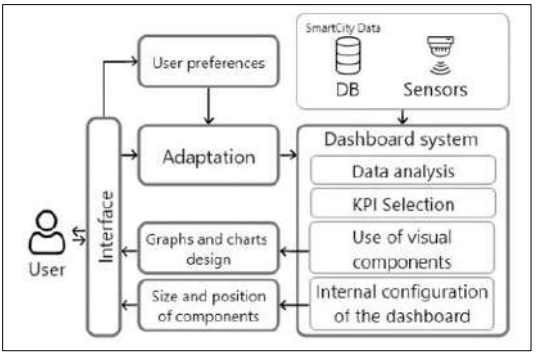

Fig. 3. Adaptable dashboard architecture

4.1 Users

The dashboard users can interact with the visualizations to select, filter, and adjust the components to change their shape and size. In this type of platform, it is necessary to consider the number of components and avoid information overload.

The number of components, the amount of information, and the way in which data are represented are elements that users unconsciously use to understand and use the dashboard visualizations [23].

Ivanov [49] mention the priorities of the users organized in the following way: performance monitoring, planning, communication, and consistency of data management.

The authors Ivanov [49] and Young [50] define three main types of users for the dashboards, 1) Operational, 2) Analysts or managers and 3) Experts or executives. The three types of users analyzed will be considered for the dashboard.

Each dashboard user must use the information with the minimum effort to make a quick decision; thus, this information must be eye-catching and interactive to be meaningful and of constant use [49].

4.2 Smart city data

The dashboard for smart city is constructed in a way that its internal elements can constantly operate through data from sensors and databases. Some existing public databases pertain to specific countries or cities; however, these data are only available to the citizens through files or application programming interfaces [51, 52]. For example, in Mexico, an information transparency portal has been in place since 2018 that provides data files that are constantly updated to collect and display information for systems that support the work of the government [53].

The source of the data is relevant because it defines the way in which it will be used in the construction of visual representations. Current web platforms use sockets to transmit information in real time, while some others use databases that are constantly updated from automated systems.

For the adaptable dashboard prototype, the Thingsboard [54] platform was selected, this platform consists of a web-based system focused on the control and deployment of dashboards for smart cities.

4.3 Adaptation

A system is called adaptable if it provides the user with tools that allow the user to change the characteristics of the system. Its objective is to provide the user with facilities to adapt the system to his personal tasks and needs. Control of the adaptation is given to the user, who must initiate the adaptation and use it [55].

Adaptable dashboards enable the selection of the most appropriate metrics and use them in a structured way. In this way, by adapting the interface to the user interaction, relevant information for each user is displayed when needed [56]. These adaptability features are implemented in a general dashboard interface to transform the way data is displayed and improve users' decision-making.

Familiarity can be generated in the user, reducing information overload, and allowing them to interact in a precise and easy way. Among the adaptable components that will be considered are: 1) Customization, 2) Zoom, 3) Colors, sizes and positions of the components, 4) Emphasize relevant information and 5) Make components simultaneously comparable [23].

Adaptability in dashboards enhances the display of information with an interactive mechanism that allows users to select the information to be displayed on their interface. In the dashboards already generated, adaptability allows the user to specify the information and the way it is presented within the graphical interface.

5. Adaptable dashboard development

As a starting point for designing adaptability within the dashboard, we considered the contributions of Strugar [56] and Young [50]. In their work, they mention that there is a set of key questions to consider in the design of a dashboard:

1) What metrics does the user need to visualize?
2) What context does each metric require to be meaningful?
3) Which visual representation best communicates the metric?

The answers to the guiding questions were defined to have a first draft of what the system and the organization of the information within the dashboard deployment will be.

The first thing to do is to install the Thingsboard platform in a controlled local environment to define the features used for its creation and deployment on a personal computer, as shown in Table 2.

Table 2. Implementation technologies

Technology	**Version**
Operating system	Linux Ubuntu 20.04.3
Data base	PostgresSQL 12.9
Java JDK	OpenJDK 11.0.13
Thingsboard	Realease 3.3.2

The details for the construction of the dashboard will be defined in the following sections. Each section answers the guiding questions posed above, the selected metrics, the type of user considered, the visual representation of the metrics, and the context to be used to classify the indicators employed.

5.1 Metrics for the user

Being a dashboard for smart cities, the metrics have a common context which is focused on improving processes and services within the city, however, the context of each of them changes according to what is required to show the user specifically. The scope of application of metrics according to the case study of the research project is focused on smart cities [57]. According to the classifications of Alvarado [58], Borjaz [2], Estrada [59], Bosh [60], Cohen[61] and Sharifi [25] the

following categories are defined for the development of the adaptable dashboard: 1) Economy, 2) People, 3) Government, 4) Environment, 5) Living, 6) Mobility and 7) Data.

To perform the analysis of the performance indicators and whether they will be useful for the users of the smart city dashboard, we used the user-centered design tool called personas. Fig. 4 shows the relevant user characteristics of this instrument: 1) Demographic data, 2) Technology-related data, 3) Personal data, 4) User's motivation.

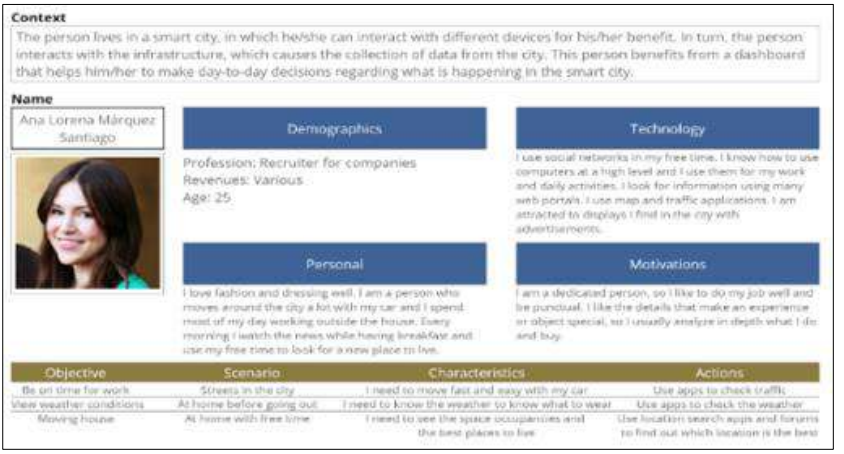

Fig. 4. Instrument Personas

The second instrument used was the empathy mapping tool (see Fig. 5). This map makes it possible to identify some interactions and experiences that users have when using the dashboard. It consists of four segments in which data is aggregated regarding what they think and feel, hear, see, say and do, and, finally, what are their pains and needs.

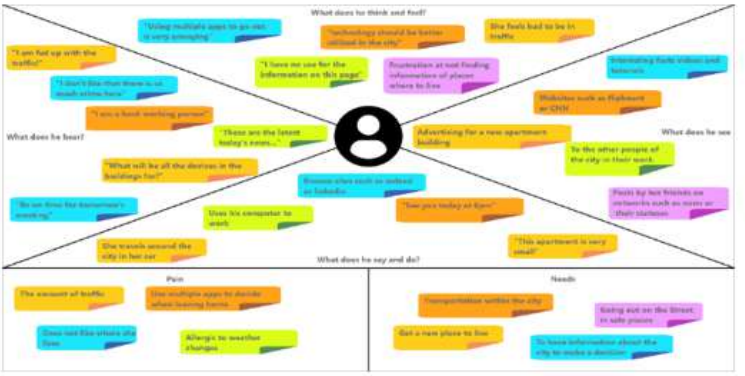

Fig. 5. Empathy map

5.2 Visual representation of metrics

The tool selected for the dashboard has a pre-established set of charts that can be used for the construction. Some of the charts mentioned in the section IV are mentioned here, and new charts belonging to the tool have been added 1) Last-value charts, 2) Time series, 3) Remote controls, 4) Alarms, 5) Maps with position indicators, 6) Gauge plots, 7) Digital calendars and 8) Customizable graphics.

5.3 Dashboard design

The adaptable dashboard interface design (see Fig. 6) follows the style of the thingsboard platform. It has a side menu where users can select the section of the system where they want to be located, it has a home section where the recently consulted information is shown, a section to visualize their configured dashboards, an information section where they can generate information reports, they can consult the information sources and the data with which the system is being fed, and it has a gallery of components to modify the parameters of the used visualization. In addition, there is a configuration section that allows you to change the way the graphical interface looks as well as colors and font sizes.

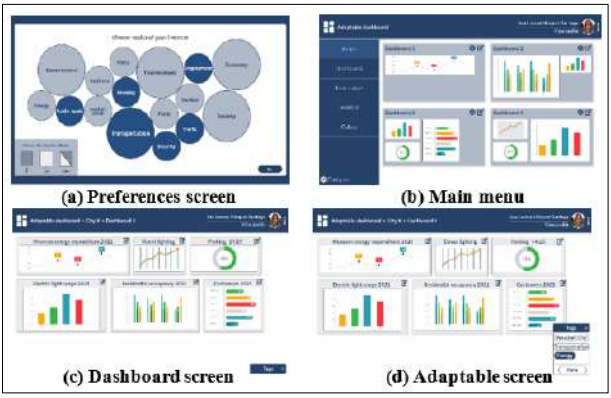

Fig. 6 Adaptable dashboard

The first image (see Fig. 6a) shows a configuration screen where users can select the different areas of interest with which they will interact. The general categories are represented as larger bubbles, while some subcategories and derived indicators are the smaller bubbles. This screen will serve as a starting point for users to select information relevant to them, from which a dashboard specific to their selection will be built. This interface complies with Raymond's usability rules [62] by making the selection of items limited to a number and with a consistent response to the user's actions.

Figure 6b corresponds to the main dashboard menu. In this screen the user can switch between the different sections of the platform: dashboards, information, sources, gallery, and configuration. In the internal screen of the dashboard, the user can interact with the charts to modify their size and position; each of the charts is inside its widget, which shows information related to the data. At the bottom is the mechanism to adapt the interface, which is a drop-down component to avoid influencing the interaction with the data.

Finally, the fourth image shows the mechanism whereby the user will be able to modify the information that is currently on the screen. The category selector allows the user to focus his attention on a single type of information and, at the same time, to display more related information without having to use a search engine.

6. Dashboard evaluation

The objective of the evaluation was to gather information on the interaction and user experience of the adaptable dashboard for smart cities through tests with users.

As part of the evaluation, the prototype shown in the previous section was used to obtain feedback on the design and use of the adaptable dashboard mechanism. Specifically, we sought to evaluate if the component that adapts the interface with the labels (see Fig. 6d) is perceived as comfortable and easy to use.

Since the perception of the information for each user may vary, we used questionnaires to gather user feedback on different aspects of the prototype, so that they could make comments that would

help improve the dashboard. We selected the post-task Single Ease Questionnaire (SEQ) [63] to obtain feedback on specific tasks with the user interface, it has 1 question for each activity carried out with a scale of 1 to 7, in addition, three questions were added to collect user context information. We additionally used the post-study System Usability Scale (SUS) [64] questionnaire to obtain user feedback on the use of the prototype, it has 10 questions with a scale of 1 to 5. Both questionnaires complete a total of 16 questions in total. These questionnaires were selected because they have been used to evaluate dashboard interfaces in different fields of study [65-68]. Sauro [69] defines the reliability of the questionnaires used.

The tasks that users had to complete are:

1) Create a new dashboard;
2) Enter a dashboard to view your charts;
3) Configure transport and energy charts in the dashboard.

In total, the evaluations were estimated to last a total of 10 minutes for each user to provide sufficient time for them to comment on the design and experience of using the prototype.

6.1 Participants and process

Four people participated in the dashboard test. Each of them lives within a city in the process of developing to be smart (Mexico City) [1]. Because they are in different locations, remote testing was conducted via video calls.

Users were asked to connect through videoconferencing software in which they were able to share their camera and screen. The first step of the testing was to provide the user with the research context, explaining what a dashboard is and its application in multiple areas. Then, the user was explained the testing procedure.

Subsequently, the user was informed of the letter of consent in which he/she agrees to the recording of the session and is notified that the data will be used privately for research purposes. A total of 3 links were provided to the user, the first was a form in which the user agreed to participate in the tests. The second was the link to the SEQ and SUS questionnaires (divided by pages), and the third was the dashboard prototype accessible via the web.

The environment for the test was set up when the user logged into the video call. We started by turning on his camera, verifying that when sharing the screen, he could share only the dashboard window and that he could access the Internet to view the questionnaires and the dashboard.

The test was controlled by segmenting the entire session into 15-minute slots, which allocated 10 minutes for testing and 5 minutes for collecting feedback and having a backup space for the next participant.

6.2 Evaluation results

The results of the SEQ questionnaire are shown in Table 3. The results obtained are divided by the average, median, success rate, error rate, and average score obtained in the SEQ questionnaires.

Table 3. SEQ questionnaire results

Activity	Average	Median	Success rate	Error rate	AVG score
1	64.33	58	75%	15%	4.25
2	22.33	18.5	100%	0%	7
3	23.33	20	100%	0%	6.25
AVG	**36.66**	**32.16**	**91.6%**	**8.4%**	**5.83**

The first result of interest is the time it took the users to complete activity 1 (creating a new dashboard). Besides being the one that took the longest time (64.3 seconds), this activity is the only task with a 75% success rate because one user was unable to complete the activity. We can also see this effect reflected in the median column, which for activity 1 was 58 seconds. Regarding the rating of the activity, the average was considerably low with a total of 4.25 being the maximum 7. This

means, in conjunction with the time obtained, that the task was complex to understand and that users have difficulties in configuring a new dashboard from the main interface.

Even though task 1 had a low result, tasks 2 and 3 had 100% success rate, each of them with a similar average time (22.3 and 23.3 seconds). The most notable difference between these two tasks lies in the rating obtained, task 2 (entering a dashboard) obtained a rating of 7, indicating that it is a simple task to perform and that there was no problem at all, while task 3 (configuring charts in the dashboard) obtained a rating of 6.25, where users considered it a little more difficult to perform. Finally, the overall rating of the dashboard prototype was 5.83, which indicates that there are still features that can be improved to give users a better experience when using the platform, these features fall mostly on the method of creating a new dashboard and the label system that adapts the interface.

Moreover, Table 4 presents the SUS evaluation results divided by user. The table shows both the values of the total sums of each questionnaire and the score obtained on a scale from 0 to 100.

Table 4. SUS questionnaire results

User	**Sum of ratings**	**SUS Rating**
1	35	87.5
2	30	75
3	33	82.5
4	29	72.5
Overall rating		**79.37**

In the ratings of this instrument, the lowest rating corresponds to user 4 with a value of 72.5 followed by user 2 with 75. In contrast, users 1 and 3 rated the usability of the dashboard with a higher value (87.5 and 82.5). This indicates that the opinions of the different users can tell us which parts of the system meet their objectives and which parts need to be corrected.

Despite rating differences, the overall grade of 79.37 is in an acceptable range according to the interpretation of the results developed by [70], specifically, the value corresponds to a grade of 'C', where a value lower than 60 is an F, between 60 and 69 to D, between 70 and 79 to C, between 80 and 89 to B and higher than 90 is an A.

The result obtained, being right in the middle of the classification with the letter C, indicates that there are still factors that can be improved for users enjoy interacting with the system, making it useful and easy to use. In the answers of this questionnaire, we were able to analyze that both the way of creating dashboards and the functionality of adaptation by tags is functional to the user, but there are sections that, with the changes suggested by them, will improve the way in which the usability of the adaptable dashboard for smart cities is perceived.

Once the results of the questionnaires have been analyzed, it is important to mention the main comments made by users about the design and functionality of the prototype dashboard.

Feedback from users is valuable because it will serve as a basis to understand what is happening with the dashboard and for further studies to improve the functionalities and aspects of the user interface.

7. Conclusions and future work

This paper presents the results of a systematic review of the literature on information visualization in adaptable dashboards. We used the methodology proposed by [15] for the elaboration of literature reviews, setting out a search strategy that starts with the research questions. We used a search string in digital databases to obtain a total of 24 relevant articles used to answer the research questions posed. In addition, using the information related to adaptability, a proposal for component specification was built for later use in the design of an adaptable dashboard. We used the user-centered tools personas and empathy map to know the characteristics of the user and then propose and evaluate the design of the interface that will allow adapting the dashboard for smart city. The evaluation was carried out with 4 users with a total of 3 tasks. It was carried out using 2 evaluation

instruments: SEQ and SUS. The results obtained for the SEQ questionnaire is a total of 5.83 points out of 7. The results of the SUS questionnaire gave a value of 79.37 being in an acceptable rating.

As future work, user feedback must be addressed in both the design and development of the dashboard to improve the user experience. Furthermore, once a refined design has been obtained, it is necessary to put it into practice in a real environment.

References

[1] Alvarado-López R.A. Ciudades inteligentes y sostenibles: una medición a cinco ciudades de México. Estudios Sociales. Revista de Alimentación Contemporánea y Desarrollo Regional, vol. 30, issue 55, 2020, 28 p. (in Spanish).

[2] Borja Zapata J.S., Arce Ruiz R.M., Soria Lara J.A. Modelo para evaluar el cumplimiento de objetivos de ciudades inteligentes. In Proc. of the Congresso Internacional "Sustentabilidade Urbana", 2018, pp. 225-234 (in Spanish).

[3] Tachtler F.M. Best way to go? Intriguing Citizens to investigate what is behind smart city technologies. In Proc. of the ACM Conference on Designing Interactive Systems, 2017, pp. 28-33.

[4] Few S. Dashboard Confusion Revisited. Perceptual Edge, 2007, pp. 1-6. URL: http://perceptualedge.com/articles/visual_business_intelligence/dboard_confusion_revisited.pdf.

[5] Vila R.A., Estevez E., Fillottrani P.R. The design and use of dashboards for driving decision-making in the public sector. In Proc. of the 11th International Conference on Theory and Practice of Electronic Governance, 2018, pp. 382-388.

[6] Latifah A., Supangkat S.H., Ramelan A. Smart Building: A Literature Review. In Proc. of the International Conference on ICT for Smart Society, 2020, pp. 1-6.

[7] Cepero-Garcia M.T., Montane-Jimenez L.G. Visualization to support decision-making in cities: Advances, technology, challenges, and opportunities. In Proc. of the 8th International Conference in Software Engineering Research and Innovation (CONISOFT), 2020, pp. 198-207.

[8] Laramee R.S., Turkay C., Joshi A. Visualization for Smart City Applications. IEEE Computer Graphics and Applications, vol. 38, issue 5, 2018, pp. 36–37.

[9] Ergasheva S., Ivanov V. et al. InnoMetrics Dashboard: The Design, and Implementation of the Adaptable Dashboard for Energy. In IFIP Advances in Information and Communication Technology, vol. 582, Springer, 2020, pp. 163-176.

[10] Tokola H., Groger et al. Designing manufacturing dashboards on the basis of a key performance indicator survey. Procedia CIRP, vol. 57, 2016, pp. 619-624.

[11] Pettit C., Widjaja I. et al. Visualisation support for exploring urban space and place. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 1, 2012, pp. 153-158.

[12] Zdraveski V., Mishev K. et al. ISO-Standardized Smart City Platform Architecture and Dashboard. IEEE Pervasive Computing, vol. 16, issue 2, 2017, pp. 35-43.

[13] Shafiq S.I., Szczerbicki E., Sanin C. Proposition of the methodology for Data Acquisition, Analysis and Visualization in support of Industry 4.0. Procedia Computer Science, vol. 159, 2019, pp. 1976-1985.

[14] Contreras-Figueroa V., Montané-Jiménez L.G. et al. Information visualization in adaptable dashboards for smart cities: A systematic review. In Proc. of the 9th International Conference in Software Engineering Research and Innovation (CONISOFT), 2021, pp. 34-43.

[15] Kitchenham B. Procedures for performing systematic reviews. Keele University Technical Report TR/SE-040, NICTA Technical Report 0400011T.1, 2004, 34 p.

[16] Few S. Information Dashboard Design: Displaying Data for AT-A-Glance Monitoring. Analytics Press, 2013, 260 p.

[17] Zhang J., Johnson K.A. et al. Human-Centered Information Visualization. Proceedings of the International Workshop on Dynamic Visualizations and Learning, 2002, 7 p.

[18] Matheus R., Janssen M., Maheshwari D. Data science empowering the public: Data-driven dashboards for transparent and accountable decision-making in smart cities. Government Information Quarterly, vol. 37, issue 3, 2020, article no. 101284, 2020, 10 p.

[19] Protopsaltis A., Sarigiannidis P. et al. Data visualization in Internet of Things: Tools, methodologies, and challenges. In Proc. of the 15th International Conference on Availability, Reliability and Security, 2020, article no. 110, 11 p.

[20] Tong X., Wu Z. Study of Chinese city “portrait” based on data visualization: take city dashboard for example. Lecture Notes in Computer Science, vol. 10919, 2018, pp. 353-364.

[21] Batty M. A perspective on city dashboards. Regional Studies, Regional Science, vol. 2, issue 1, 2015, pp. 29-32.

[22] Han Q., Nesi P. et al. Smart City Dashboards: Design, Development, and Evaluation. In Proc. of the IEEE International Conference on Human-Machine Systems, 2020, 4 p.

[23] Dobraja I., Kraak M.-J., Engelhardt Y. Facilitating insights with a user adaptable dashboard, illustrated by airport connectivity data. Proceedings of the International Cartographic Association, vol. 1, 2018, article no. 30, 6 p.

[24] Kourtit K., Nijkamp P. Big data dashboards as smart decision support tools for i-cities. an experiment on Stockholm. Land Use Policy, vol. 71, 2018, pp. 24-35.

[25] Sharifi A. A typology of smart city assessment tools and indicator sets. Sustainable Cities and Society, vol. 53, 2019, article no. 101936, 15 p.

[26] O’Neill D., Peoples C. Using IT to monitor well-being and city experiences. IEEE Potentials, vol. 35, issue 6, 2016, pp. 29-34.

[27] Peddoju S.K., Upadhyay H. Evaluation of IoT data visualization tools and techniques. In Data Visualization, Springer, 2020, pp. 115-139.

[28] Lv Z., Li X. et al. Government affairs service platform for smart city. Future Generation Computer Systems, vol. 81, 2018, pp. 443-451.

[29] Vicuna P., Mudigonda S. et al. A generic and flexible geospatial data warehousing and analysis framework for transportation performance measurement in smart connected cities. Procedia Computer Science, vol. 155, 2019, pp. 226–233.

[30] Purahoo Z., Cheerkoot-Jalim S. SenseAPP: An IoTBased Mobile Crowdsensing Application for Smart Cities. In Proc. of the 3rd International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering, 2020, pp. 47-52.

[31] Moustaka V., Maitis A. et al. CityDNA Dynamics: A Model for Smart City Maturity and Performance Benchmarking. In the Companion Proceedings of the Web Conference, 2020, pp. 829-833.

[32] Chen S., Dong H. Visualizing Toronto City Data with HoloLens: Using Augmented Reality for a City Model. IEEE Consumer Electronics Magazine, vol. 7, issue 3, 2018, pp. 73-80.

[33] Brunetto O. City dashboards: The case of Trieste: Trieste overview Lecture Notes in Computer Science, vol. 10407, 2017, pp. 710-721.

[34] Habibzadeh H., Kaptan C. et al. Smart City System Design. ACM Computing Surveys, vol. 52, issue 2, 2019, pp. 1-38.

[35] Rolim D., Silva J. et al. Web-based development and visualization dashboards for smart city applications. Lecture Notes in Computer Science, vol. 12128, 2020, pp. 337-344.

[36] Chrysantina A., Sæbø J.I. Assessing user-designed dashboards: a case for developing data visualization competency. IFIP Advances in Information and Communication Technology, vol. 551, Springer, 2019, pp. 448-459.

[37] Elshehaly M., Randell R. et al. QualDash: Adaptable generation of visualisation dashboards for healthcare quality improvement. IEEE Transactions on Visualization and Computer Graphics, vol. 27, issue 2, 2021, pp. 689-699.

[38] Dostal R., Pribyl O., Svitek M. City infrastructure evaluation using urban simulation tools. In Proc. of the Smart Cities Symposium Prague, 2020, 6 p.

[39] Lu M., Liu S. et al. Design and Implementation of Power Information Visualization Platform Based on Smart Meter. In Proc. of the 3rd International Conference on Information Science and Control Engineering, 2016, pp. 297-301.

[40] Curry E., Fabritius W. et al. A Model for Internet of Things Enhanced User Experience in Smart Environments. In Real-time Linked Dataspaces, Springer, 2020, pp. 271-294.

[41] Molich R., Nielsen J. Improving a Human-Computer Dialogue. Communications of the ACM, vol. 33, issue 3, 1990, pp. 338-348.

[42] Limon-Ruiz M., Larios-Rosillo V.M. et al. User-oriented representation of Smart Cities indicators to support citizens governments decision-making processes. In Proc. of the 5th IEEE International Smart Cities Conference, 2019, pp. 396-401.

[43] Chan A.L., Chua G.G. et al. Practical experience with smart cities platform design. In Proc. of the IEEE World Forum on Internet of Things, 2018, pp. 470-475.

[44] Few S. Why Most Dashboards Fail. Perceptual Edge, 2007, 2 p. URL: https://www.perceptualedge.com/articles/misc/WhyMostDashboardsFail.pdf.

[45] Alves A.P., Milani A.M., Manssour I.H. Visual Analytics System for Energy Data in Smart Cities and Buildings. In Proc. of the IEEE International Smart Cities Conference, 2020, 8 p.

[46] Rojas E., Bastidas V., Cabrera C. Cities-Board: A Framework to Automate the Development of Smart Cities Dashboards. IEEE Internet of Things Journal, vol. 7, issue 10, 2020, pp. 10128-10136.

[47] Silva J., Mojica J. et al. Algorithms for the Control of Key Performance Indicators for Smart Cities. Procedia Computer Science, vol. 170, 2020, pp. 971-976.

[48] Pressman R. Ingenieria del software: Un enfoque practico. 7ma. McGrawhill, 2010, 777 p. (in Spanish).

[49] Ivanov V., Larionova D. et al. Design of a dashboard of software metrics for adaptable, energy efficient applications. In Proc. of the International Distributed Multimedia Systems Conference on Visualization and Visual Languages, 2019, pp. 75-82.

[50] Young G.W., Kitchin R., Naji J. Building city dashboards for different types of users. Journal of Urban Technology, vol. 28, issues 1-2, 2021, pp. 289-309.

[51] Open data, Paris. URL: https://opendata.paris.fr/pages/home/, 19 de enero 2022.

[52] VancouverCity. URL: https://vancouver.opendatasoft.com/pages/help/, 19 de enero 2022.

[53] Gobierno de México. URL: https://datos.cdmx.gob.mx/group/educacionciencia-y-tecnologia, 19 de enero 2022.

[54] Thingsboard. URL: https://thingsboard.io/, 19 de enero 2022.

[55] Oppermann R. Adaptively supported Adaptability. International Journal of Human-Computer Studies, vol. 40, issue 3, 1994, pp. 455-472.

[56] Strugar D. Complex Systems: On Design and Architecture of Adaptable Dashboards, Lecture Notes in Computer Science, vol. 11771, 2019, pp. 176-186.

[57] U4SSC - Collection Methodology for Key Performance Indicators for Smart Sustainable Cities, 2017, 134 p.

[58] Alvarado López R. Ciudad inteligente y sostenible: una estrategia de innovación inclusiva. PAAKAT: Revista de Tecnología y Sociedad, vol. 7, issue 13, 2017, 17 p. (in Spanish).

[59] Estrada E., Peña A. et al. Algoritmos para el control de indicadores clave de desempeño para Smart Cities. Research in Computing Science, vol. 147, issue 8, 2018, pp. 121-133 (in Spanish).

[60] Bosch P., Jongeneel S. et al. CITYkeys indicators for smart city projects and smart cities. European Commission within the H2020 Programme, Technical Report, 2017, 306 p.

[61] Cohen B. What exactly is a smart city? Co. Exist, issue 19, 2012. URL: https://www.fastcompany.com/1680538/what-exactly-is-a-smart-city, 19 de enero 2022.

[62] Raymond E.S., Landley R.W. The Art of UNIX Usability. 2004. URL: http://www.catb.org/esr/writings/taouu/taouu.html, 19 de enero 2022.

[63] Tedesco D., Tullis T. A comparison of methods for eliciting post-task subjective ratings in usability testing. In Proc. of the Usability Professionals Association Conference, 2006, pp. 1-9.

[64] Brooke J. SUS: A 'Quick and Dirty' Usability Scale. In Usability Evaluation in Industry, CRC Press, 1996, pp. 107-114.

[65] Yamamoto S., Mori H., eds. Human Interface and the Management of Information. Visual Information and Knowledge Management. Lecture Notes in Computer Science, vol. 11569, 2019, 652 p.

[66] Celi E. Application of Dashboards and Scorecards for Learning Models IT Risk Management: A User Experience. Lecture Notes in Computer Science, vol. 9188, 2015, pp 153-165.

[67] Wu D.T.Y., Vennemeyer S. et al. Usability testing of an interactive dashboard for surgical quality improvement in a large congenital heart center. Applied Clinical Informatics, vol. 10, issue 5, 2019, pp. 859-869.

[68] Ullmann T., De Liddo A., Bachler M. A visualization dashboard for contested collective intelligence learning analytics to improve sensemaking of group discussion. RIED. Revista Iberoamericana de Educaci´on a Distancia, vol. 22, issue 1, 2019, p. 41-80.

[69] Sauro J., Lewis J. Standardized usability questionnaires. In Quantifying the User Experience. 1st edition. Morgan Kaufmann, 2012, pp. 185-240.

[70] Bangor A., Kortum P., Miller J. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. Journal of Usability Studies, vol. 4, issue 3, 2009, pp. 114-123.

Information about authors / Информация об авторах

Victor CONTRERAS-FIGUEROA – Master student in User-Centered Interactive Systems. Main research interests: Visualization, Human-Computer Interaction, Smart Cities and Web Development.

Виктор КОНТРЕРАС-ФИГЕРОА – студент магистратуры в области интерактивных систем, ориентированных на пользователя. Основные научные интересы: визуализация, взаимодействие человека с компьютером, умные города и веб-разработка.

Luis Gerardo MONTANÉ-JIMÉNEZ, Ph.D. in Computer Science, Professor at the Faculty of Statistics and Informatics. Main research interests: Computer-Supported Cooperative Work (CSCW), Visualization, Human-Computer Interaction, Context-Aware Computing and Videogame Development.

Луис Херардо МОНТАНЕ-ХИМЕНЕС, кандидат компьютерных наук, профессор факультета статистики и информатики. Основные исследовательские интересы: совместная работа с компьютерной поддержкой, визуализация, взаимодействие человека и компьютера, контекстно-зависимые вычисления и разработка видеоигр.

María CEPERO-GARCÍA – Master's Degree in User-Centered Interactive Systems, Professor at the Faculty of Statistics and Informatics of the Universidad Veracruzana. Her areas of interest are: Computer-Supported Cooperative Work (CSCW), Human-Computer Interaction, Data visualization, and Smart Cities.

Мария СЕПЕРО-ГАРСИА – профессор факультета статистики и информатики. Сферы ее интересов: совместная работа с компьютерной поддержкой, взаимодействие человека и компьютера, визуализация данных и умные города.

Edgard BENÍTEZ-GUERRERO – Ph. D. in Computer Science from the University of Grenoble in France, Professor at the Faculty of Statistics and Informatics. Research interests: Human Computer Interaction, Artificial Intelligence, Collaborative Computing, Data Management and Visualization.

Эдгар БЕНИТЕС-ГЕРРЕРО – кандидат компьютерных наук Гренобльского университета во Франции, профессор факультета статистики и информатики. Научные интересы: взаимодействие человека с компьютером, искусственный интеллект, совместные вычисления, управление данными и визуализация.

Carmen MEZURA-GODOY – Ph. D. in Computer Science from the University of Savoie in France, Professor at the Faculty of Statistics and Informatics. Main research interests: Human Computer Interaction, User Experience, Visualization, Computer Support Collaborative Work and Multiagent Systems.

Кармен МЕЗУРА-ГОДОЙ – кандидат компьютерных наук Университета Савойи во Франции, профессор факультета статистики и информатики. Основные исследовательские интересы: взаимодействие человека с компьютером, пользовательский опыт, визуализация, совместная работа компьютерной поддержки и многоагентные системы.

DOI: 10.15514/ISPRAS-2023-35(1)-2

Challenges in Conversational Agents to support Informal Caregivers of People with Dementia

[1] *S. Jiménez, ORCID: 0000-0003-0938-7291 <samantha.jimenez@tectijuana.edu.mx>*
[2] *J. Favela, ORCID: 0000-0003-2967-9654 <favela@cicese.mx>*
[1] *A. Quezada, ORCID: 0000-0001-5706-8047 <angeles.quezada@tectijuana.edu.mx>*
[3] *R. Ramachandran, ORCID: 0000-0003-0355-698X <raj.ramachandran@uwe.ac.uk>*
[4] *R. Juárez-Ramírez, ORCID: 0000-0002-5825-2433 <reyesjua@uabc.edu.mx>*

[1] *Instituto Tecnológico de Tijuana,*
Tijuana, México, 22424
[2] *Centro de Investigación Científica y de Educación Superior de Ensenada*
3918, Ensenada-Tijuana Highway, Ensenada, México, 22860
[3] *University of the West of England,*
Coldharbour ln, Bristol, BS16 1QY, UK
[4] *Universidad Autónoma de Baja California,*
Ensenada, México, 21100

Abstract. People who have dementia (PwD) experience deteriorating executive functions, in particular their working memory, and therefore find it hard to complete multistep tasks or activities of daily living. There is no doubt that during the pandemic, PwD and their caregivers were particularly vulnerable, often isolated which affected their mental and physical health. Their ability to live independently was hampered, fomenting depression in the PwD and burnout on informal caregivers. Information technology can support dementia care improving the quality of life of PwD and easing the burden on caregivers. There is an increasing demand to support informal caregivers and improve their well-being by making dementia challenges less severe. This study uses qualitative techniques to design a model with technological strategies based on semi-structured interviews applied to seven informal caregivers from two different countries. Based on these interviews we developed design insights for implementing solutions to help informal caregivers take care of their PwD at home using conversational agents. We hope that the findings presented in this study will help researchers, and developers design solutions that can support PwD and informal caregivers.

Keywords: Dementia; caregivers; Semi-structured interviews; Conversational agent

For citation: Jiménez S., Favela J., Quezada A., Ramachandran R., Juárez-Ramírez R. Challenges in Conversational Agents to support Informal Caregivers of People with Dementia. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 25-34. DOI: 10.15514/ISPRAS-2023-35(1)-2

Проблемы использования разговорных агентов для поддержки неформальных опекунов людей с деменцией

[1] *С. Хименес, ORCID: 0000-0003-0938-7291 <samantha.jimenez@tectijuana.edu.mx>*
[2] *Х. Фавела, ORCID: 0000-0003-2967-9654 <favela@cicese.mx>*
[1] *А. Кесада, ORCID: 0000-0001-5706-8047 <angeles.quezada@tectijuana.edu.mx>*
[3] *Р. Рамачандран, ORCID: 0000-0003-0355-698X <raj.ramachandran@uwe.ac.uk>*
[4] *Р. Хуарес-Рамирес, ORCID: 0000-0002-5825-2433 <reyesjua@uabc.edu.mx>*

[1] *Тихуанский технологический институт,*
Мексика, 22414, Нижняя Калифорния, Тихуана
[2] *Центр научных исследований и высшего образования,*
Мексика, 22860, Нижняя Калифорния, Энсенада, ш. Тихуана-Энсенада, 3918
[3] *Университет Западной Англии,*
Великобритания, BS16 1QY, Бристоль, Колдхарбор-лейн
[4] *Автономный университет Нижней Калифорнии (UABC),*
Мексика, 21100, Нижняя Калифорния, Энсенада

Аннотация. У людей с деменцией (PwD) ухудшаются исполнительные функции, в частности их кратковременная память, и поэтому им трудно выполнять повседневные многоэтапные задачи или действия. Нет сомнений в том, что во время пандемии инвалиды и лица, осуществляющие уход за ними, были особенно уязвимы, часто изолированы, что сказывалось на их психическом и физическом здоровье. Их способность жить независимо была ограничена, что провоцировало депрессию у людей с инвалидностью и эмоциональное истощение у неформальных опекунов. Информационные технологии могут способствовать лечению деменции, улучшая качество жизни людей с инвалидностью и облегчая нагрузку на лиц, осуществляющих уход. Растет потребность в поддержке неформальных опекунов и улучшении их благополучия за счет уменьшения серьезности проблем с деменцией. В этом исследовании используются качественные методы для разработки модели с использованием технологических стратегий, которые основываются на полуструктурированных интервью, примененных к семи неформальным опекунам из двух разных стран. На основе этих интервью мы разработали идеи по внедрению решений, которые помогут лицам, осуществляющим неформальный уход, заботиться об инвалидах дома с помощью диалоговых агентов. Мы надеемся, что результаты, представленные в этом исследовании, помогут исследователям и разработчикам разработать решения, которые могут помочь людям с инвалидностью и неформальным опекунам.

Ключевые слова: деменция; опекуны, полуструктурированные интервью; разговорные агенты

Для цитирования: Хименес С., Фавела Х., Кесада А., Рамачандран Р., Хуарес-Рамирес Р. Проблемы использования разговорных агентов для поддержки неформальных опекунов людей с деменцией. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 25-34. DOI: 10.15514/ISPRAS–2023–35(1)–2

1. Introduction

According to the 2016 World Alzheimer Report it is expected that by 2050 there will be 131.5 million of People with Dementia (PwD). Between the 60 and the 70 percent of all dementia cases are estimated to have Alzheimer's disease. People who have dementia typically experience deteriorating executive functions, particularly their working memory, and therefore find it hard to complete multistep tasks or activities of daily living [1]. Currently, there is no known cure for dementia, with pharmacological and non-pharmacological interventions focused on improving the quality of life of patients, and caregivers [2].

There is no doubt that during the COVID-19 pandemic PwD and their caregivers were particularly vulnerable considering their mental health and physical health [3]. Recent studies show that the elderly's mental health was affected by isolation, their ability to live independently was hampered, provoking in some cases depression in the PwD and exhaustion in care providers [3].

Information technology can support dementia care easing the burden on caregivers [4]. Studies have explored the use of virtual assistants, robots, virtual reality, music technology, and Internet of Things to assist PwD [5]. Specifically, Intelligent voice assistants have been suggested as a potential source of assistance to caregivers, who are usually older adults themselves, and have limited technological skills [6].

Rugiano et al. [6] evaluated intelligent voice assistants from a usability perspective, assessing efficiency, effectiveness, and satisfaction. The voice assistants have shown to be useful not only for elderly people but for children with autism as well [7], [8].

There is an increasing demand to provide support to informal caregivers and improve their well-being by lessening the challenges associated to dementia. Devices such as smartphones, tablets, and computers can be a helpful tool in alleviating the caregiver's psychological burden, encouraging social engagement, and easing the burden of daily activities [6], [9]. However, devices with a touch screen are harder to use for elderly people while voice assistant provide a more natural interface [8].

The objective of this study is to design a model with technological strategies based on a qualitative analysis of semi-structured interviews conducted with informal caregivers. The proposed model aims to support technological solutions to help informal caregivers take care of their PwD at home. The findings presented in this study mean to inform researchers and developers in the desing design solutions that can support PwD and informal caregivers.

The rest of the paper is structured as follows. Section 2 describes the related work, Section 3 explains the methodology that guided the interviews, and in Section 4 we present the model that resulted from its analysis. Then, Section 5 presents design insights derived from the study and Section 6 concludes the study and suggests some lines of future work.

2. Related work

A qualitative study analyzed the experience of family caregivers for PwD in China [10]. The authors found that caregivers could positively interact with PwD by employing positive strategies, creating opportunities, and organizing meaningful activities [10]. Such methods are essential for maintaining harmonious family relationships. The authors suggested that nurses can implement or support these activities to the caregivers.

In a systematic review of technology-based interventions for dementia caregivers, it was reported that technology-based interventions often demonstrated efficacy in improving psychosocial outcomes but have not demonstrated efficacy in improving caregiving skills or care self-efficacy [6]. The benefits of using chatbots for healthcare have also been identified for patients and healthcare systems alike [6].

Another study presents a qualitative and quantitative analysis on the use and acceptance of technology in caregivers [3]. This study reveals that computer use decreases with age. And the higher the level of caregivers' education, the more often they use both smartphones and computers. Noteworthy, the level of education decreased as age increased, which could be an additional factor in technology use difficulties. According to [11] voice-based conversational agents are easier to use than touch interfaces, they were proved with autistic users.

Previous studies have highlighted that voice-based chatbots may be especially useful for older adults for health-related communication and information seeking, because they operate through voice-driven conversation, which may be helpful for those with low computer literacy [6].

As conversational agents become pervasive, studies are being conducted to assess its utility and adoption among older adults. A qualitative study with 37 community-dwelling older adults, for instance, found that they have a positive attitude towards the adoption of this technology, particularly to support their health management, although some concerns were raised regarding privacy [12]. Interestingly, a study on the adoption of voice interface technologies among patients with heart failure found that older participants used the technology more frequently [13]. There is increasing evidence that older adults with limited previous exposure to smart speakers adopt smart speakers

without much concern, with playing musing and asking health related questions among the most popular tasks [14].

While studies involving people with dementia are less frequent, some early evidence provide optimism about their adoption and their efficacy for caregivers. For instance, a 12-week trial conducted in Canada found that depression and anxiety among caregivers improved as a result of the intervention involving smart technologies with audio prompts to monitor the sleep of PwD [15]. Smart speakers have shown to improve intelligibility among adults with intellectual disability [16]. Solutions have also been proposed to address specific problems, such as a personalized diet voice-assistant implemented to support caregivers of people with Alzheimer's [17]. Finally, a recent systematic review of chatbots to support PwD and their caregivers found only 6 specifically designed for this audience, that while being easy to use the authors found them to have limitations in performance and content, suggesting that more research and development is needed in this area [6].

Speech is largely considered as the most powerful and effective communication mode for an assistive social robot to interact with its users. Recent technological developments and research results are contributing to solving the challenges that characterize the design and implementation of spoken dialogue systems for human-robot interaction with PwD [18].

Additional research has been conducted in addressing symptoms of dementia. For instance, in [13] an ontology is proposed for representing the domain knowledge for agitation in dementia. It represents the domain knowledge specific to non-pharmacological intervention for agitation in dementia, particularly in long-term care setting. In a similar direction [19] proposes an ontology nonpharmacological intervention for dementia to support a model proposed for ambient-assisted intervention systems (AAIS) that really on ambient computing to monitor symptoms and enact interventions.

3. Method

This section presents the design of the qualitative study, describes the participants, data collection and data analysis.

3.1 Design

A descriptive phenomenological qualitative study was conducted [10], [20]. This approach lends to a deep understanding of the experiences and feelings of the caregivers who interact with Patients with Dementia (PwD). The questionnaire designed for data gathering included open-ended questions which let the interviewee describe daily activities, experiences, difficulties, and feelings of interacting with PwD.

3.2 Participants

Caregivers of patient with dementia (describe how these patients were diagnosed) were included. These participants should meet the flowing inclusion criteria: 1) age of 18 years and above, 2) main care for PwD for at least 6 months; and 3) currently living with the PwD. We contacted the participants thought telephone, and seven caregivers completed the interviews. Four women and three men from Mexico and UK participated in this study. All were family caregivers for PwD. The duration of the experience taking care the patient varied from 1-10 years. All the caregivers lived with the PwD. Table 1 shows the characteristics of the participants.

Table 1. Characteristics of participants

P	AP	GP	TE	RPwD	GPwD	APwD
1	45	Male	Y	Son	Female	80
2	74	Female	N	Daughter	Female	96
3	22	Female	Y	Grand daughter	Female	101

4	42	Female	Y	Daughter	Female	68
5	34	Male	Y	Grandson	-	-
6	43	Female	Y	Daughter	Male	80
7	33	Male	Y	Grandson	-	-
P: participant no, AP: age of participant, TE: techonolog experience, RPwD: relationship with the PwD, GPwD: gender of PwD, APwD: age of PwD						

3.3 Data Collection

A semi structured interview guide was developed and then refined via discussion with the study team. The final guide included the following sections: 1) demographic information, 2) experience in the use of technology, 3) experience in caring the PwD, 4) faced difficulties. The instrument has 27 questions, the following are some examples of the questions that we applied:

- How was your experience taking care of the PwD?
- What were the insights that motivates you to search medical help?
- Can you mention a recent situation of frustrations recently?
- Thinking about the last year, what were the most challenging situations that you faced with the PwD related to his/her behavior?
- What was your reaction in the previously mentioned situations?
- Does the PwD faced repetitive questioning or behaviors? Explain them.
- What was the context when the PwD experimented the repetitive behaviors?
- How do you think that a chatbox like Alexa could help in the caring of the PwD?

The interviewer explained the purpose of the interview and discussed the caregiver right to discontinue at any time for any reason. Interview data were collected by audio recording for future analysis. The interviews were conducted at the preferred time of the caregivers to respect their schedules and provide a suitable environment to share their thoughts and experiences. Data were collected from September 2021 to March 2022. The interviews lasted an average of 22 mins (max=40 min, min=12 min).

3.4 Data Analysis

Interviews recordings were transcribed, and the transcripts were read and analyzed by the team. A coding framework was developed through thematic analysis [21]. The researchers became familiar with the information by listening to the interview's recordings during the transcription and repeatedly reading the transcripts. The data were then stored and coded in Atlas.ti. To maintain qualitative rigor, two researchers refined the coding categories.

4. Results

After the interview data were analyzed to identify core categories which describe the family caregivers experience of interacting with PwD. In each one of the categories, we present some of the interviewee answers, also we suggest how the conversational agents could help the PwD and the family caregivers.

4.1 Challenging behaviors of the PwD

It is well known that all the PwD have different behaviors and symptoms and frequently unexpected things happen. For that reason, it is difficult for the family caregivers to be prepared for all the different challenging situations she/he might experience. However, they start knowing the PwD and detecting the most common situations. For instance, in the interviews the participants mentioned:

P1: *"She forgets her things like money and medications".*

P5: *"He is asking the same questions again and it became frustrating."*

Forgetting things and repetitive questioning were common challenging behaviors mentioned by the interviewees. As we mentioned before the patients are different and each family faces different situations, for that reason the caregivers need strategies to face these specific problems.

4.2 Strategies to face challenging behaviors of PwD

Informal caregivers learn about the disease and start implementing different positive strategies to face the challenging behaviors they experience. For instance, they try to distract or entertain the PwD by incorporating activities that they like to do.

P1: *"I go out with her and try to make her feel busy".*

Frequent greetings and presentations are important to situate the PwD know and made her aware of who are the people around. One informant suggested that:

P6: *"It is important to greet him, ask him some questions".*

Formal caregivers and doctors suggest making the patients remember things so they can train their minds, having conversations about their youth will help too. The main goal is to keep their minds working. As one participant commented:

P6: *"Just ask him questions, I often start the conversation about something from his youth, or his teachings in school, so I just do that now, it definitely works, and it helps him."*

It is important to follow the conversation with the patient and answer their questions as many times as needed. The participants shared some of their strategies to handle difficult situations with the PwD.

P2: *"I repeat the things over and over again".*

It is important that the solutions provided to the PwD and the caregivers consider several aspects of both types of user such as: mental health, emotions, physical health, medication and personality. All these aspects could be used by the caregiver assistant to personalize or tailor the tasks.

P5: *"People with dementia tend to ask things, if you tell them to do something they ask why I should do that, so you must explain to them why it is good for them to do that. Basically, give them a short answer."*

The idea is to make them feel comfortable and loved but not patronize them making them feel as kids.

4.3 Support family caregivers' emotions

The family caregivers try to make the PwD feel as comfortable as possible, and sometimes they are not taking care of themselves [22]. They expressed that most of the time they feel sad to see how the disease is progressing. They also experience frustration facing all the repetitive behaviors and questioning. However, they are happy to have their loved one with them no matter the circumstances. For that reason, it will be important to provide some solutions that help the family caregivers to take care of the PwD.

P2: *"She is my mother, it does not matter the conditions I want to have her".*

P4: *"I feel frustrated and desperate".*

In the next section, we present some design insights based on the results of the interviews proposing a caregiver assistant. The proposal is presented as a conceptual meta-model.

5. Design insight

All the tasks where the caregiver assistant can help are non-pharmacological interventions for example, playing music, telling jokes, or executing relaxation routines. In the medical service interventions, the caregiver assistant will be more limited because it won't be able to diagnose the

disease or provide medication, but it could be useful providing medication alarms and reminders, also it has the option of calling 911 in case of an emergency.

The caregiver assistant can also assess the PwD asking specific questions or tasks to evaluate the PwD's performance and the progression of the disease. These assessments could be helpful for the family caregivers and as well as the physician.

The conversational agents can interact with the patients answering repetitive questions (Fig. 1). They can also act as medication reminders or other scheduled activities. All these activities can be easily handled by these agents helping the caregivers on the support to the PwD and reducing the stress and frustration. There are several attempts on including conversational agents in the activities of daily life like playing music, some relaxation activities, but to the best of our knowledge these conversational agents have not been used to address the phenomena of repetitive questions.

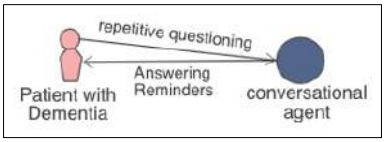

Fig. 1. Patient with Dementia and Conversational Agent Interaction

This model suggest that the caregiver assistant should be aware of the environment. For example, it needs to know if the informal caregiver left the house, if the PwD is moving around the house or if the PwD left the house so it can send a message or call the caregiver to alert the situation. Also, the awareness can be useful for pattern recognition in the future understanding when, where and why the PwD experiences those behaviors.

It is important to understand how the conversational agents will help informal caregivers so we can evaluate how helpful and useful the system is for the caregivers. The evaluation could be performed conducting interviews, questionnaires, and usability evaluation. It is important that the informal caregiver uses the caregiver assistant to relax and reduce the negative emotions mentioned by the caregivers in the interviews (Fig. 2). The conversational agents can provide some relaxation routines to the family caregivers and answering some of the PwD's repetitive questions reducing the frustration of the family.

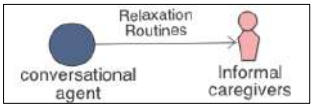

Fig. 2. Informal caregiver and conversational agent interaction

The incorporation of persuasive strategies such as personalization, challenges, tailoring can be an additional value (Fig. 3). These strategies can keep the caregivers and PwD using the conversational agent, because as researchers and developers we want to keep the attention of the users, but also, we want to change their attitudes and eventually impact in the behaviors.

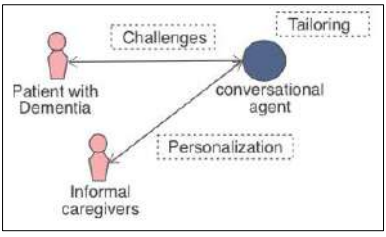

Fig. 4. Strategies in the use of conversational agent

Conversational agents can help informal caregivers provide personalized greetings and some specific clarification to the PwD every morning to start the day or at the end of the day. In specific cases the agent can play some greeting with the voice of the caregiver or with a different voice depending on their needs. Thus, we suggest the following persuasive strategies.

5.1.1 Personalization

The users should be able to personalize the system according to their preference, including name, nickname, gender of voice. Caregivers can modify to its convenience some of the features of the conversational agents.

5.1.2 Challenges

In these cases, the conversational agents could be a useful tool by reading audiobooks, playing music, or other recreational activities such as games.

5.1.3 Tailoring

Informal caregivers can let the conversational agent know that they are going to go out or when they are going to get the meal, so when the PwD asks about it, the conversational agent will answer what is happening or will happen next. The agent can then suggest calling the caregiver or someone else if the PwD feels anxious.

6. Conclusion and future work

In this paper, we presented a qualitative study based on interviews to seven informal PwD caregivers from two different countries. The results suggest that the most challenging behaviors of the PwD are forgetting things and repetitive questioning. Caregivers have faced these problems and have developed different strategies to deal with them such as: follow the conversation, go out with the PwD to distract him/her, ask questions for the patient to reflect on where he is and who he is with, and for mental stimulation.

In all the informal caregivers' interviews, the caregivers expressed that taking care their loved ones trigger negative emotions like frustration and stress.

Based on these results, we suggest a model that describes the ecosystem of PwD assistance and how a caregiver assistant can support informal caregivers with some activities. The conversational agents can be an effective tool to support PwD answering repetitive questions, monitoring and evaluating the PwD. The evaluation could be useful for the assessment of disease progression. These patterns can detect specific situations in specific days or with specific context.

The proposal includes some persuasion strategies can help on change the PwD attitude and eventually change their behaviors.

This study suggests the next lines for future work:

1) Design an ontology using the conceptual model presented in this paper. The conversational agents can be developed taking advantage of such ontology.
2) Design a conversational agent that implements the strategies presented in this model.
3) Use a commercial conversational agent such as Alexa, Siri or Cortana to determine the acceptance and impact on PwD and informal caregivers.

The proposed model can be a basis for researchers and developers for implementing strategies and suggestions to support PwD and informal caregivers.

References

[1] Boyd H.C., Evans N.M. et al. Using simple technology to prompt multistep tasks in the home for people with dementia: An exploratory study comparing prompting formats. Dementia, vol. 16, issue 4, 2017, pp. 424-442.

[2] Kelly P.A., Cox L.A. et al. The effect of PARO robotic seals for hospitalized patients with dementia: A feasibility study. Geriatric Nursing, vol. 42, issue 1, 2021, pp. 37-45.

[3] Wójcik D., Szczechowiak K. et al. Informal dementia caregivers: Current technology use and acceptance of technology in care. International Journal of Environmental Research and Public Health, vol. 18, issue 6, 2021, pp. 1-14.

[4] Guan C., Bouzida A., Oncy-Avila R.M. Taking an (embodied) cue from community health: Designing dementia caregiver support technology to advance health equity. In Proc. of the Conference on Human Factors in Computing Systems, 2021, article no. 655, 16 p.
[5] Huelat B., Pochron S.T. Stress in the Volunteer Caregiver: Human-Centric Technology Can Support Both Caregivers and People with Dementia. Medicina (B Aires), vol. 56, issue 6, 2020, article no. 257, 17 p.
[6] Ruggiano N., Brown E.L et al. Chatbots to support people with dementia and their caregivers: Systematic review of functions and quality. Journal of Medical Internet Research, vol. 23, issue 6, 2021, article no. e25006, 11 p.
[7] Chung K. Elderly Users' Interaction with Conversational Agent. In Proc. of the 7th International Conference on Human-Agent Interaction, 2019, pp. 277-279.
[8] Houben M., Brankaert R. et al. Foregrounding everyday sounds in dementia. In Proc. of the 2019 ACM Designing Interactive Systems Conference, 2019, pp. 71-83, 2019.
[9] Brown E.L., Ruggiano N. et al. Smartphone-Based Health Technologies for Dementia Care: Opportunities, Challenges, and Current Practices. Journal of Applied Gerontology, vol. 38, issue 1, 2019, pp. 73-91.
[10] Yang L., Ye H., Sun Q. Family caregivers' experiences of interaction with people with mild-to-moderate dementia in China: A qualitative study. International Journal of Nursing Practice, vol. 27, issue 4, 2021, pp. 1-8, 2021, article no. e12892, 8 p.
[11] Cha I., Kim S. et al. Exploring the Use of a Voice-based Conversational Agent to Empower Adolescents with Autism Spectrum Disorder. In Proc. of the 2021 Conference on Human Factors in Computing Systems, 202, article no. 42, 15 p.
[12] Choi Y.K., Thompson H.J., Demiris G. Internet-of-Things Smart Home Technology to Support Aging-in-Place: Older Adults' Perceptions and Attitudes. Journal of Gerontological Nursing, vol. 47, issue 4, 2021, pp. 15-21.
[13] Apergi L.A., Bjarnadottir M.V. et al. Voice Interface Technology Adoption by Patients With Heart Failure: Pilot Comparison Study. JMIR Mhealth Uhealth, vol. 9, issue 4, 2021, article no. e24646, 15 p.
[14] Kim S. Exploring How Older Adults Use a Smart Speaker–Based Voice Assistant in Their First Interactions: Qualitative Study. JMIR Mhealth Uhealth, vol. 9, issue 1, 2021, article no. e20427, 12 p.
[15] Ault L., Goubran R. et al. Smart home technology solution for night-time wandering in persons with dementia. Journal of Rehabilitation and Assistive Technologies Engineering, vol. 7, 2020, article no. 205566832093859, 8 p.
[16] Smith E., Sumner P. et al. Smart speaker devices can improve speech intelligibility in adults with intellectual disability. International Journal of Language & Communication Disorders, vol. 56, issue 3, 2021, pp. 583-593.
[17] Li J., Maharjan B. et al. A personalized voice-based diet assistant for caregivers of Alzheimer disease and related dementias: System development and validation. Journal of Medical Internet Research, vol. 22, issue 9, 2020, article no. e19897, 11 p.
[18] Russo A., D'Onofrio G. et al. Dialogue Systems and Conversational Agents for Patients with Dementia: The Human–Robot Interaction. Rejuvenation Research, vol. 22, issue 2, 2019, pp. 109-120.
[19] Navarro R.F., Rodríguez M.D., Favela J. Intervention tailoring in augmented cognition systems for elders with dementia. IEEE Journal of Biomedical and Health Informatics, vol. 18, issue 1, 2014, pp. 361-367.
[20] Schumacher C., Dash D. et al. A qualitative study of home care client and caregiver experiences with a complex cardio-respiratory management model. BMC Geriatrics, vol. 21, issue 1, 2021, article no. 295, pp. 1–11, 2021, 11 p.
[21] Braun V., Clarke V. Using thematic analysis in psychology. Qualitative Research in Psychology, vol. 3, issue 2, 2006, pp. 77-101.
[22] Connors M.H., Seeher K. et al. Dementia and caregiver burden: A three-year longitudinal study," International Journal of Geriatric Psychiatry, vol. 35, issue 2, 2020, pp. 250-258.

Information about authors / Информация об авторах

Samantha JIMÉNEZ, Doctor of Science, Full Professor. Research interests include Software Engineering, Usability, Educational Technology, Human-Computer Interaction.

Саманта ХИМЕНЕС, кандидат наук, профессор. Область научных интересов включает разработку программного обеспечения, удобство использования, образовательные технологии, взаимодействие человека и компьютера.

Jesús FAVELA, Ph.D., Professor, Computer Science Department. Research interests: Ubiquitous Computing, Human-Computer Interaction, Medical Informatics.

Хесус ФАВЕЛА, кандидат наук, профессор кафедры компьютерных наук. Область научных интересов: повсеместные вычисления, взаимодействие человека и компьютера, медицинская информатика.

Ángeles QUEZADA, Ph.D. in Computer Science. Research interests: Neural Networks, Pattern Recognition, Fuzzy Logic, Neural Networks and Artificial Intelligence, Computational Intelligence, Fuzzy Clustering, Computer Vision, Autism Spectrum Disorders, Autism.

Анхелес КЕСАДА, кандидат компьютерных наук. Научные интересы: нейронные сети, распознавание образов, нечеткая логика, нейронные сети и искусственный интеллект, вычислительный интеллект, нечеткая кластеризация, компьютерное зрение, расстройства аутистического спектра, аутизм.

Raj RAMACHANDRAN, Lecturer in Computer Science. Research interests: speech to text, software engineering, requirement engineering, user acceptance, philosophy in technology.

Радж РАМАЧАНДРАН, преподаватель компьютерных наук. Научные интересы: преобразование речи в текст, разработка программного обеспечения, инженерия требований, приемлимость для пользователей, философия в технологии.

Reyes JUÁREZ-RAMÍREZ, Doctor of Computer Science, Full Professor. Research interests include software Engineering, software uncertainty estimation, and human-computer interaction.

Рейес ХУАРЕС-РАМИРЕС, кандидат компьютерных наук, профессор. Область научных интересов включает разработку программного обеспечения, оценку неопределенности программного обеспечения и взаимодействие человека и компьютера.

DOI: 10.15514/ISPRAS-2023-35(1)-3

How COVID-19 Pandemic affects Software Developers' Wellbeing, and the New Trends in Soft Skills in Working from Home

[1] *R. Juárez-Ramírez, ORCID: 0000-0002-5825-2433 <reyesjua@uabc.edu.mx>*
[1] *C.X. Navarro, ORCID: 0000-0002-7220-7006 <cnavarro@uabc.edu.mx>*
[1] *G. Licea, ORCID: 0000-0002-7304-8051 <glicea@uabc.edu.mx>*
[2] *S. Jiménez, ORCID: 0000-0003-0938-7291 <samantha.jimenez@tectijuana.edu.mx>*
[3] *V. Tapia-Ibarra, ORCID: 0000-0002-0501-8600 <veronica.tapia@leon.tecnm.mx>*
[4] *C. Guerra-García, ORCID: 0000-0002-9290-6170 <cesar.guerra@uaslp.mx>*
[4] *H.G. Perez-Gonzalez, ORCID: 0000-0003-3331-2230 <hectorgerardo@uaslp.mx>*

[1] *Universidad Autónoma de Baja California,*
Tijuana, México, 22390
[2] *Instituto Tecnológico de Tijuana,*
Tijuana, México, 22424
[3] *Instituto Tecnológico de León,*
León, Guanajuato, México, 37290
[4] *Universidad Autónoma de San Luis Potosí,*
San Luis Potosí, SLP, México, 78000

Abstract. The coronavirus COVID-19 swept the world in early 2020, working from home was a necessity. In the software industry, thousands of software developers began working from home, many did so on short notice, under difficult and stressful conditions. The emotions of developers can be affected by this situation. On the other hand, some well-known soft skills have been emphasized as required for working remotely. Software engineering research lacks theory and methodologies for addressing human aspects in software development. In this paper, we present an exploratory study focused on the developers' wellbeing during pandemic, expressed as emotions, and the perceptions of the level in which soft skills are practiced/required in the working from home mode. The results show that high percent expressed to experience positive emotions, however, a portion of respondents expressed to feel negative emotions. In the case of soft skills, some of them are revealed as practiced in high level in working from home, but still there is not consensus.

Keywords: COVID-19; working remotely; software development; developers' wellbeing; soft skills

For citation: Juárez-Ramírez R., Navarro C.X., Licea G., Jiménez S., Tapia-Ibarra V., Guerra-García C., Perez-Gonzalez H.G. How COVID-19 Pandemic affects Software Developers' Wellbeing, and the New Trends in Soft Skills in Working from Home. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 35-56. DOI: 10.15514/ISPRAS-2023-35(1)-3

Влияние пандемии COVID-19 на психофизическое состояние разработчиков программного обеспечения и новые тенденции в области гибких навыков при работе из дома

[1] *Р. Хуарес-Рамирес, ORCID: 0000-0002-5825-2433 <reyesjua@uabc.edu.mx>*
[1] *К.К. Наварро, ORCID: 0000-0002-7220-7006 <cnavarro@uabc.edu.mx>*
[1] *Г. Лисеа, ORCID: 0000-0002-7304-8051 <glicea@uabc.edu.mx>*
[2] *С. Хименес, ORCID: 0000-0003-0938-7291 <samantha.jimenez@tectijuana.edu.mx>*
[3] *В. Тапиа-Ибарра, ORCID: 0000-0002-0501-8600 <veronica.tapia@leon.tecnm.mx>*
[4] *С. Герра-Гарсия, ORCID: 0000-0002-9290-6170 <cesar.guerra@uaslp.mx>*
[4] *Г.Г. Перес-Гонсалес, ORCID: 0000-0003-3331-2230 <hectorgerardo@uaslp.mx>*

[1] *Автономный университет Нижней Калифорнии (UABC), Мексика, 22390, Нижняя Калифорния, Тихуана*
[2] *Тихуанский технологический институт, Мексика, 22414, Нижняя Калифорния, Тихуана*
[3] *Леонский технологический институт, Мексика, 37290, Гуанахуато, Леон*
[4] *Автономный университет Сан-Луис-Потоси, Мексика, 78000, SLP, Сан-Луис-Потоси*

Аннотация. В начале 2020 года коронавирус COVID-19 распространился по всему миру, и работа на дому стала необходимостью. В индустрии программного обеспечения тысячи разработчиков программного обеспечения начали работать из дома, многие сделали это в короткие сроки, в сложных и напряженных условиях. Эта ситуация могла воздействовать на эмоции разработчиков. С другой стороны, стало понятно, что для удаленной работы необходимы некоторые хорошо известные навыки межличностного общения. Исследованиям в области программной инженерии не хватает теории и методологий для рассмотрения человеческих аспектов в разработке программного обеспечения. В этой статье мы представляем предварительное исследование, посвященное психофизическому состоянию разработчиков во время пандемии, выраженному в испытываемых эмоциях и понимании того уровня, на котором используются/требуются гибкие навыки при работе из дома. Результаты показывают, что эмоции большей части разработчиков были положительными, однако часть респондентов выразила отрицательные эмоции. Что касается гибких навыков, некоторые из них оказываются очень востребованными при работе из дома, но единого мнения нет.

Ключевые слова: COVID-19; удаленная работа; разработка программного обеспечения; психофизическое состояние разработчиков; гибкие навыки

Для цитирования: Хуарес-Рамирес Р., Наварро К.К., Лисеа Г., Хименес С., Тапиа-Ибарра В., Герра-Гарсия С., Перес-Гонсалес Г.Г. Влияние пандемии COVID-19 на психофизическое состояние разработчиков программного обеспечения и новые тенденции в области гибких навыков при работе из дома. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 35-56. DOI: 10.15514/ISPRAS–2023–35(1)–3

1. Introduction

According to a World Economic Forum (WEF) forecast of employment trends [1], remote work is the future; it is one of the biggest drivers of transformation in workplaces around the world, with around 40% of full-time employees already used to some form of remote work/telecommuting in the USA and Europe. Also, the WEF stated that [1]: "On average, by 2020, more than a third of the desired core skill sets of most occupations will be comprised of skills that are not yet considered crucial to the job today, according to our respondents." Gartner [2] stated that as digitalization moves from an innovative trend to a core competency, enterprises need to deliver products, attract, and retain talent.

According to specialized forums on Internet [3-6], recently, there are multiple options for hiring software developers in the marketplace; hiring remote dedicated develop is one of them. A company needs to hire remote dedicated developers when it feels and notices the following signs [7], among others: a) Struggle for desired skills in their teams; and b) Need of faster delivery of the project.

The recent situation of COVID-19 spread represents a starting of disruption in skills for remote work, as it suggested in Internet forums [8-10]. In this time, employees need to adopt or develop habits for working, which represent abilities and skills to possess. Professional software developers and practitioners are now working from home (WFH), and they should possess those skills.

Some studies present evidence of the "dilemma" about if software developers like to work from home in formal jobs [11-17], its benefits and challenges; this is also well expressed in popular forums on Internet [5, 18-20]. The ability to work remotely has been long touted as a prime perk by recruiters and hiring managers, especially in industrialized countries, contracting outsourcing services; furthermore, even when some developers like to work remotely, many others do not. However, data from a Survey [5] of the well-known blog called Stack Overflow Developer suggests that before pandemic a slight majority of developers (57.90 %) want to work in the office as opposed to home (33.20 %), while a mere 8.80% want to work out of another place such as a coworking space or café.

Working remotely has advantages and disadvantages, as expressed in Internet forums [21, 22], some of them are cited here:

Advantages: Reduced cost of software development; increased productivity; full flexibility of work; easy to scaleup the software development teams; faster to decrease/increase the manpower; faster delivery of projects; effective use of diverse time-zones; and access to a large pool of talent across the globe.

Disadvantages: Reduced control over the remote team; communication-related problems; reduced human connection; cross-culture gap; issues related to the quality of work; and increased risk of project failure due to reliability matters.

Traditionally, the productivity and performance of software developers and other jobs [23] is assessed and measured, as indicators of company capacities to produce software, but software developers are not machines, they are human beings; this fact stablishes a connection between cognitive and psychological aspects [24]. There are several studies that describe the connection between developers' wellbeing and their performance and productivity as it is cited in [25-27]. However, in the software industry there is a lack of attention addressing the human aspects in software development [28, 29]. Software systems are designed and used by humans; and the human being is characterized, among other things, by emotions. Given this fact, the process of designing and developing software systems is, like any other facet in our lives, driven by emotions [24].

By today, during the COVID-19 pandemic, in the software industry, thousands of software developers began working from home, many did so on short notice, under difficult and stressful conditions. Companies that fire for remote software developer jobs are [30]: *Amazon*, *Facebook*, *CrowdStrike*, *GitHub*, *Oracle*, *Slack*, *Twilio*, *Twitter*, *Spotify*, *SAP*, *VMware*, *Salesforce*. Several studies have concluded that remote developers are happier, more productive, and focused when working from home as is stated in [31, 32]. So, employees can harness flexible hours to optimize their schedule and work when they feel the most productive if they deliver work on time.

As we mentioned early, there are advantages and disadvantages of working from home. The elapsed period of pandemic has shown that some precise advantages are flexible schedule, increasing productivity, and work-life balance. However, working from home has some challenges, such as the following requirements: self-management and discipline, effective communication, and teamwork synchronization. So, it is time to analyze the wellbeing of developers expressed in a set of emotions working in the pandemic lockdown and the new normality. Furthermore, it is important to revisit the set of soft skills that commonly are demanded in software development, and especially in the working from home mode because it is a new fashion derived from COVID-19 pandemic.

In a previous version on this paper ("How COVID-19 Pandemic affects Software Developers' Wellbeing: An Exploratory Study in the West Border Area of Mexico-USA" [33]), we presented the first part of this exploratory study with a survey applied to practitioners in the west border area of Mexico-USA, emphasizing the conditions of working from home, the support from the companies, and the set of emotions they have experienced. In this extended version, we add a descriptive analysis about the most common skills of software developers, emphasizing those soft skills that participants considered more required/practiced in working from home mode. Also, we include the corresponding background.

The rest of the paper is organized as follows. Section 2 contains a background describing what emotions are, and the new trends in soft skills requirements. Section 3 describes some related work. Section 4 describes the methodology used. Section 5 contains the results in a descriptive way. Section 6 contains the discussion. Finally, section 7 has the conclusions and future work.

2. Background

2.1 The stereotype of software developers and emotions

There are many stereotypes about software developers. A common trope is the idea that they are emotionless, completely rational robots.

Software engineering research lacks theory and methodologies for addressing human aspects in software development [28, 29, 34]. Software development tasks are undertaken through cognitive processing activities. Affects (emotions, moods, feelings) have a linkage to cognitive processing activities and the productivity of individuals. Software engineering research needs to incorporate affect measurements to valorize human factors and to enhance management styles.

In [35], a theoretical framework for supporting emotions in the context of workplace was presented, which is the Affective Events Theory (AET). In AET, the work environment settings (e.g., the workplace, the salary, promotion opportunities, etc.) mediate work events that cause affective reactions, which are interpreted according to the individuals' disposition. Affective reactions then influence work-related behaviors, including emotions.

Emotions have been defined as the states of mind that are raised by external stimuli and are directed toward the stimulus in the environment by which they are raised [36]. However, several definitions have been produced for this term [37], and no consensus within the literature has been reached. For practical aspects, this term has been taken for granted and is often defined with references to a list, e.g. anger, fear, joy, surprise [38].

Other related terms are moods and feelings. Moods have been defined as emotional states in which the individual feels good or bad, and either likes or dislikes what is happening around him or her [39]. Feelings have been defined as the conscious subjective experience of emotions [40]. One of the most related terms is happiness, which has been defined as the emotional evaluation of life measured as the sum of the frequency of emotions in a timespan [41-43].

In our study, we are focused on a set of emotions, and we are going to introduce the way they are considered in the software development context. Software companies nowadays often aim for flourishing happiness among developers. There are several ways to make software developers happy, for instance [44]: Perks, playground rooms, free breakfast, remote office options, sports facilities near the companies, etc. Graziotin et al. present several studies [44-49], which relate developers' happiness with productivity, solving problems in a better way, better performance, and so on.

In [29], a set of emotions experienced by programmers are presented: a) Positive: Happy, Enthusiastic, Pleased, Optimistic, Enjoying, Content; b) Negative: Depressed, Frustrated, Angry, Disgusted, Unhappy, Disappointed. These emotions could occur during programming and affect productivity.

2.2 The new trends in soft skills

In a software developers' working career, they consider whether their level of hard and soft skills is appropriate. Employers know that professional and technical skills alone cannot help to achieve organizational goals and maintain company competitiveness. The qualities preferred for software developers include technical knowledge within the field or the position as well as soft skills [50]. Currently, employers consider that to achieve the company's goals, the employees must master the technical and professional knowledge and superstructure of soft skills. This means that soft skills are becoming critical to the success of a company.

A hard skill is the ability to conduct a particular type of task or activity using technical knowledge and experience, while a soft skill relates to a person's relationships with others and can be applied widely [50]. Hard skills involve technical knowledge of programming languages, compilers, base software systems [51-56], and other specific technical knowledge such as effort estimation, and so on [57]. Hard skills are acquired in formal courses and training.

In the case of soft skills, many definitions can be found in the literature. In general terms, soft skills are personal quality attributes divided in interpersonal, intrapersonal, and high-order thinking (cognitive) skills [55, 58], which are characterized as self-identity, self-control, social skills, communication, and mindset [58-60]. Soft skills characterize certain career attributes that individuals may possess like the ability to work in a team, communication skills, leadership skills, customer service, emotional intelligence, and problem-solving skills [61]. Soft skills are personal qualities, attributes, or the level of commitment of a person setting him or her apart from other individuals who may have similar skills and experience [62]. They are the intangible, non-technical, personality-specific skills that characterize a person as a leader, facilitator, mediator, and negotiator [50].

Much formal and informal literature cite the importance of the programmer role [63], and the skills or attributes that programmers and software developers should have [64]. Next, we summarize some proposals for soft skills, which are appropriate for software developers: creativity [65-68], critical thinking (problem solving, analysis) [67, 68], self-learning, reading and comprehension, interpretation, inference, explanation, open-mindedness, self-regulation [67], effective communication, effective cooperation [65, 69], engagement, commitment, teamwork [61, 68], leadership skills, customer service, emotional intelligence [61], motivation [70], tasks identification, planning and scheduling, conflict resolution [68].

In the last six years, the WEF [1] is leading in the looked at current employment, skills, and workforce strategies to identify the top ten skills everyone will need in the fourth industrial revolution. The report compares the shift in the soft skills needed to succeed in maintaining a job. In Table 1 we can see how the top ten skills have shifted between 2015 and 2020 [1, 50].

As we can see, *complex problem* solving continue in the top by 2020. *Critical thinking* moved forward two places, going to the second place. *Creativity* moved from the tenth place in 2015 to the third in 2020. *Emotional intelligence* appears as a new skill, at sixth place in 2020. *Active listening* disappears in the list of 2020.

Table 1. The shift in the top ten soft skills in 2015 and 2020

Top 10 Soft-skills in 2015	Top 10 Soft-skills in 2020
1. Complex Problem Solving	1. Complex Problem Solving
2. Coordinating with Others	2. Critical Thinking
3. People Management	3. Creativity
4. Critical Thinking	4. People Management
5. Negotiation	5. Coordinating with Others
6. Quality Control	6. Emotional Intelligence
7. Service Orientation	7. Judgment and Decision Making
8. Judgment and Decision Making	8. Service Orientation

9. Active Listing	9. Negotiation
10. Creativity	10. Cognitive Flexibility

3. Related work

In [71], a study is presented, which investigates the effects of the COVID-19 pandemic on developers' wellbeing and productivity. These authors used a questionnaire survey created mainly from existing, validated scales and translated into 12 languages. This work oriented the wellbeing to emotional status. The authors stated that individuals' wellbeing while working remotely is influenced by their emotional stability (that is, a person's ability to their control emotions when stressed). This proposal is supported by the suggestion of [72]. The main results presented in [71] are: (1) the pandemic has had a negative effect on developers' wellbeing and productivity; (2) productivity and wellbeing are closely related; (3) disaster preparedness, fear related to the pandemic and home office ergonomics all affect wellbeing or productivity. In general terms, this study reports that the COVID-19 pandemic has not been good for emotional stability, as it also supported by [73].

In [74], the Construx Software company presented a study, where surveyed software professionals to determine the effect that working from home during the COVID-19 pandemic is having on software development. The survey explored changes in communication and the impact on individuals, on teamwork, on leaders' ability to lead, and on specific technical practices. The study presents long-term recommendations for WFH based on survey findings. Next, we highlight some of them: (1) recommendations for individuals, oriented to orchestrate a tech infrastructure at home, workday time scheduling, and disposing time space for personal and family issues; (2) recommendations for teams, oriented to organize synchronized teams' work, stablishing clear expectations and communicate them to the team, and enabling efficient communication channels and practices; and (3) recommendations for leaders, oriented to develop remote-leadership skillset, maintain support to remote teams, and consider human aspects such as emotional needs of team members and try to give support on it.

In [75], the authors presented a study to analyze and understand how a typical working day looks like when working from home during the pandemic and how individual activities affect software developers' wellbeing and productivity. Results suggested that the time software engineers spent doing specific activities from home was similar when working in the office. However, they also found some significant differences. An interesting finding was that the amount of time developers spent on each activity was unrelated to their well-being and perceived productivity. So, the authors concluded that working remotely is not per se a challenge for organizations or developers.

In [76], the authors presented a study, emphasizing on that COVID-19 pandemic has provoked an overnight exodus of developers that normally worked in an office setting to working from home. To find out how developers and their productivity were affected, the authors distributed two surveys to understand the presence and prevalence of the benefits, challenges, and opportunities to improve this special circumstance of remote work. One of the main findings is that there is a dichotomy of developer experiences influenced by many different factors, which for some are a benefit, while for others a challenge. For example, a benefit for some was being close to family members, but for others having family members share their working space and interrupting their focus, it was a challenge.

4. Methodology

4.1 The survey

The objective of the survey is to identify the level of positive and negative emotions that software developers experienced working from home during the COVID-19 pandemic. Furthermore, we are going to identify the perception of participants about which soft skills are practiced in the working

from home mode. We designed the survey organized in the following sections: a) demographic, b) job position aspects, c) conditions of working from home (equipment, etc.), d) the impact of COVID-19 in working life, e) effectiveness of working from home strategies (including emotions), and f) programmer/developer attributes. The survey was presented in Google forms and applied by April 2021.

The set of emotions. Developing software systems is driven by emotions, as is stated in [24]. Programmers go through positive and negative emotions [29] in software development activities. Based on this, we considered a set of emotions, both positive and negative, supported by proposals extracted from [44, 45, 46, 47, 48, 49].

The set of soft skills. As we mentioned early, the WEF suggested a set of skills needed for the fourth industrial revolution [1, 50], as it is shown in Table 1. We adapted such proposal to the software development context considering intra and interpersonal attributes. We integrated a set of soft skills considering those suggested in [64-66, 68, 69] and our experience in real software projects with industry.

The survey has three important parts for this study: (1) the main difficulties faced in working from home during pandemic, 14 items (one per each difficulty) in a 5-point Likert scale; (2) emotions lived in full working from home mode during pandemic, 17 items (one per each emotion) in a 6-point Likert scale; (3) soft skills or attributes practiced in working from home during pandemic, 23 items (one per each soft skill) in a 4-point Likert scale. The consistency of the items was evaluated with Cronbach's alpha test, having the following results for the three sets: items of difficulties 0.794, acceptable; items of emotions 0.775, acceptable; items of soft skills 0.990, excellent.

4.2 The sample

This research involves a social study, so we considered graduates of the Computer Engineering undergraduate program from the Universidad Autónoma de Baja California, Tijuana campus. Due to the limitations to contact complete graduated classes, it was not feasible to do random sampling, so we used a non-probabilistic sample, that is, a convenience sample, considering a list of graduates with possibilities for contacting them. We did an invitation to 65 graduates who are working in companies in the Tijuana-USA border, including Silicon Valley and Seattle, WA. We included big companies worldwide known, considering that such level of companies could provide media needed to work from home. The invitation was sent to graduates through email or chat contact, encouraging them to answer in a period of 15 days. One kindly reminder was sent to 30% of the invited graduates before the deadline. Finally, 45 answers were collected.

The sample was 45 developers from companies of the west side of the Mexico-USA border: Baja California (Mexico) and California (USA). 62.2% of respondents work and live in Tijuana, 11.1% live in other cities close the border, while 26.6% developer work and live in the USA side. The gender, 98% male and 2% female; this proportion is common in software development practitioners. 53.0% of respondents live at family home, while 47.0% do not. About 60.0% are single.

About the work experience in industry, the distribution is as follows: Less than 1 year: 2.2%; 1 year: 13.3%; 2 to 3 years: 17.8%; 4 to 5 years: 22.2%; 6 to 7 years: 17.8%; 8 to 9 years: 11.1%; 10 or more years: 15.6%. A significant percent expressed to have 4 or more years (66.7%).

The respondents reported to perform a mix of activities related with software development. They emphasized to do software design. The percentages for activity are expressed as follows: Software design: 82.2%; Programming: 57.8%; Maintenance: 66.7%; System/requirements analysis: 28.9%; Code review: 28.9%; Project management: 17.8%; Manager 66.7%; Team Leader: 66.7%; Other: 11.1%; Administrator: 0%.

5. Results

The results are expressed in the followings terms: 1) equipment and infrastructure for working from home; 2) the satisfaction with the company' support; 3) worries about conditions derived from working from home; 4) difficulties faced; 5) experienced emotions; 6) preference for the working from home mode; 7) level of satisfaction with working from home, and 8) the soft skills practiced.

5.1 Equipment and infrastructure

The conditions of equipment and workspace available for working from home is shown in Fig. 1.

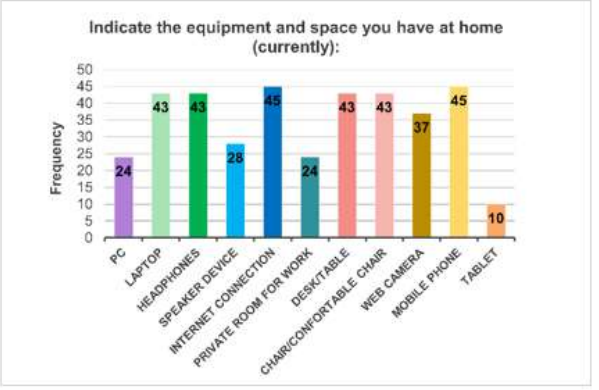

Fig. 1. Equipment in home office

The most basic equipment is possessed by almost all the respondents (laptop, headphones, Internet service, desk/table, comfortable chair, and mobile phone). However, important items required for working from home are not possessed by all the respondents, for instance, only 62.2% have speaker device, and only 53.3% have a private room for working. Speaker device is required for conversations and meetings. A private room for office is very recommendable for comfort.

5.2 Satisfaction with the company' support

The satisfaction with the company is expressed as follows (see Fig. 2, 3):

- "The general support you are getting from your Company to help you transition to taking your work from home": 77.7%.
- "The support you are getting from your Company to provide you equipment and tech for taking your work from home": 77.7%.
- "The communication they are getting from the Company about its ongoing responses to COVID-19, e.g. how long time to stay working from home, when to return back to office": 80.0%.
- "The communication you are getting from your Company about its ongoing responses to COVID-19, e.g. how long time to stay working from home, when to return back to office": 75.5%.
- "Encouragement to teamwork": 75.5%.
- "Promotion of using methodologies for working better": 68.8%. "Recognition of people, team, individual, effort/performance": 77.7%.
- "Informing protocols to visit office if it necessary": 71.1%.

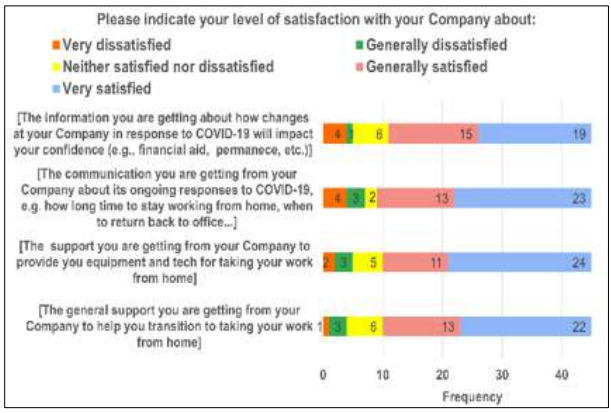

Fig. 2. Satisfaction with the company - Part I

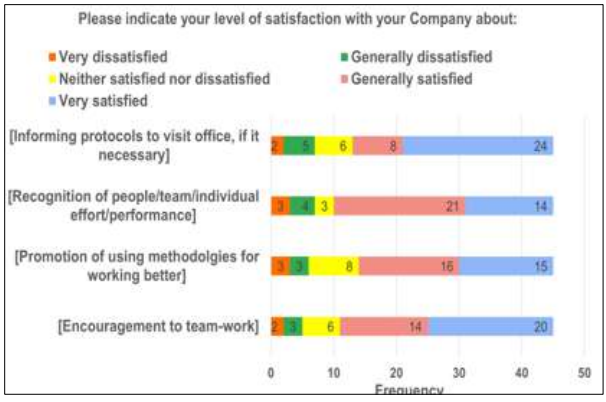

Fig. 3. Satisfaction with the company - Part 2

5.3 Worries

Considering the answers “sometimes” to “very often”, the percentages of worries are as follows (see Figs. 4, 5): Doing well in the Company now that many or all your work is from home: 51.1%; losing friendships and social connections now that work is from home: 46.6%; accessing and successfully using the technology needed for your work, from home: 42.5%; and having access to health care: 46.6%.

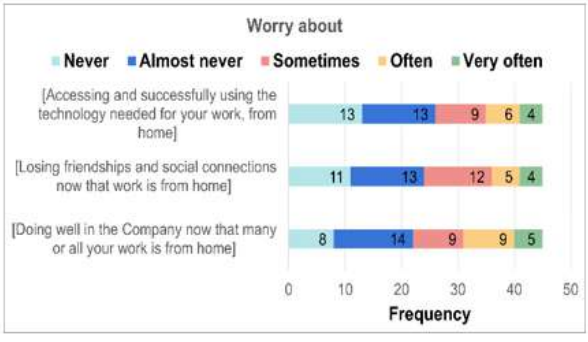

Fig. 4. Worries -Part 1

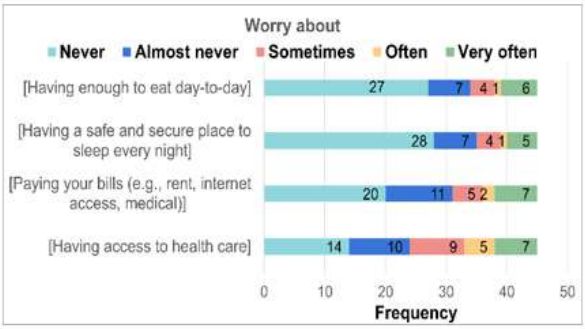

Fig. 5. Worries -Part 2

In general terms, respondents are more concerned with social interaction with work colleagues, and lack of social experience with other work colleagues, expressed as 55.5% (extracted from Figs. 7, 8) in both cases. Also, significant difficulties are to have access to team leader and team for face-to-face conversations (37.7%, 42.2%, respectively, extracted from Fig. 7). Another significant percent is the difficulty in keeping a regular work schedule (40.0% extracted from Fig. 6). Finally, a clear concern is anxiety with respect to COVID-19 risks (44.4%).

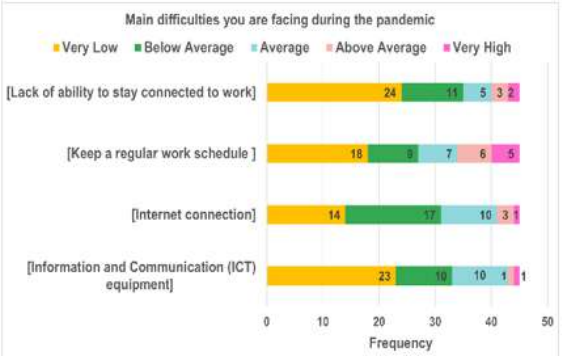

Fig. 6. Difficulties -Part I

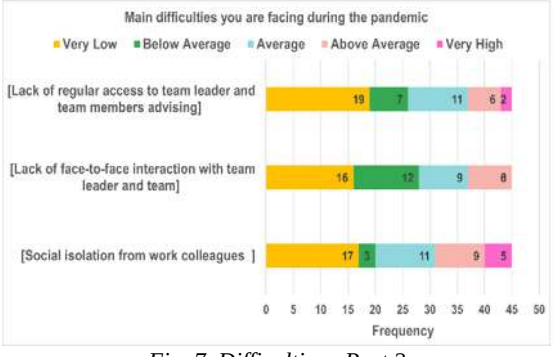

Fig. 7. Difficulties - Part 2

In general terms, respondents are more concerned with social interaction with work colleagues, and lack of social experience with other work colleagues, expressed as 55.5% (extracted from Figs. 7, 8) in both cases. Also, significant difficulties are to have access to team leader and team for face-to-face conversations (37.7%, 42.2%, respectively, extracted from Fig. 7). Another significant percent is the difficulty in keeping a regular work schedule (40.0% extracted from Fig. 6). Finally, a clear concern is anxiety with respect to COVID-19 risks (44.4%).

5.4 Faced difficulties

Seven difficulties are shown in Figs. 6, 7. 26.6% expressed to have difficulties with information and communication equipment. 31.1% expressed to have difficulties with Internet connection. 40.0% expressed to have difficulties in keeping a regular work schedule. 22.2% expressed to have difficulties to stay connected to work. 55.5% expressed to have difficulties with social isolation from work colleagues. 37.7% expressed to have lack of face to face-to-face interaction with team leader and team. 42.2% expressed to have lack of regular access to team leader and team members advising.

The second set of seven difficulties are shown in Figs. 8, 9. 35.5% expressed to lack of opportunities to request better performance of team leader or team. 55.5% expressed to have lack of social experience with other work colleagues. 24.4% expressed to have more difficulties to complete work. 24.4% expressed working from home experience not engaging. 44.4% expressed to have general anxiety with respect to COVID-19 risks. 22.2% expressed to have anxiety about working from home. 22.2% expressed to have economic concerns.

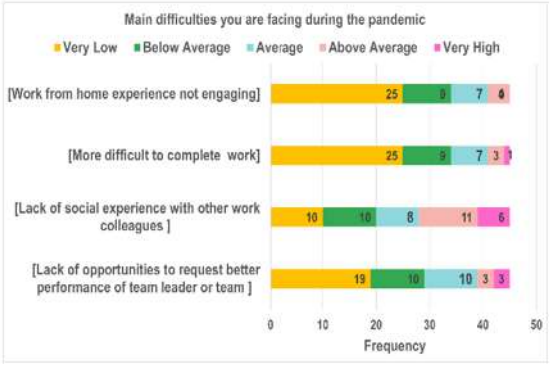

Fig. 8. Difficulties - Part 3

In general terms, respondents are more concerned with social interaction with work colleagues, and lack of social experience with other work colleagues, expressed as 55.5% (extracted from Figs. 7, 8) in both cases. Also, significant difficulties are to have access to team leader and team for face-to-face conversations (37.7%, 42.2%, respectively, extracted from Fig. 7). Another significant percent is the difficulty in keeping a regular work schedule (40.0% extracted from Fig. 6). Finally, a clear concern is anxiety with respect to COVID-19 risks (44.4%), Fig. 9.

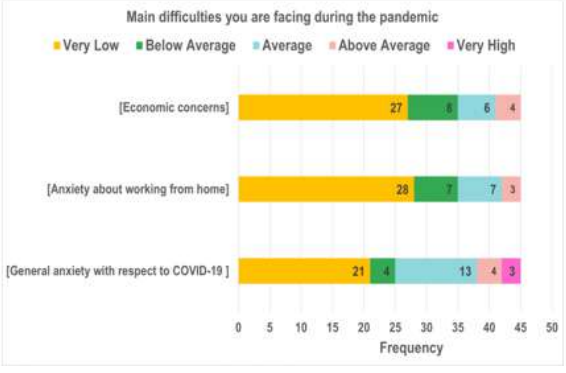

Fig. 9. Difficulties - Part 4

5.5 Emotions

A set of positive emotions are shown in Fig. 10. Considering answers from average and more, 88.8% of respondents are optimistic from average to very high, and 64.4% are optimistic above average

and more. 86.6% have serenity from average to very high, and 55.5% have serenity above average and more. 95.5% are happy from average to very high, and 64.4% are happy above average and more. 97.% have acceptance from average to very high, and 84.4% have acceptance above average and more. 93.3% have trust from average to very high, and 75.5% have trust above average and more. As we can see, the respondents present high levels of positive emotions.

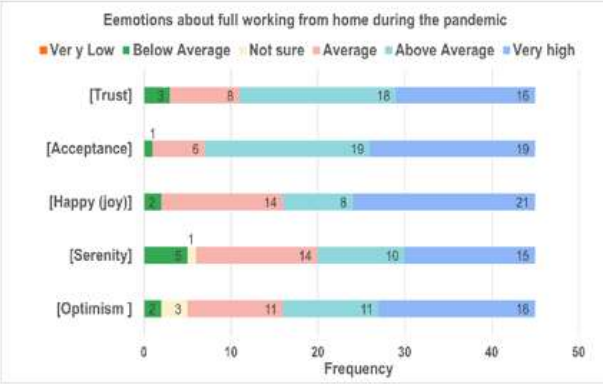

Fig. 10. Emotions –Part 1

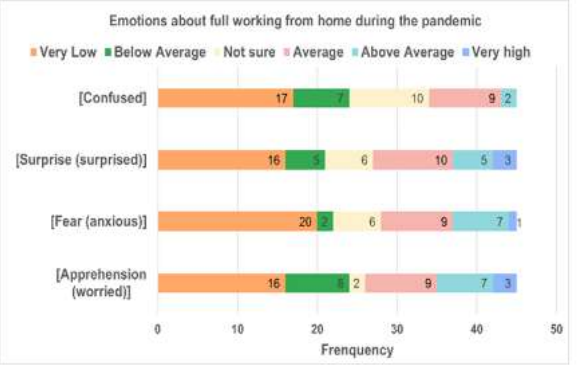

Fig. 11. Emotions –Part 2

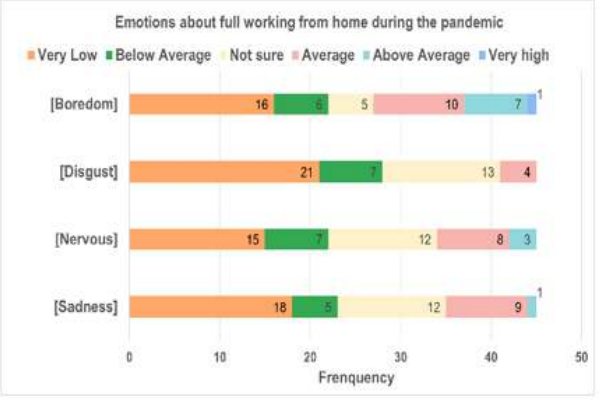

Fig. 12. Emotions –Part 3

In Figs. 11, 12 the responses about negative emotions are presented. Let see the results for negative emotions. 53.3% of respondents expressed not to have apprehension, while 42.2% expressed to have it in average and more, 4.4% said not to be sure. In the case of fear, 48.8% expressed not be afraid,

while 37.7% expressed to be afraid in average and more, and 13.3% said not be sure. About to be confused, 53.3% expressed they are not, while 24.4% expressed to be and 22.2% were not sure.

The next set of negative and positive emotions are shown in Fig. 13. In the case of anger, 57.7% expressed not having anger, while 8.8% expressed being angry, and 33.3% said they were not sure. 51.1% expressed not being annoyed, while 20% expressed having annoyance, and 28.8% said not be sure.

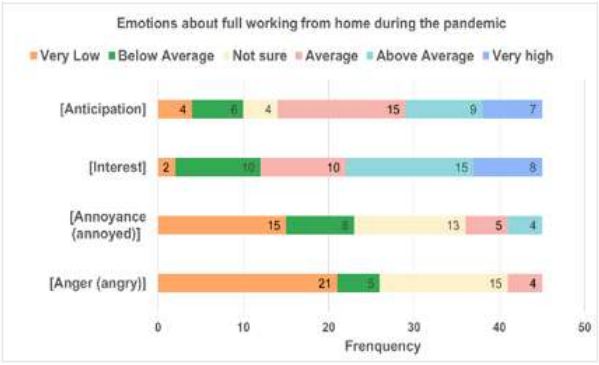

Fig. 13. Emotions –Part 4

5.6 Preference of working mode

The preference in shown in Fig. 14. 44.4% have high preference for working from home, while 35.5% have medium preference, and 20.0% have low preference. On the other side, 24.4% have high preference for working in the presence face-to-face mode, and 15.5% have medium preference, and 60.0% have low preference.

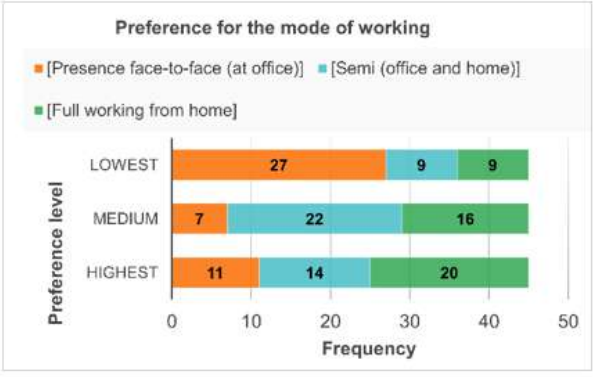

Fig. 14. Preference for working mode

5.7 Satisfaction with working from home

As we can see in Fig. 15, 91.0% shown satisfaction of working from home, considering responses satisfied and very satisfied. There is a significant percent of people which are satisfied of working from home, even when some of them have experienced negative emotions and have some difficulties as it is expressed in previous sections.

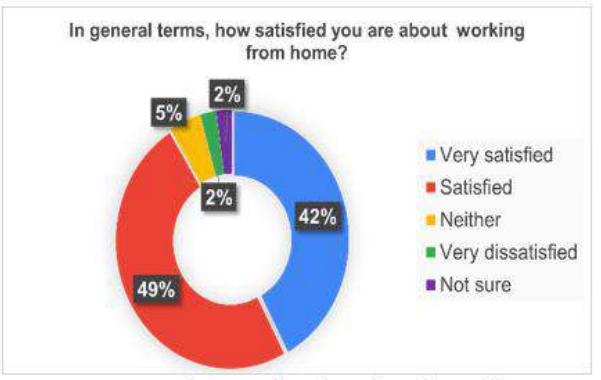

Fig. 15. Satisfaction of working from home during pandemic.

5.8 Attributes/soft skills of software developers practiced

We collected responses about the three modes: Presence face-to-face, semi-online mode, and working from home; however, in this section we only present the results of the participants opinion on how soft skills are practiced in the working from home mode. We present the results for a sample of skills, organized in terms of the level of practice in the working from home mode, so firstly we present the more practiced in this mode, and later the less practiced.

Table 2 shows the frequencies for each soft skill. We present the results in terms of the 4-point Likert scale: Nome (N), Low (L), Moderate (M), High (H). As we can see, most of the soft skills have significant frequencies from moderate to high, except those "negative" attributes or attitudes such as impatience, laziness, and hubris.

The skills more practiced in high level are self-learning (66.6%), reading & comprehension (64.4%), adaptability (64.4%), and good team player (64.4%), curiosity (60.0%), attention to details (57.7%), task & time management (57.7%), quick learning (55.5%), and quick learning outside programming (55.5%).

Table 2. Soft skills: Working from home (Frequency)

Soft Skills	**N**	**L**	**M**	**H**
Self learning	0	5	10	30
Reading & comprehension	0	4	12	29
Adaptability	1	4	11	29
Good team player	0	5	11	29
Curiosity	0	2	16	27
Attention to details	0	3	16	26
Task & time management	0	5	14	26
Quick learning	0	3	17	25
Quick learning outside prog.	0	4	16	25
Pair support	0	8	13	24
High end-user focus	0	3	19	23
Communication	0	3	19	23
Confidence	0	6	16	23
Supreme analysis	0	1	23	21
Deep & broad tech. capacity	1	4	21	19
Clear thinking	0	4	23	18
Impatience	3	7	18	17
Laziness	3	10	15	17
Hubris	5	13	12	15

As we can see, fortunately, impatience, laziness, and hubris were experienced in a lower level in the scales moderate and high, however, these kinds of attitudes were lived.

6. Discussion

6.1 Emotions and feelings

In the case of emotions, there is a mix of perceptions. High percent of respondents expressed to experience positive emotions in good level, such as optimism, serenity, happiness, acceptance, trust, interest, and anticipation. This is nearing the conclusion made by [75], which stated that "working remotely is not per se a challenge for organizations or developers." In this case, especially developers shown good level of wellbeing in terms of positive emotions.

On the other hand, significant percent expressed not having negative emotions in considerable level, such as apprehension, fear, confused, anger, and annoyance. However, part of the respondents expressed to experience some negative emotions in a significant level in average and more, such as sadness (22.2%), nervous (24.4%), and boredom (40.0%).

6.2 Skills/attributes practiced

The bigger consensus is 66.6% of *self-learning* for working from home compared to 44.4% for presence face-to-face mode. *Adaptability* is the second one with 64.4% for working from home compared to 53.3% for presence face-to-face mode, as the same as reading and comprehension with 64.4% for working from home compared to 40.0% for presence face-to-face mode. *Curiosity* is in the third place with 60.0% for working from home compared with 33.3% for presence face-to-face mode. *Time and task management* is in fourth place with 57.7% for working from home compared to 48.8% for presence face-to-face mode. These results allow us to suggest these four skills as the more practiced in the new fashion of working from home.

On the other hand, 57.7% respondents considered that *quick learning* is practiced in high level in presence face-to-face mode compared to 55.5% in working from home mode. In the case of *deep and broad technical capacity* 46.6% of respondents considered that this skill is practiced in high level in presence face-to-face mode, while 42.2% for working from home mode. The respondents considered that *clear thinking* is more practiced in high level in presence face-to-face mode (44.4%), compared to working from home (40.0%). These results suggest that respondents considered that they are more active and challenged in presence face-to-face mode.

The respondents considered that *good team player* and *pair support* are mode practiced in presence face-to-face mode than in working from home, which corresponds with the perceptions on worries and difficulties discussed in early sections. These skills correspond to the sixth ranked skill in [1, 50], "*Coordinating with Others"*.

In the case of *high end-user focus*, the respondents considered it is practiced in high level in the working from home mode with 51.1%, as same as in the presence face-to-face mode. This skill corresponds to "Service Orientation" considered in the eighth place by [1, 50]. *Communication* is also considered as practiced in the same level in both modes, with 51.1%.

Analyzing the negative moods, *impatience*, *laziness,* and *hubris* are slightly more practiced in high level in working from home mode compared to presence face-to-face. From these results we can deduce that some negative moods are more lived in the virtuality. In the case of *confidence*, it is practiced in the same level for both modes; this is a positive result, which mean that virtuality does not reduce the confidence of software developers.

6.3 Comparing results with related work

In terms of emotions, [71] did not present a set of specific emotions, however, it covers well the topic of wellbeing in general terms; this is the same case for [73, 75]. Our study presents seventeen specific emotions, assessed in a 6-pint Likert scale, which means a real expression of how the respondents felt working from home during pandemic, having similarities with respect to the results presented in [29].

In the case of problems and difficulties faced by individuals and teams during pandemic, [74] reports implications in communication, teamwork and so on, however, this study does not present specific assessment of each difficulty. Furthermore, the consulted related work does not present specific assessment of soft skills or attributes practiced in working from home during pandemic. Our work presents the assessment of specific soft skills in a 4-point Likert scale, which represents a significant contribution on the acknowledgement that developers do about how human aspects are involved directly in software development activities.

7. Conclusions and future work

Working remotely have been announced some years ago by the WEF, for becoming to be a fashion by 2020. The preference of developers was divided as it is shown by some studies such as [5], but the COVID-19 pandemic forced to go home and working remotely. Infrastructure of ICT, office conditions, time management and some other aspects had to be considered. Also, the wellbeing of developers is important.

In this paper, firstly, we have presented a descriptive analysis of the wellbeing of software developers working from home during the COVID-19 pandemic. The study was focused on a set of emotions experienced by developers, under some conditions working from home.

With respect to the worries imposed by remotely working, the respondents expressed three main concerns: a) lack of social experience with other work colleagues; b) difficulties with social isolation from work colleagues; and c) social interaction with work colleagues, team leader and team. This is evidence of the importance of the social part in the software development process and during pandemic working from home.

On the other hand, the results shown that a significant percent of developers experienced positive emotions in high level, and not having negative emotions in significant levels. However, there is a part of respondents that experienced negative emotions.

Even when not the total of respondents prefers working from home, 91.0% expressed to be satisfied and very satisfied of working from home during the pandemic. This fact could introduce a new trend of acceptance of working remotely even after the pandemic.

In the case of software developers' skills/attributes practiced, in general terms, the results expressed in an early section allow us to conclude that most of the skills proposed in [1, 50] are evidenced as practiced by software developers in the working from home mode during the elapsed period of pandemic. However, it is important to continue investigating the progress of adequacy of software developers to the new normality, to assess how all the skills are practiced, because in some cases, the respondents expressed to practice some skills more in the presence face-to-face mode than in working from home mode.

Our study introduces the consideration of human aspects of developers, which is a topic not commonly addressed in the software engineering research, so we are adding more evidence that emotions and soft skills play a significant role in the software development context.

Continuing in this line, we have formulated as future work:

- Make a distinction between "working from home" and "working from home during pandemic", characterizing conditions, skills required, and wellbeing experienced.
- How to improve social experience with other work colleagues in remote working? Facing the

limitations imposed by distance and technology.

- Exploring the new position/opinion of developers and companies to WFH after pandemic. Before, 57.90 % of developers preferred working at office, as it is expressed in [5], however, our study revealed that 91.0% are satisfied of working from home.
- Looking for a consensus about which soft skills are more required/practiced in the new normality of working from home. It is important to reach for the perceptions of both employers and software developers.

Our work has a limitation in the size of sample, so it convenient to address new studies in this topic, trying for reaching lager samples, to gather more consensus.

The new trends in working from home have direct impact in the academic environment; the new software engineers will need to have the skills required to work from home. It is important to prepare students (as future software workforce) to acquire the digital skills as well as soft skills [56], which have important impact on the working from home productivity and wellbeing of software developers [77]. Some challenges will be faced to convince current students to accept the full online learning (from home) to acquire the most required skills for working remotely [77, 78]: *self-management and discipline, time management, effective communication skills*, and so on. It is important to increasing the quality and sustainability of education at universities regarding the requirements of employers in terms of soft skills [50].

References / Список литературы

[1] World Economic Forum (WEF). The Future of Jobs Employment, Skills and Workforce Strategy for the Fourth Industrial Revolution. Global Challenge Insight Report. Online: January 2016, available at: http://www3.weforum.org/docs/WEF_Future_of_Jobs.pdf, accessed June 01, 2021.

[2] Gartner. Building the Digital Platform: Insights From the 2016 Gartner CIO Agenda Report. Online: 2016, available at: https://www.gartner.com/imagesrv/cio/pdf/cio_agenda_insights_2016.pdf, accessed June 01, 2021.

[3] Top Jobs to Work Remotely. Available at: https://remoteok.io/remote-work-statistics, accessed June 01, 2021.

[4] Where and How to Find Software Developers for Remote Work on US Projects in 2021. Available at: https://youteam.io/blog/where-to-find-software-developers-for-remote-work-on-us-projects/, accessed June 01, 2021.

[5] Stack Overflow. Developer Survey Results 2019: Where Do Developers Want to Work?. Available at: https://insights.stackoverflow.com/survey/2019, accessed June 01, 2020.

[6] EPAM Anywhere Business. How to Find and Hire Remote Developers When The Best Are Flying off the Shelf in a Day. Available at: https://anywhere.epam.com/business/how-to-find-and-hire-remote-developers, accessed June 01, 2022.

[7] Digneo G. 10 Essential Signs You Need to Hire a Remote Dedicated Development Team. Online: Nov 26, 2019, available at: https://runningremote.com/remote-development-team/, accessed June 01, 2021.

[8] Python Django Remote Jobs. Available at: https://www.ziprecruiter.com/Jobs/Python-Django-Remote, accessed June 01, 2021.

[9] Cote A. Remote Teams Guide: How to Manage Your Remote Software Development Team. Online: 15 April 2020, available at: https://www.freecodecamp.org/news/remote-teams-manager-guide/, accessed June 01, 2021.

[10] Doyle A. Important Job Skills for Software Engineers. Online: July 09, 2019, available at: https://www.thebalancecareers.com/software-engineer-skills-list-2062483, accessed June 01, 2021.

[11] Pounder C. Homeworking: No longer an easy option? Computers & Security, vol. 17, issue 1, 1998, pp. 27-30.

[12] Guo H. Special requirements for software process improvement applied in teleworking environments. In Proc. of the Second Asia-Pacific Conference on Quality Software, 2001, pp. 331-340.

[13] Herbsleb J.D. Global software engineering: The future of socio-technical coordination. In Proc. of the Future of Software Engineering Conference, 2007, pp. 188-198.

[14] Šmite D., Wohlin C. et al. Empirical evidence in global software engineering: a systematic review. Empirical Software Engineering, vol. 15, issue 1, 2010, pp. 91-118.

[15] Deshpande A., Sharp H. L. et al. Remote working and collaboration in agile teams. In Proc. Iof the nternational Conference on Information Systems, 2016, paper no. 12, 17 p.

[16] Mazzina A. What it means to be a remote-first company. Online: February 8, 2017, available at: https://stackoverflow.blog/2017/02/08/means-remote-first-company/, accessed June 1, 2021.

[17] Meyer A.N., Barr E.T. et al. Today Was a Good Day: The Daily Life of Software Developers. IEEE Transactions on Software Engineering, vol. 47, issue 5, 2021, pp. 863-880.

[18] Digneo G. Are Remote Workers Happier Than Office Employees? Available at: https://biz30.timedoctor.com/remote-workers-infographic/, accessed June 01, 2021.

[19] Wachal M. What is it like to work remotely as a software developer? Online: Sep 10, 2019, available at: https://blog.softwaremill.com/what-is-it-like-to-work-remotelyas-a-software-developer-1c0777e4a2a9, accessed June 01, 2021.

[20] Can a Software Developer Work from Home? Available at: https://www.ecpi.edu/blog/can-a-software-developer-work-from-home, accessed June 01, 2021.

[21] The Top 6 Challenges of Working Remotely And How You Can Overcome Them. Online: March 13, 2018, available at: https://www.timecamp.com/blog/2018/03/top-6-challenges-of-a-remote-work-and-how-to-overcome-them/, accessed June 01, 2020.

[22] Mishchenko A. Does Remote Work in Software Development Lead to Better Productivity? Available at: https://www.timedoctor.com/blog/remote-software-development/, accesed June 01, 2021.

[23] Turetken O., Jain A. et al. An Empirical Investigation of the Impact of Individual and Work Characteristics on Telecommuting Success. IEEE Transactions on Professional Communication, vol. 54, issue 1, 2011, pp. 56-67.

[24] Colomo-Palacios R., Casado-Lumbreras C. et al. Using the Affect Grid to Measure Emotions in Software Requirements Engineering. Journal of Universal Computer Science, vol. 17, issue 9, 2011, pp. 1281-1298.

[25] Miller C., Rodeghero P. et al. How was your weekend? Software Development teams working from home during COVID-19. In Proc. of the IEEE/ACM 43rd International Conference on Software Engineering, 2021, pp. 624–636.

[26] Miller C., Rodeghero et al. Survey Instruments for "How Was Your Weekend?" Software Development Teams Working from Home During COVID-19. In Companion Proc. of the IEEE/ACM 43rd International Conference on Software Engineering, 2021, pp. 223-223.

[27] Bao L., Li T. et al. How does working from home affect developer productivity? — A case study of Baidu during the COVID-19 pandemic. Science China. Information Science, vol. 65, issue 4, 2022, article no. 142102, 17 p.

[28] Graziotin D., Wang X., Abrahamsson P. Do feelings matter? On the correlation of affects and the self-assessed productivity in software engineering." Journal of Software: Evolution and Process, vol. 27, isssue 7, 2015, pp. 467-487.

[29] Wrobel M.R. Emotions in the software development process. In Proc. of the 6th International Conference on Human System Interactions (HSI), 2013, pp. 518-523.

[30] Understanding Remote Software Engineering Jobs. Online, February 24, 2021, available at: https://www.interviewkickstart.com/blog/remote-software-engineering-jobs, accessed April 1, 2022.

[31] Terminal. The State of Remote Engineering 2022 EDITION. Available at https://terminal.io/state-of-remote-engineering, accessed April 01, 2022.

[32] Owl Labs. 2021 State of Remote Work Report. Available at https://owllabs.com/state-of-remote-work/2021, accessed April 1, 2022.

[33] Juárez-Ramírez R., Navarro C.X. et al. How COVID-19 Pandemic affects Software Developers' Wellbeing: An Exploratory Study in the West Border Area of Mexico-USA. In Proc. of the 9th International Conference in Software Engineering Research and Innovation (CONISOFT), 2021 pp. 112-121.

[34] Serebrenik A. Emotional labor of software engineers. Proc. of the 16th BElgian-NEtherlands Software eVOLution Symposium, 2017, pp. 4-5.

[35] Graziotin D., Wang X., Abrahamsson P. How do you feel, developer? An explanatory theory of the impact of affects on programming performance. PeerJ Computer Science, vol. 1, issue 1, 2015, article no. e18, 32 p.

[36] Plutchik R., Kellerman H. Theory of emotion , vol. 1. Emotion: theory, research, and experience, Academic Press: London, 1980, 207 p.

[37] Kleinginna P.R., Kleinginna A.M. A categorized list of emotion definitions, with suggestions for a consensual definition. Motivation and Emotion, vol. 5, issue 4, 1981, pp. 345-379.

[38] Cabanac M. What is emotion? Behavioural Processes, vol. 60, issue 2, 2002, pp.69-83.

[39] Parkinson B., Briner R. et al. Changing moods: The psychology of mood and mood regulation, Addison-Wesley Longman, 1996, 264 p.

[40] VandenBos G.R., ed. APA dictionary of clinical psychology. American Psychological Association, 2012, 636 p.

[41] Diener E. Subjective well-being. Psychological Bulletin, vol. 95, issue 3, 1984, pp. 542-575.

[42] Dogan T., Totan T., Sapmaz F. The Role Of Self-esteem, Psychological Well-being, Emotional Selfefficacy, And Affect Balance on Happiness: A Path Model. European Scientific Journal, vol. 9, issue 20, 2013, pp. 31-42.

[43] Diener E., Wirtz D. W. et al. New Well-being Measures: Short Scales to Assess Flourishing and Positive and Negative Feelings. Social Indicators Research, vol. 97, issue 2, 2009, pp. 143-156.

[44] Graziotin D., Fagerholm F. Happiness and the productivity of software engineers. In Rethinking Productivity in Software Engineering. Apress Open, 2019, pp. 109-124.

[45] Graziotin D., Wang X., Abrahamsson P. Happy software developers solve problems better: psychological measurements in empirical software engineering. PeerJ, vol. 2, 2014, article no. 289, 23 p.

[46] Graziotin D., Fagerholm F. et al. On the Unhappiness of Software Developers. In Proc. of the 21st International Conference on Evaluation and Assessment in Software Engineering, 2017, pp. 324-333,

[47] Graziotin D., Wang X., Abrahamsson P. Software Developers, Moods, Emotions, and Performance. IEEE Software, vol. 31, issue 4, 2014, pp. 24-27.

[48] Graziotin D., Fagerholm F. et al. What happens when software developers are (un)happy. Journal of Systems and Software, vol. 140, 2018. pp. 32-47.

[49] Graziotin D., Fagerholm F. et al. Online appendix: the happiness of software developers. Figshare, 2017. Available at: https://figshare.com/collections/Online_appendix_the_happiness_of_software_developers/3355707, accessed June 01, 2021.

[50] Sujová E., Cierna H. et al. Soft Skills Integration into Business Processes Based on the Requirements of Employers — Approach for Sustainable Education. Sustainability, vol. 13, issue 24, article no. 13807, 13 p.

[51] Oeda S., Kosaku H. Development of a Check Sheet for Code-review towards Improvement of Skill Level of Novice Programmers. Procedia Computer Science, vol. 126, 2018, pp. 841-849.

[52] Bocharov N.V. Concurrent Programming Technologies and Techniques. Programming and Computer Software, vol. 29, issue 1, 2003, pp. 2-12 / Бочаров Н.В. Технологии и техника параллельного программирования. Программироание, том 29, вып. 1, 2003 г., стр. 5-23.

[53] Zavriev N.K Experience of teaching programming in the lyceum of information technologies. Programming and Computer Software, vol. 37, issue 6, 2011, pp. 288-291 / Завриев Н.К. Опыт изучения программирования в лицее информационных технологий. Программироание, том 37, вып. 6, 2011 г., стр. 19-25.

[54] V'yukova N.I., Galatenko V.A., Samborskii S.V. Support for Parallel and Concurrent Programming in C++". Programming and Computer Software, vol. 44, issue 1, 2018, pp. 35-42 / Вьюкова Н.И. Галатенко В.А., Самборский С.В. Поддержка параллельного и конкурентного программирования в языке C++. Программироание, том 43, вып. 5, 2017 г., стр. 48-59..

[55] Babić V., Slavković M. Soft and hard skills development: a current situation in Serbian companies. In Proc. of the International Conference on Management, Knowledge and Learning, 2011, pp. 407-414.

[56] Raposo V.S., Dias Meireles M.A. et al. Soft Skills, evaluation by teachers and self-evaluation by students from academic study groups in basic, technical and technological education, Research, Society and Development, vol. 9, issue 11, 2020, article no. e66391110345, 13 p. (in in Portuguese (Brazilian)).

[57] Durán M., Juárez-Ramírez R. et al. User Story Estimation Based on the Complexity Decomposition Using Bayesian Networks. Programming and Computer Software, vol. 46, issue 8, 2020, pp. 569-583 / Дуран М., Хуарес-Рамирес Р. и др. Оценка пользовательских историй на основе декомпозиции сложности с использованием байесовских сетей. Труды ИСП РАН, том 33, вып. 2, 2021 г., стр. 77-92. DOI: 10.15514/ISPRAS–2021–33(2)–4.

[58] Cerezo-Narváez A., Bastante Ceca M.J., Yagüe Blanco J.L. Traceability of Intra- and Interpersonal Skills: From Education to Labor Market. In Human Capital and Competences in Project Management, IntechOpen, 2017, pp. 87-110.

[59] Panth B., Maclean R. Introductory Overview: Anticipating and Preparing for Emerging Skills and Jobs—Issues, Concerns, and Prospects. In Anticipating and Preparing for Emerging Skills and Jobs: Key Issues, Concerns, and Prospects, Education in the Asia-Pacific Region: Issues, Concerns and Prospects, vol. 55, Springer, 2020, pp. 1-10.

[60] Ignatowski C. What Works in Soft Skills Development for Youth Employment? A Donors' Perspective. YEFG Steering Committee, 2017, 32 p. Available at: https://mastercardfdn.org/wp-content/uploads/2018/08/soft-skills-youth-employment-accessible2.pdf, accessed June 01, 2021.

[61] James R.F., James M.L. Teaching career and technical skills in a 'mini; business world. Business Education Forum, vol. 59, issue 2, 2004, pp. 39-41.

[62] Perreault H. Using podcasts to develop skills for the global workplace. Business Education Forum, vol. 61, issue 3, 2007, pp. 59-61.

[63] Fauzi R., Andreswari R. Business process analysis of programmer job role in software development using process mining. Procedia Computer Science, vol. 197, 2022, pp. 701-708.

[64] Yang H.-L., Cheng H.-H. Creative self-efficacy and its factors: An empirical study of information system analysts and programmers. Computers in Human Behavior, vol. 25, 2009, pp. 429-438.

[65] Schlichtera B.R., Buchynskab T. Soft skills of delivery managers in a co-sourced software project. Procedia Computer Science, vol. 181, 2021, pp. 905-912.

[66] Amin A., Basri S. et al. The impact of personality traits and knowledge collection behavior on programmer creativity. Information and Software Technology, vol. 128, 2020, article no. 106405, 13 p.

[67] Li W. Studying creativity and critical thinking skills at university and students' future income. Thinking Skills and Creativity, vol. 43, 2022, article no. 100980, 16 p.

[68] Younis A.A., Sunderraman R. et al. Developing parallel programming and soft skills: A project based learning approach. Journal of Parallel and Distributed Computing, vol. 158, 2021, pp. 151-163.

[69] Corno F., De Russis L., Sáenz J.P. On the challenges novice programmers experience in developing IoT systems: A Survey. The Journal of Systems and Software, vol. 157, 2019, article no. 110389.

[70] Hardy III J.H., Day E.A., Steele L.M. Interrelationships Among Self-Regulated Learning Processes: Toward a Dynamic Process-Based Model of Self- Regulated Learning. Journal of Management, vol 45, issue 8, 2019, pp. 3146-3177.

[71] Ralph P., Baltes S. et al. Pandemic Programming: How COVID-19 affects software developers and how their organizations can help. Empirical Software Engineering, vol. 25, issue 6, 2020, pp. 4927-4961.

[72] Perry S.J., Rubino C., Hunter E.M. Stress in remote work: two studies testing the demand-control-person model. European Journal of Work and Organizational Psychology, vol. 27, issue 5, 2018, pp. 577-593.

[73] Angus Reid Institute (ARI). Worry, gratitude & boredom: As covid-19 affects mental, financial health, who fares better; who is worse? Online: April 17, 2020, available at: http://angusreid.org/ covid19-mental-health/, accessed April 27, 2021.

[74] Construx. WFH in the Age of Coronavirus Lessons for Today and Tomorrow. Online: May 1, 2020, available at: https://www.construx.com/wp-content/uploads/2020/04/WFH-in-the-Age-of-Coronavirus-Report-by-Construx.pdf, accessed May 30, 2021.

[75] Russo D., Hanel P.H.P. et al. The Daily Life of Software Engineers During the COVID-19 Pandemic. In Proc. of the IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), 2021, pp. 364-373.

[76] Ford D., Storey M.-A. et al. A Tale of Two Cities: Software Developers Working from Home During the COVID-19 Pandemic. ACM Transactions on Software Engineering and Methodology, vol. 31, issue 2, article no.: 27, 37 p.

[77] Saputra N., Nasip I., Sudiana K. The Effect of Availability Digital Facility at Home on Work Productivity. In Proc. of the International Conference on Information Management and Technology (ICIMTech), 2021, pp. 783-788.

[78] Butler J., Jaffe S. Challenges and Gratitude: A Diary Study of Software Engineers Working From Home During Covid-19 Pandemic. In Proc. of the IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), 2021, pp. 362-363.

Information about authors / Информация об авторах

Reyes JUÁREZ-RAMÍREZ, Doctor of Computer Science, Full Professor. Research interests include software Engineering, software uncertainty estimation, and human-computer interaction.

Рейес ХУАРЕС-РАМИРЕС, кандидат компьютерных наук, профессор. Область научных интересов включает разработку программного обеспечения, оценку неопределенности программного обеспечения и взаимодействие человека и компьютера.

Christian NAVARRO-COTA, Master of Engineering, Assistant professor. Research interests: Ubiquitous Computing, Mobile Computing, User Experience, Mobile Application, Development, Usability, Human Machine Interaction.

Кристиан НАВАРРО-КОТА, магистр технических наук, ассистент. Научные интересы: повсеместные вычисления, мобильные вычисления, взаимодействие с пользователем,

мобильное приложение, разработка, удобство использования, взаимодействие человека и машины.

Guillermo LICEA, Doctor of Computer Science, Full Time Professor. Research interests: Software Engineering, Mobile Application Development.

Гильермо ЛИСЕА, кандидат компьютерных наук, профессор. Область научных интересов: разработка программного обеспечения, разработка мобильных приложений.

Samantha JIMÉNEZ, Doctor of Science, Full Professor. Research interests include Software Engineering, Usability, Educational Technology, Human-Computer Interaction.

Саманта ХИМЕНЕС, кандидат наук, профессор. Область научных интересов включает разработку программного обеспечения, удобство использования, образовательные технологии, взаимодействие человека и компьютера.

Verónica TAPIA-IBARRA, Researcher, Research interests: Java Programming, C++, SQL.

Вероника ТАПИА-ИБАРРА, исследователь, Область научных интересов: программирование на Java, C++, SQL.

César Arturo GUERRA GARCÍA, Doctor of Computer Science, Full Time Professor. Research interests include Software Engineering, Data and Information Quality, Requirements Engineering.

Сезар Артуро ГЕРРА ГАРСИА, кандидат компьютерных наук, профессор. Область научных интересов включает разработку программного обеспечения, качество данных и информации, разработку требований.

Hector Gerardo PEREZ-GONZALEZ, Ph.D., Full Time Professor. Research interests include Software Engineering, Software Design, Requirements Engineering.

Гектор Херардо ПЕРЕС-ГОНСАЛЕС, кандидат наук, штатный профессор. Область научных интересов включает разработку программного обеспечения, проектирование программного обеспечения, разработку требований.

DOI: 10.15514/ISPRAS-2023-35(1)-4

Microservice Deployment

V.M. Niño-Martínez, ORCID: 0000-0002-8436-1430 <ninomtz.victor@gmail.com>
J.O. Ocharán-Hernández, ORCID: 0000-0002-2598-1445 <jocharan@uv.mx>
X. Limón, ORCID: 0000-0003-4654-636X <hlimon@uv.mx>
J.C. Pérez-Arriaga, ORCID: 0000-0003-2354-2462 <juaperez@uv.mx>

University of Veracruz,
Xalapa, Veracruz, 91020, Mexico

Abstract. Modern software development requires agile methods to deploy and scale increasingly demanded distributed systems. Practitioners have adopted the microservices architecture to cope with the challenges posed by modern software demands. However, the adoption and deployment of this architecture also creates technical and organizational challenges, potentially slowing down the development and operation teams, which require more time and effort to implement a quality deployment process that allows them to constantly release new features to production. The adoption of a DevOps culture, along with its practices and tools, alleviates some of these new challenges. In this paper we propose a guide for the deployment of systems with a microservices architecture, considering the practices of a DevOps culture, providing practitioners with a base path to start implementing the necessary platform for this architecture. We conducted this work following the Design Science Research Methodology for Information Systems (DSRM). In this way, we identified the problem, and also defined the solution objectives through the execution of a Systematic Literature Mapping and a Gray Literature Review, having as a result the proposed guide. This work can be summarized as follows: (I) Identification of practices and technologies that support the deployment of microservices. (II) Identification of recommendations, challenges, and best practices for the deployment process. (III) Modeling of the microservices deployment process using SPEM. (IV) Integration of the knowledge in a guide to deploy microservices by adopting DevOps practices.

Keywords: agile methods; microservices architecture; deployment; DevOps; Systematic Literature Mapping; Gray Literature Review

For citation: Niño-Martínez V.M., Ocharán-Hernández J.O., Limón X., Pérez-Arriaga J.C. Microservice Deployment. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 57-72. DOI: 10.15514/ISPRAS-2023-35(1)-4

Развертывание микросервисов

В.М. Ниньо-Мартинес, ORCID: 0000-0002-8436-1430 <ninomtz.victor@gmail.com>
Х.О. Очаран-Эрнандес, ORCID: 0000-0002-2598-1445 <jocharan@uv.mx>
К. Лимон, ORCID: 0000-0003-4654-636X <hlimon@uv.mx>
Х.К. Перес-Арриага, ORCID: 0000-0003-2354-2462 <juaperez@uv.mx>

Университет Веракруса,
91020, Мексика, Веракрус, Халапа

Аннотация. Современная разработка программного обеспечения требует гибких методов для развертывания и масштабирования все более востребованных распределенных систем. Практики применяют архитектуру микросервисов, чтобы справиться с проблемами, связанными с современными требованиями к программному обеспечению. Однако использование этой архитектуры также создает технические и организационные проблемы, потенциально замедляя работу групп разработки и

эксплуатации, которым требуется больше времени и усилий для реализации качественного процесса развертывания, позволяющего им постоянно выпускать новые функции в рабочую среду. Принятие культуры DevOps вместе с ее методами и инструментами смягчает некоторые из этих новых проблем. В этой статье мы предлагаем руководство по развертыванию систем с микросервисной архитектурой с позиции культуры DevOps, предоставляющей практикам путь к началц внедрения необходимой платформы для этой архитектуры. Мы провели эту работу в соответствии с Методологией исследований в области проектирования информационных систем. Таким образом, мы определили проблему, а также определили цели решения путем выполнения систематического обзора литературы, включая серую литературу. Эту работу можно резюмировать следующим образом: (I) определение методов и технологий, поддерживающих развертывание микросервисов; (II) определение рекомендаций, проблем и лучших практик для процесса развертывания; (III) моделирование процесса развертывания микросервисов с помощью; (IV) интеграция знаний в руководство по развертыванию микросервисов с применением практики DevOps.

Ключевые слова: гибкие методы; микросервисная архитектура; развертывание; DevOps; систематический обзор литературы; обзор серой литературы

Для цитирования: Ниньо-Мартинес В.М., Очаран-Эрнандес Х.О., Лимон К., Перес-Арриага Х.К. Развертывание микросервисов. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 57-72. DOI: 10.15514/ISPRAS-2023-35(1)-4

1. Introduction

In the 1990s, the popularization of the World Wide Web (WWW) and the subsequent dot-com gold rush introduced the world to software as a service (SaaS), leading to entire industries built on this SaaS model. This motivated the development of applications that required more resources, making them more complex to develop, maintain and deploy. Nowadays, enterprise systems need to transfer information with other systems, internal or external to the organization, even at a global scale. Companies such as Amazon, Netflix [1], Uber [2], LinkedIn [3] and, SoundCloud [4], among others, found the need to migrate to a software architecture that allows them to undertake the complexity and constant need of evolution of their systems. To this end, they chose to adopt a Microservices Architecture (MSA). Not only have these large companies migrated to an MSA, but small and medium-sized companies have also done so, all of them seeking the benefits that this architecture brings, such as scalability, heterogeneity, and extensibility, among others.

A MSA is an approach to developing a distributed system as a set of small services. Each of these services runs in its process and communicates using lightweight mechanisms, like an HTTP resource API [5]. One of the characteristics that make this architecture different is the granularity of the services, which must be small and highly cohesive. Microservices adopt the single responsibility principle approach, which states “Gather together the things that change for the same reasons, separate those things that change for different reasons” [6], focusing the service boundaries on the business boundaries, in this way, preventing services from growing too large as well as the difficulties that this may introduce. The key benefits that microservices architecture offers over conventional architectural patterns are: the heterogeneity of technologies, fault tolerance, agile deployment, scalability, alignment with organizational structure, replaceability, and agile development of business functionality [7, 8].

Software deployment is a stage of the software development life cycle in which a system is put into operation and transition issues are resolved [9]. Deployment combines two closely related concepts, the first one is the deployment process, which consists of a series of steps that must be executed by the developers or those in charge of managing the system infrastructure to put the software into a production environment, and the second is the deployment architecture, which defines the structure of the software execution environment [10]. An application is only useful when deployed to users. Mature deployment practices are crucial to building reliable and stable microservices.

Unlike a monolithic system, optimized for a single-use case, microservices deployment practices need to scale to multiple services; it is possible to have tens or hundreds of microservices, written in different programming languages and frameworks. Each microservice is a small application with a specific process and architecture, which operators and developers need to deploy in production. If operators and developers are not able to quickly and reliably deploy microservices, then the added development speed gained from microservices would be useless. Therefore, a mature deployment process and automated deployments are essential for developing microservices at scale.

When migrating from a monolithic approach to deploy microservices, the main challenges are the familiarization with the variety of technologies and tools, the automation of the process, and the implementation of a pipeline to continuously deploy [11]. In addition, among the most important challenges related to the deployment of this type of architecture are: 1) maintaining stability for a large volume of releases and component changes; 2) avoiding coupling between components, leading to dependencies in the build or release times; 3) managing changes in the service API, as changes could negatively affect the clients; and 4) removing and updating production services [12]. The practices found in DevOps aid to alleviate the mentioned challenges, these practices include: Continuous Integration (CI), Continuous Delivery (CD), Configuration Management (CM), and monitoring, among others. The implementation of these practices generates new challenges regarding: communication and coordination between teams; lack of investment in costs; lack of experience and skills; conflict management; design and code dependencies between components; implementation and release of software to customers [13].

To help developers and people in charge of creating a stable infrastructure to deploy microservices, we decided to elaborate a guide for the deployment of microservices-based systems, considering DevOps culture practices. The goal of the guide is to reduce the effort associated with creating an ecosystem for the microservices architecture. The guide integrates different organizational technical decisions, technologies, and tools successfully used by organizations, as well as the associated DevOps practices. The guide helps all related parties in the process of adopting a microservices architecture.

In order to create the guide, we followed the Design Science Research Methodology (DSRM) methodology[14], consisting of six phases. We have already completed the following phases: identification of practices, technologies, tools, activities, and recommendations for the deployment of microservices, through a previous work [15] consisting of a systematic mapping of the literature and a review of gray literature; classification and grouping of the information found; MSA process adoption modeling; and the selection and integration of related activities according to the adoption process. With these phases covered, it is possible to have a first version of the microservices deployment guide, leaving the demonstration and evaluation as future work.

This paper is organized as follows. Section 2 gives an overview of some studies focused on the deployment of microservices and the adoption of DevOps practices. Section 3 presents the followed method to develop our microservices deployment guide, based on the DSRM methodology [14]. Section 4 describes the proposed deployment guide and its structure. Finally, Section 5 features the conclusion and future work.

2. Research Method

We followed the Design Science Research Methodology (DSRM) [14], establishing the recognition and legitimization of aims, processes, and investigation outputs, and helping researchers to present their work according to a common framework. The methodology incorporates principles, practices, and procedures required to carry out such research, meeting three objectives: consistency with prior literature, providing a nominal process model for doing Design Science (DS) research, and providing a mental model for presenting and evaluating DS research. Several studies have used this methodology to develop artifacts and validate its process, for example [16, 17]. DSRM includes six

steps: problem identification and motivation, the definition of the objectives, design, and development, demonstration, evaluation, and communication. We detail these phases in the following subsections.

2.1 Problem identification and motivation

For the identification of the problem related to microservices deployment and its importance, we performed a preliminary literature review. The concepts and topics analyzed were the microservice architecture style; advantages and drawbacks of its use; processes to deploy microservices; aspects that affect the deployment; and DevOps culture and its practices.

One of the main challenges we found, is the familiarization with the variety of technologies and tools, as well as the automation and implementation of a pipeline to deploy continuously [11]. Moreover, the implementation of practices such as Continuous Integration (CI), Continuous Delivery (CD), Configuration Management (CM), and monitoring; bring new challenges, such as communication and coordination between teams; lack of investment in costs; lack of experience and skills; conflict management; design and code dependencies between components; challenges in the implementation and release of software to customers [13, 18].

With our literature review, we developed a cause-effect diagram to reflect the factors that impact the deployment of microservices and convert it into a challenging process. Figure 1 shows the cause-effect diagram.

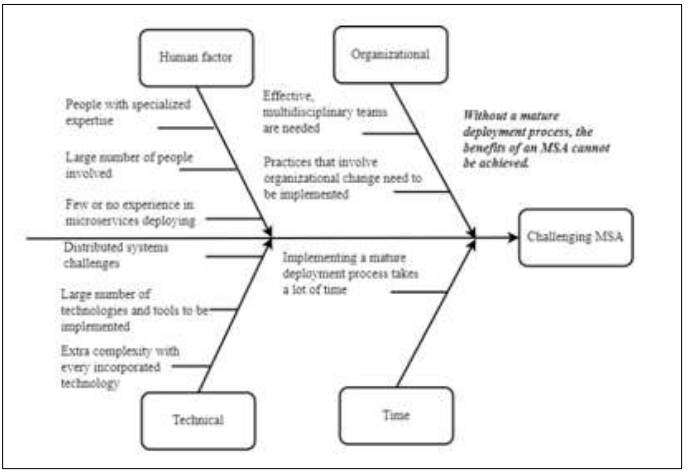

Fig. 1. Cause-effect diagram of a microservice deployment

2.2 Design the objectives for a Solution

Once we identified the problems, we concluded that a guide to deploy a microservice architecture could help to solve the problems. To know the state of the art, and the possible solutions, we performed a Systematic Mapping Study and A Gray Literature Review, both with the aim to identify practices, processes, technologies, recommendations, and lessons learned and reported by practitioners.

2.2.1 Systematic Mapping Study

We conducted the study following the guidelines of Kitchenham, Budgen, and Brereton [19], the guidelines describe a process to perform the mapping in Software Engineering. The objective of a mapping study is to survey the available knowledge about a topic. It is possible to synthesize information by categorization, identify "clusters" of studies that could form the basis of a fuller

review, and also identify "gaps", indicating the need for more primary studies. We executed the mapping study in three main phases: planning, conduction, and results report. Some activities carried out within these phases were: a preliminary literature review; definition of the research questions and search keywords; database selection; inclusion, and exclusion criteria; methods for the data extraction and analysis.

Planning

Research questions: Derived from the objective of the work, we formulated four research questions (RQ). The questions compiled the state of the art, showing us the techniques and technologies that researches, and practitioners use to deploy microservices, along with the related DevOps practices. The RQs and their motivation are shown in Table 1.

Table 1. Research Questions and Motivation

Questions	Motivation
RQ-1: What DevOps practices and approaches support the deployment of Microservices?	Identify the practices and approaches used in the DevOps culture and classify the technologies needed for each practice
RQ-2: What technologies do DevOps practices use to deploy Microservices?	It is important to identify the technologies that are used in each DevOps practice, to understand which are the most suitable for a given situation
RQ-2: What technologies do DevOps practices use to deploy Microservices?	It is important to identify the technologies that are used in each DevOps practice, to understand which are the most suitable for a given situation
RQ-3: What challenges does the literature report regarding the adoption of DevOps practices in the deployment of microservices?	Many problems can emerge in the implementation of the practices and this question aims to know what they are and how often they are reported.
RQ-4: What lessons does the literature report for successful microservices deployment?	This question aims to identify the processes, best practices, and recommendations that practitioners implemented in the deployment of their systems and serve as a guide for those in the same situation.

Research process: We performed a preliminary literature review, identifying a series of articles that helped us to define a set of keywords representing the main concepts around the research questions and, some of their related concepts. In the end, we decided to run an automated search for selecting primary studies. We constructed a base string with the search terms identified, refined, and validated using the Recall and Precision techniques. The generated string is the following:

> *(microservices OR "microservice architecture" OR micro-services OR "architecting microservices") AND (DevOps OR development OR operations OR "continuous integration" OR CI OR "continuous deployment" OR "continuous delivery" OR CD OR migration OR automation OR tools OR adoption OR monitoring OR cloud)*.

Table 2. Selected Electronic Databases

Database	Link
IEEE Xplore Digital	https://ieeexplore.ieee.org
Elsevier Science Direct	https://www.sciencedirect.com
Springer Link	https://link.springer.com
Wiley Online Library	https://onlinelibrary.wiley.com
ACM Digital Library	https://dl.acm.org

Table 2 shows the selected databases that to conduct the search. We chose these databases because they compile the most significant number of works related to Software Engineering. In addition, in a previous manual review, we found results in the mentioned sources. ACM Digital Library and Elsevier Science Direct repositories have some considerations in their search engines, so we adjusted the search string. Due to the large number of results obtained in ACM, we decided to search only using the title as the indexer. In the Wiley repository, we used the exact string as in Science Direct, because, in the first tests, we observed that it performed better. We present the search string of each

database in Table 3. We only covered the last five years in the study, in these years the topics of DevOps and Microservices had more relevance in research articles. We have also observed in these years an increase in popularity of the topics of interest, and therefore it is of relevance for the study. We defined a list of inclusion and exclusion criteria for the studies, presented in Table 4.

Table 3. String Adjusted to each database

Source	String
ACM Digital Library	[microservice* OR “microservice architecture” OR "architecting microservices"] AND [DevOps OR development OR operations OR “continuous integration” OR CI OR “continuous deployment” OR “continuous delivery” OR CD OR migration]
Elsevier Science Direct	(microservice OR “microservice architecture”) AND (devops OR development OR operations OR “continuous integration” OR “continuous deployment” OR “continuous delivery” OR migration)
Springer Link	(devops OR development OR operations OR “continuous integration” OR “continuous deployment” OR “continuous delivery” OR migration)
Wiley Online Library	(devops OR development OR operations OR “continuous integration” OR “continuous deployment” OR “continuous delivery” OR migration)

Table 4. Inclusion and Exclusion Criteria

Database	Link
IC-1: Studies published between 2015 and 2020.	EC-1: It is an abstract, workshop, opinion article, presentations, book chapters, or conference notes.
CI-2: Articles written in English	EC-2: The study does not focus on Microservices and DevOps process deployment
IC-3: The title and abstract contain information indicating that the full text could answer at least one research question.	EC-3: The study is an earlier version of more recent work
IC-4: The full text answers at least one research question	

Data extraction: We defined a template to extract the necessary information from each article to answer the research questions. Data D1-D10 contains the general information of each study, and data D11-D16 helped to extract qualitative data that answers the research questions. We used a spreadsheet to collect the information.

Data synthesis: For information synthesis, we used the meta-aggregation method [20]. The synthesis brings together the study findings, communicated as themes, metaphors, categories, or concepts; and grouped by further aggregation based on similarity of meaning [20]. This method helped us to identify lessons learned, common mistakes and understand why the literature reports certain technologies a higher number of times. Moreover, with the information classified and grouped, its analysis becomes a more straightforward process.

Conduction

We conducted the selection process in three stages, implementing the inclusion and exclusion of the strings in the different sources, and using the filters provided by each of them, the CI-1 and CI-2 criteria corresponding to the years of publication and their language were applied. In addition, in databases such as Science Direct, Springer Link, and ACM Digital Library, we used filters to only include research articles and not book chapters or lecture notes, thus applying the execution criteria CI-3 as well as the exclusion criteria CE-1 and CE-2. In the third stage, we read the full text, and the inclusion and exclusion criteria CI-4, and CE-3 were applied. Figure 2 shows the results after applying the inclusion and exclusion criteria by stage and database. At the end of the third stage, we obtained a total of 21 primary studies.

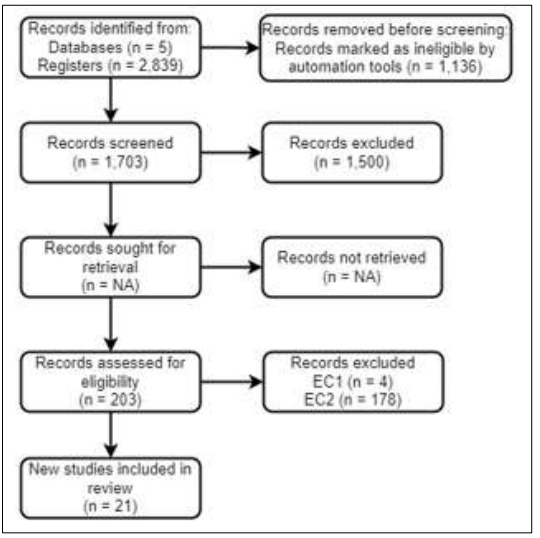

Fig. 2. Selection process

Data extraction and analysis: Once we selected the primary studies, we created a spreadsheet in which each column presents the to be extracted data. We performed a complete reading of each article, highlighting the information that answered the research questions and capturing this information in the spreadsheet; we performed this process for each of the primary studies selected. With the extracted information, we proceeded to apply the meta-aggregation method. This method has three main steps: (I) Identify and assemble findings from all included studies; (II) Aggregate well-founded and explicit findings; (III) Synthesis of findings implications. We also captured the findings in a spreadsheet, and with all the findings identified, we iteratively created categories and grouped findings on them.

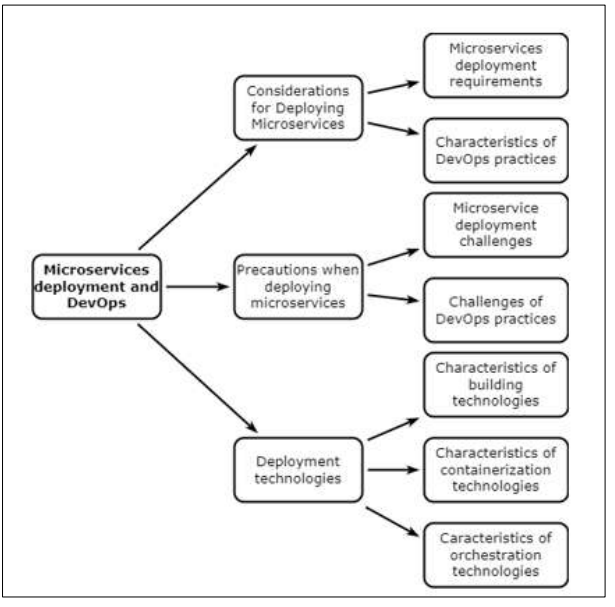

Fig. 3. Meta-aggregation classification

Results

Meta-aggregation results: After the application of the method, we extracted classified 43 findings into seven categories. These categories were grouped into three synthesized findings Considerations for Deployment Microservices, Precautions when Deploying Microservices, and Deployment Technologies. Figure 3 shows the associations between categories.

Microservices deployment requirements: In this category, we identified requirements that practitioners, from their experience in the area, considered necessary for a successful microservices deployment. We found that architectural support is crucial for the adoption of DevOps practices, as well as having a mature operations team, to allow continuous deployment of numerous microservices. Furthermore, developers need to consider microservices' backward compatibility, and microservices upgrading with minimum effort and application downtime. Flexible and maintainable delivery systems support these needs.

Characteristics of DevOps practices: We grouped in this category, requirements, tips, and lessons learned by practitioners when implementing DevOps practices as well as deployment pipelines. The practitioners agree that pipelines are one of the key parts in the deployment of microservices because without good construction of pipelines, long wait times for releases and builds occur. To prevent it, it is necessary to apply DevOps principles in building CI/CD pipelines, automation is paramount to successful deployment.

Microservices deployment challenges: The findings related to this category are challenges those practitioners identified when adopting a microservices-based architecture. One of the challenges identified is the release of a new version of a microservice, because one or more microservices may depend on it. In addition, when adopting this architecture, there is a great effort in the context of new tools and frameworks. Microservices configuration is essential to achieve the expected results.

Challenges of DevOps practices: In this category, we grouped a set of challenges related to practices and technologies related to DevOps practices. The constant updating of tools and libraries makes development difficult, as well as the lack of tools for specific tasks that developers need to automate. For example, monitoring has several challenges such as lack of commercial options, lack of standardization, and lack of faster learning curves.

Characteristics of building technologies: Technologies are an important part of software deployment and construction; therefore, it is an aspect that practitioners pay particular attention to. In this category, we gathered characteristics mentioned by practitioners for these technologies. Some examples are integration servers such as Jenkins, GitLab CI, and Travis CI. Also, as part of the findings of the category, we made a comparison and characteristics of usage of each technology.

Characteristics of containerization technologies: The use of containerization technologies, such as Docker, is one of the characteristics that popularized the microservices architecture. Many studies recommend the use of this technology, contrasting the advantages with respect to virtual machines. Deploying microservices using containers takes significantly less time than using virtual machines. The use of containers makes deployment a simple, fast, and platform-independent process. The mentioned benefits come from the fact that developers can automate the construction and provisioning of containers using scripts.

Characteristics of orchestration technologies: As a result of the wide adoption of containerization technologies, solutions for their orchestration have emerged. Technologies such as Kubernetes, Docker Swarm, and Docker-compose, among others, provide practitioners with various deployment benefits. This category presents a comparison in terms effectiveness of these technologies, and also compiles the experiences that developers had with their adoption. Kubernetes for container orchestration is the most suitable method for deploying microservices when the application demands high availability and scalability, however when it comes to security Kubernetes and Docker Swarm do not provide complete isolation between deployed containers, which introduces security issues.

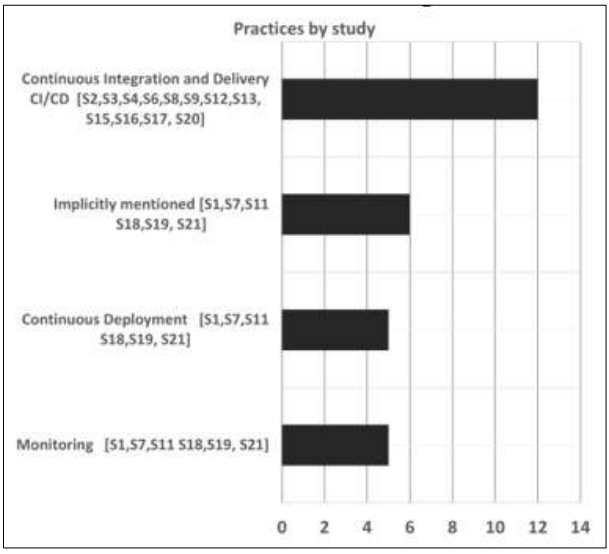

Fig. 4. Practices mentioned by study

Answers to research questions: *What DevOps practices and approaches support the deployment of Microservices?* The studies mentioned the DevOps practices of Continuous Integration (CI), Continuous Delivery (CD), Continuous Deployment and Monitoring. However, some studies did not directly mention the use of DevOps practices but used the processes and activities of these practices. Figure 4 shows the practices reported and related articles. It is worth noting that some studies mentioned more than one practice.

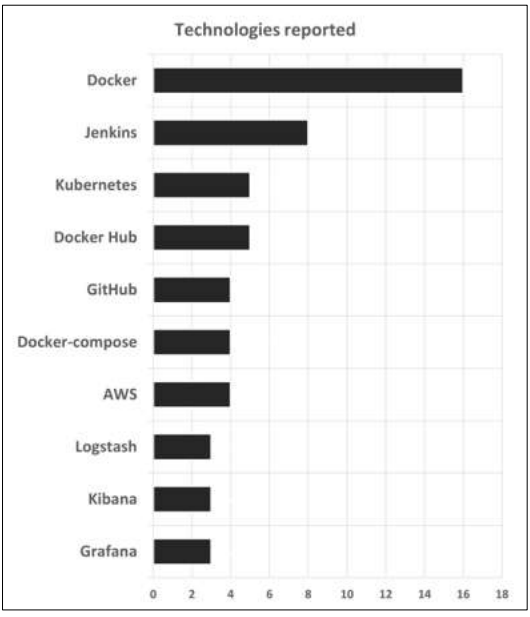

Fig. 5. Technologies reported by the studies

What technologies do DevOps practices use to deploy Microservices? We found several technologies for the construction and deployment of microservices. Figure 5 presents the ten most frequently reported technologies.

Studies mentioned Docker, a containerization technology, 16 times. The literature compares containers with other similar technologies such as Virtual Machines (VM), and in each comparison, the studies concluded that the former provided more significant benefits. The literature also highlights DockerHub as a repository for container images.

Another important technology is Jenkins, a building technology used in CI/CD practices, mentioned in the literature eight times. In contrast, the literature only mentions once Circle CI and Travis-CI, which are similar to Jenkins.

Among deployment and orchestration technologies, the literature mentions Kubernetes, Docker-compose, and Docker Swarm. Kubernetes was the most used because it provides significant benefits in systems with many microservices. Finally, the literature also mentions GitHub and Gitlab four and three times, respectively.

What challenges does the literature report regarding the adoption of DevOps practices in the deployment of microservices?

Publishing and upgrading microservices: Updating and publishing a new microservice version is a significant challenge, developers have to be careful since a microservice may depend on many others [21]. In addition, service discovery is a challenging aspect affected by upgrading a new version of a microservice and deploying it [22].

Technologies and tools required for building and deploying microservices: Developers make a great effort to adopt new tools and frameworks for each practice that they implement [23]. It is crucial to choose the right tools to protect the DevOps approach; otherwise, the rollback or tool change is very costly in time and effort [24]. Developers must perform careful initial configuration of the tools as this will allow correct automation [25]. Constant updates of libraries and tools make development and maintenance difficult.

Monitoring of a microservices architecture: The challenges that practitioners must face are the lack of commercial monitoring options, lack of standardization, and lack of faster learning curves [26].

What lessons does the literature report for successful microservices deployment? We grouped the lessons learned into two main topics: Solid architectural foundations and Attention to DevOps principles.

Solid architectural baseline: A long and scalable system requires a good architectural foundation that supports DevOps [27]. Every change in the architecture imposes new requirements on the delivery system and the implementation of new components and technologies [28]. Backward compatibility between microservices, separation of domains, and responsibilities for each service helps to prevent cross-configuration and keep services running smoothly.

Attention to DevOps principles: Applying DevOps principles in building CI/CD pipelines makes them leaner and more robust. Principles such as automation in all processes (integration, testing, deployment, analysis, and monitoring) are key to ensuring system reliability [29]. Good design and implementation of deployment pipelines allow rapid error detection [24]. Maintenance and updating of pipelines should take priority over code development. When problems arise, it is important to centralize error handling, in order to reduce the work of developers and operators. System monitoring should be flexible and scalable.

2.2.2 Gray literature review

We conducted a gray literature review to complement the mapping findings. For the review, we considered books, and electronic resources focused on the topics of DevOps, microservices deployment, and associated technologies. We searched the resources using the search engines Google Scholar, Google Books, and Google. We used these three since we aimed to have as much information as possible. In addition, we applied the snowballing method [30], which consists of searching for the material cited or referenced in the mapping articles. The steps carried out for the

selection of the resources were as follows: For the selection of books, the process consisted of reading the table of contents, and the chapters that corresponded to the deployment of microservices or some DevOps practice related to microservices. For the selection of electronic resources such as company blogs, standards, and technical documentation, we read the content to determine if it would be useful. We investigated each of the resources to answer the research questions formulated in the MSL, or at least to find information that contributes to the findings.

Once we identified the resources, we continued with the reading of the most relevant aspects. Following a process similar to the meta-aggregation method used in the mapping, we identified important ideas or findings, and classify them according to their type. Among the types identified are deployment patterns, principles, practices, advantages, and disadvantages of technologies and resources.

2.3 Design and Development

In the design and development phase, we performed a series of activities, these activities consisted of grouping and classifying the information obtained from the white and gray literature reviews. We focus on the implementation modeling process of a microservices architecture, aiming to provide an order to the set of tasks and activities that we identified in previous phases. Finally, using the modeling and the information obtained, we integrated the microservices deployment guide, which we structured according to the modeling phases, having as content the related activities in each phase.

2.4 Demonstration and Evaluation

The demonstration aims to use the artifact to solve one or more instances of the problem. To achieve it, the authors propose certain approaches such as experimentation, simulation, case study, or other appropriate activity. Once performed the demonstration is needed to observe and measure how well the artifact supports a solution to the problem. However, given the complexity, the amount of time, personnel, and resources involved in building a microservices architecture large enough to be applied as a case study, as well as the number of case studies that would be needed to have deterministic results, it was decided not to include this phase in the scope of this work.

For the evaluation of the guide, we decided to use another approach and analyze the evaluation method that best suits our problem, so far, we are considering using the work of Garousi et al. [31] and focusing on the evaluation of quality for technical software documents, thus the application of the evaluation is planned as future work.

2.5 Communication

As a part of the communication phase, we communicated the importance of the problem through the paper publication Microservice Deployment: A Systematic Mapping Study [15]. For the artifact communication, its utility, and effectiveness we present the current paper, and we are developing a website to publish the guide so it could be accessible for the practitioners.

3. Proposed Deployment Guide

The guide works as a path where practitioners can identify their starting point and gradually adopt practices and strategies for microservices deployment. The guide includes practices, patterns [32], technologies, and tips found in the literature. The guide organizes possible decisions according to the phases of the microservices deployment process. Organizations interested in adopting the MSA can follow the guide, in this way, the person in charge of design or deployment can consult the practices and strategies recommended for each specific phase. The intention of showing the decisions in a modular way is that the managers can consult the parts they need, without the need to

read the whole guide, or if practitioners have already managed to adopt some practices, they can find additional information that allows them to improve their current process.

We used SPEM 2.0 (Software & Systems Process Engineering Metamodel) for the modeling of the guide, it is a standard for defining software processes. SPEM uses the UML (Unified Modeling Language) notation, which provides components that allow the standardized representation of methods, life cycles, roles, activities, tasks, and work products used in Software Engineering. The main process consists of three phases. Each phase can have different iterations, an iteration is a set of activities performed iteratively, and each activity has one or many tasks needed to complete the activity. Due to the time involved in having a platform that supports the microservices architecture, practitioners can perform all these activities iteratively and incrementally as the project develops, thus adding value to the deployment process as the project and its needs grow.

The first phase corresponds to the architectural design, separating the problem domain, identifying the required microservices, the communication style between them, and the deployment method for orchestrating the microservices. The second phase presents the preparation of the development environment for each microservice; the related activities in the construction; integration and delivery of each service; and finally, the strategies for delivery and observability of the microservices in the production environment. The third and last phase, covers microservice construction, following the design and platform created in the previous phases. The following is a description of the sections that make up the guide as well as the related activities and tasks.

3.1 Deployment design

This section of the guide covers the design and deployment planning iteration, which has four main activities for those responsible for the design and implementation of the system. Each activity has an output that serves as input for the next task, the first activity is the selection of the deployment strategy, followed by the selection of technologies, and finally, the last two activities, possibly executed in parallel, corresponding to the design of configurable services, and the design of observable services can.

The activities described in this section contain the following information: Name, Roles in charge, Description, List of identified methods or patterns, and Recommendations. Each identified pattern has the following properties: Characteristics, Advantages, Disadvantages, and Technology.

3.2 Configuration management and development environment

This section encompasses the Iteration Delivery Environment Preparation activity for the preparation of the deployment pipeline. This activity is very important since it is the basis that will allow the implementation of a deployment pipeline, the person in charge of the deployment has the task of implementing a set of practices and technologies that allow the control of the changes made in the service's code, as well as the automation of the processes for the construction of services. The activities implemented are the Implementation of version control, Establishment of development guidelines, Implementation of patterns for source code branch management, implementation of unit tests, and automation of the build and test processes.

3.3 Deployment pipeline

This section of the guide presents DevOps activities related to Continuous Integration and Continuous Delivery practices. The section incorporates two activities from the iteration phase of the deployment pipeline: the preparation of the built environment, and the preparation of the delivery environment. These activities are fundamental to constantly building and releasing microservices, a key aspect of successfully implementing MSA. The section features recommendations, technologies, and features for each task. The first activity corresponds to the practice of Continuous

Integration, this activity concerns the implementation of a continuous integration system; automation of the compilation process; implementation of unit and acceptance tenting; implementation of code analysis and generation of binaries; and packaging artifacts. The second activity, focused on the Continuous Delivery practice, concerns the tasks of environment configuration; implementation of smoke tests; implementation of manual tests; acceptance or performance tests; and deployment and release to a production environment.

3.4 Infrastructure management and System observability

This section presents the tasks that correspond to DevOps culture practices, such as Infrastructure as Code and GitOps. Here we present the description of these practices, the description of the existing technologies, as well as good practices found in the literature for their correct implementation. In addition, the last section presents the practices we found in the literature to achieve adequate observability of the services deployed in a production environment.

4. Conclusion and Future Work

This paper presented the current results of a project to build a deployment guide for applications with a microservices architectural style. To this end, we conducted a systematic mapping study to identify the practices, tools, technologies, activities, and recommendations used in microservices deployment, we also complemented the information found with a gray literature review. We integrated into the guide all the elements and models found.

As for future work, we plan to perform the evaluation phase of the DSRM methodology. This phase is for analyzing the guide and related artifacts, to know if they meet the intended objectives. To perform the evaluation of the guide we intend to use the work of Garousi et al. [31] for the evaluation of the use and quality of software technical documentation.

The present version of the artifact does not cover organizational aspects of the DevOps culture. To obtain the benefits of a DevOps culture, organizations not only have to adopt technologies and practices, but they also have to adopt an organizational and cultural base, driven by the highest levels of the organization. Therefore, as future work, the guide will incorporate the organization of effective teams for microservices deployment. In this way, the work would bring additional value to organizations and to all those who seek to adopt a DevOps culture.

References / Список литературы

[1] Mauro T. Adopting Microservices at Netflix: Lessons for Architectural Design. NGINX Blog, 2015. Available at: https://www.nginx.com/blog/microservices-at-netflix-architectural-best-practices/, accessed May 10, 2021.

[2] Reinhold E. Rewriting Uber Engineering: The Opportunities Microservices Provide. Uber Engineering Blog. Available at: https://eng.uber.com/building-tincup-microservice-implementation/, accessed May 10, 2021.

[3] Ihde S., Parikh K. From a Monolith to Microservices + REST: the Evolution of LinkedIn's Service Architecture. Mar. 2015. Available at: https://www.infoq.com/presentations/linkedin-microservices-urn/, accessed Mar. 22, 2022)

[4] Calcado P. Building Products at SoundCloud —Part I: Dealing with the Monolith. SoundCloud Backstage Blog, Jun. 11, 2014. Available at: https://developers.soundcloud.com/blog/building-products-at-soundcloud-part-1-dealing-with-the-monolith, accessed Mar. 22, 2022.

[5] Lewis J., Fowler M. Microservices. Mar. 25, 2014. Available at: https://martinfowler.com/articles/microservices.html, accessed Nov. 16, 2021.

[6] Martin R.C. The Single Responsibility Principle. Clean Coder Blog, May 08, 2014. Available at: https://blog.cleancoder.com/uncle-bob/2014/05/08/SingleReponsibilityPrinciple.html, accessed Jan. 26, 2022.

[7] Newman S. Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, 2015, 280 p.

[8] Indrasiri K., Siriwardena P. Microservices for the Enterprise: Designing, Developing, and Deploying. Apress, 2018, 441 p.

[9] IEEE Standard for DevOps: Building Reliable and Secure Systems Including Application Build, Package, and Deployment: IEEE Standard 2675-2021.

[10] Richardson C. Microservices Patterns: with examples in Java. Manning, 2018, 520 p.

[11] Fritzsch J., Bogner J. et al. Microservices Migration in Industry: Intentions, Strategies, and Challenges. In Proc. of the IEEE International Conference on Software Maintenance and Evolution (ICSME), 2019, pp. 481-490.

[12] Bruce M., Pereira P.A. Microservices in action. Manning, 2018, 392 p.

[13] Shahin M., Ali Babar M., Zhu L. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. IEEE Access, vol. 5, 2017, pp. 3909-3943.

[14] Peffers K., Tuunanen T. et al. A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, vol. 24, issue 3, 2007, pp. 45-77.

[15] Niño-Martínez V.M., Ocharán-Hernández J.O. et al. Microservices Deployment: A Systematic Mapping Study. In Proc. of the 9th International Conference in Software Engineering Research and Innovation (CONISOFT), 2021, pp. 24-33.

[16] Hyvärinen H., Risius M., Friis G. A Blockchain-Based Approach Towards Overcoming Financial Fraud in Public Sector Services. Business & Information Systems Engineering, vol. 59, issue 6, 2017, pp. 441-456.

[17] Tello-Rodríguez M., Ocharán-Hernández J.O. et al. A Design Guide for Usable Web APIs. Programming and Computer Software, vol. 46, issue 8, 2020, pp. 584-593 / Тельо-Родригес М., Очаран-Эрнандес Х.О., Перес-Арриага Х.К., Лимон К., Санчес-Гарсия А.Х. Путеводитель по проектированию удобных Web-API. Труды ИСП РАН, том 33, вып. 1, 2021 г., стр. 173-188. DOI: 10.15514/ISPRAS-2021-33(1)-12.

[18] Chen L. Continuous Delivery: Overcoming adoption challenges. Journal of Systems and Software, vol. 128, 2017, pp. 72-86.

[19] Kitchenham B., Budgen D., Brereton P. Evidence-Based Software Engineering and Systematic Reviews. Chapman and Hall/CRC, 2015, 399 p.

[20] Pearson A., Robertson-Malt S., Rittenmeyer L. Synthesizing Qualitative Evidence. Lippincott Williams & Wilkins, 2011, 80 p.

[21] Kargar M.J., Hanifizade A. Automation of regression test in microservice architecture. In Proc. of the International Conference on Web Research (ICWR), 2018, pp. 133-137.

[22] Singh V., Peddoju S.K. Container-based microservice architecture for cloud applications. In Proc. of the International Conference on Computing, Communication and Automation (ICCCA), 2017, pp. 847-852.

[23] Richter D., Konrad M. et al. Highly-Available Applications on Unreliable Infrastructure: Microservice Architectures in Practice. In Proc. of the IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2017, pp. 130-137.

[24] Soenen T., Van Rossem S. et al. Insights from SONATA: Implementing and integrating a microservice-based NFV service platform with a DevOps methodology. In Proc. of the IEEE/IFIP Network Operations and Management Symposium, 2018, pp. 1-6.

[25] Fan C.Y., Ma S.P. Migrating Monolithic Mobile Application to Microservice Architecture: An Experiment Report. In Proc. of the IEEE International Conference on AI & Mobile Services (AIMS), 2017, pp. 109-112.

[26] Tamburri D.A., Miglierina M., Di Nitto E. Cloud applications monitoring: An industrial study. Information and Software Technology, vol. 127, 2020, article no. 106376, 28 p.

[27] Chen H.M., Kazman R. et al. Architectural Support for DevOps in a Neo-Metropolis BDaaS Platform. In Proc. of the IEEE 34th Symposium on Reliable Distributed Systems Workshop (SRDSW), 2015, pp. 25-30.

[28] Steffens A., Lichter H., Döring J.S. Designing a next-generation continuous software delivery system: Concepts and architecture. In Proc. of the 4th International Workshop on Rapid Continuous Software Engineering, 2018, pp. 1-7.

[29] Hasselbring W., Steinacker G. Microservice architectures for scalability, agility and reliability in e-commerce. IEEE International Conference on Software Architecture Workshops (ICSAW), 2017, pp. 243-246.

[30] Wohlin C. Guidelines for snowballing in systematic literature studies and a replication in software engineering, In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, 2014, article no. 38, 10 p.

[31] Garousi G., Garousi V. et al. Evaluating Usage and Quality of Technical Software Documentation: An Empirical Study. In Proc. of the 17th International Conference on Evaluation and Assessment in Software Engineering, 2013, pp. 24-35.

[32] Valdivia J.A., Lora-González A. et al. Patterns related to microservice architecture: a multivocal literature review. Programming and Computer Software, vol. 46, issue 8, 2020, pp. 594-608 / Вальдивия Х.А., Лора-Гонсалес А. и др. Паттерны микросервисной архитектуры: многопрофильный обзор литературы. Труды ИСП РАН, том 33, вып. 1, 2021 г., стр. 81-96. DOI: 10.15514/ISPRAS-2021-33(1)-4.

Information about authors / Информация об авторах

Victor M. NIÑO-MARTÍNEZ, Software Engineer Student. Research interests: Software Engineering, Software Architecture, and DevOps.

Виктор М. НИНЬО-МАРТИНЕС, студент-программист. Область научных интересов: разработка программного обеспечения, архитектура программного обеспечения и DevOps.

Jorge Octavio OCHARÁN-HERNÁNDEZ, Doctor in Computing Sciences, Professor at the School of Statistics and Informatics. Research interests: software engineering, software architecture, requirements engineering, API design

Хорхе Октавио ОЧАРАН-ЭРНАНДЕС, кандидат компьютерных наук, профессор факультета статистики и информатики. Область научных интересов: разработка программного обеспечения, архитектура программного обеспечения, разработка требований, разработкаAPI.

Xavier LIMÓN, Doctor of Artificial Intelligence, Associate Professor of the Statistics and Informatics Faculty. Research interests: Distributed Systems, Software Architectures, Multi-agent systems, Machine Learning.

Ксавье ЛИМОН, кандидат наук в области искусственного интеллекта, доцент факультета статистики и информатики. Область научных интересов: распределенные системы, архитектуры программного обеспечения, многоагентные системы, машинное обучение.

Juan Carlos PÉREZ-ARRIAGA, Master in Computer Science, Software Developer. Research interests include software architecture, software engineering, software metrics, software tools, software quality.

Хуан Карлос ПЕРЕС-АРРИАГА, магистр компьютерных наук, разработчик программного обеспечения. Область научных интересов: архитектура программного обеспечения, инженерия программного обеспечения, показатели программного обеспечения, программные инструменты, качество программного обеспечения.

DOI: 10.15514/ISPRAS-2023-35(1)-5

Influence of Belbin's Role on Database Design: An Exploratory Experiment

[1] *R. Aguilar, ORCID*: *0000-0002-1711-7016 <avera@correo.uady.mx>*
[2] *A. Peña, ORCID: 0000-0001-6823-2367 <adriana.pena@cucei.udg.mx>*
[1] *J. Díaz, ORCID: 0000-0002-3005-3432 <julio.diaz@correo.uady.mx>*
[1] *J, Ucán, ORCID: 0000-0002-1013-6396 <juan.ucan@correo.uady.mx>*

[1] *Universidad Autónoma de Yucatán,*
Mérida, México, 97000
[2] *Universidad de Guadalajara,*
Guadalajara, México, 44100

Abstract. Software process has been studied from various perspectives, among them, the human factor is one of the most important due to the intrinsic social aspect of the discipline. This study aims to explore the benefits of using Belbin's role theory in tasks —team and individual— related to the software development process, particularly in Database Design (DB) Design. In this paper two controlled experiments with students are presented. In the first experiment integrated teams with compatible roles identified in the students and teams integrated through a traditional strategy were compared, during the task of DB conceptual design. In the second experiment, individual students were the experimental subjects, the performance of the Belbin roles identified in them were compared, in the task of the DB logical design. The dependent variables in both experiments were the effort in the task, and the quality of the generated design. Results in the first experiment did not show significant differences in both variables, a possible limitation was the complexity of the task. The second experiment also did not show significant differences in the effort variable; however, in the variable related to the quality of the logical design, the monitor-evaluator role presented significant differences when compared with the other six identified roles; these results are consistent with previous studies identified in the literature. We plan to continue experimenting with other tasks in order to get a deeper understanding of applying the Belbin's theory in software process to accumulate experiences.

Keywords: software process; human factor; Belbin's roles; database design, quality

For citation: Aguilar R., Peña A., Díaz J., Ucán J. Influence of Belbin's Roles on Database Design: An Exploratory Experiment. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 73-86. DOI: 10.15514/ISPRAS-2023-35(1)-5

Влияние ролей Белбина на дизайн базы данных: исследовательский эксперимент

[1] *Р. Агилар, ORCID: 0000-0002-1711-7016 <avera@correo.uady.mx>*
[2] *А. Пенья, ORCID: 0000-0001-6823-2367 <adriana.pena@cucei.udg.mx>*
[1] *Х. Диас, ORCID: 0000-0002-3005-3432 <julio.diaz@correo.uady.mx>*
[1] *Х. Укан, ORCID: 0000-0002-1013-6396 <juan.ucan@correo.uady.mx>*

[1] *Автономный университет Юкатан,*
Мексика, 97000, Мерида
[2] *Университет Гвадалахары,*
Мексика, 44100, Гвадалахара

Аннотация. Программный процесс изучался с различных точек зрения, среди которых человеческий фактор является одним из наиболее важных в связи с присутствующим социальным аспектом. Это исследование направлено на изучение преимуществ использования ролевой теории Белбина в задачах – командных и индивидуальных, – связанных с процессом разработки программного обеспечения, особенно в проектировании баз данных (БД). В этой статье представлены два контролируемых эксперимента с участием студентов. В первом эксперименте сравнивались интегрированные команды с совместимыми ролями, определенными у студентов, и команды, интегрированные с помощью традиционной стратегии, во время решения задачи концептуального проектирования БД. Во втором эксперименте испытуемыми выступали отдельные студенты, и сравнивалиось выполнение выявленных у них ролей Белбина в задаче логического проектирования БД. Зависимыми переменными в обоих экспериментах были трудозатраты при выполнении задачи и качество созданного дизайна. Результаты в первом эксперименте не показали существенных различий по обеим переменным, возможным ограничением была сложность задачи. Второй эксперимент также не показал существенных различий в переменной трудозатрат; однако в переменной, связанной с качеством логического плана, роль наблюдателя-оценщика показала значительные отличия по сравнению с другими шестью идентифицированными ролями; эти результаты согласуются с предыдущими исследованиями, указанными в литературе. Мы планируем продолжить эксперименты с другими задачами, чтобы получить более глубокое понимание применения теории Белбина в программном процессе для накопления опыта.

Ключевые слова: процесс разработки программного обеспечения; человеческий фактор; роли Белбина; дизайн базы данных, качество

Для цитирования: Агилар Р., Пенья А., Диас Х., Укан Х. Влияние ролей Белбина на дизайн базы данных: исследовательский эксперимент. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 73-86. DOI: 10.15514/ISPRAS-2023-35(1)-5

1. Introduction

The study of development and management processes in Software Engineering (SE) has been developed considering various variables; however, the social aspect, intrinsic to this discipline highlights the human factor as a research topic [1]. In a recent article about the software process improvement, a categorization of the human factor is proposed [2]; where each team member should present a commitment with the assigned task according to his/her role, that is their responsibilities role. Even though, the role tasks achievement can be analyzed from an individual point of view, we have to consider that many tasks or activities are performed in a development team context. In [3] the author pointed out that the formation of a software development team is not an accident, but a complex process in which the team members establish work relations, get agreements on the project goals, and determine their functions as part of the team.

The knowledge and technical skills with which an individual contributes to the organization, according to his/her position, is the well-known functional role. On the other hand, the behavior of an individual in interrelation with colleagues is another type of role known as team role, a role with

not direct association with the required skills for a particular task at hand. However, the team role absence or presence has a significant influence in the project and the team success [4].

The purpose of this study is to explore whether Belbin's role theory can be useful to the manager of a software project, in the assigning the right personnel to tasks that are developed both as a team and individually. For its empirical validation, the study is contextualized in tasks related to the design process of a database.

The following section presents the theoretical framework that supports Belbin's role theory, as well as the process of designing a database. Section three presents a controlled experiment carried out with teams of students —as experimental subjects— in the conceptual design task of a database. The fourth section describes a second experiment, in this case, with individuals as experimental subjects in the logical design task of a database. Finally, section five presents the conclusions of the empirical study, as well as future work identified by the researchers.

2. Background

Among team role studies, the Belbin work [5, 6] is well-known among consultors and researchers, mainly because this theory offers a mechanism to identify the primary role a person can assume in a team, according to his/her behavior, as well as a balance and compatibility approach among team roles.

In SE, there are studies that underpin the Belbin team roles theory to explore benefits for the software development process and the software products [7]. Most of those studies are based on the software development team formation as an alternative to get compatible roles [8, 9]. A second group of studies are focus on individual task performance [10, 11].

It should be noted that the authors have carried out a set of linked experiments in recent years, both with development tasks [12, 13] and software management [14, 15].

2.1 Belbin team roles theory

The Belbin roles proposal presents three role categories: action oriented roles: Sharper, Implementer, and Completer/Finisher; person oriented roles: Chairman, Resource/Investigator, and Teamworker; and cognitive oriented or mental roles: Plant, Monitor/Evaluator, and Specialist [3]. In addition to this classification, the Belbin offers some identification mechanisms for the primary role a person can assume in a team, accordingly to that person behavior. We believe that the Belbin theory main contribution is the analysis of the interaction among roles, towards inside the team [16].

2.2 Database design task

The database design process, as part of an information system, is typically conceived as an abstraction process with different representation levels. According to [17] the most consensual process has three stages, and therefore, three representation levels: conceptual design, logical design, and physical design.

A database design has its origin in the data requirements specified in the requirement software development phase, which are transformed to a first level of abstraction – conceptual design – in which are represented the information resources of the organization, regardless the users or the applications. The most recognized data model for such abstraction is the Entity/Relation Model, based on the real world perception, with objects called entities and the relations among those entities [18].

A second level abstraction aims to transform the conceptual model by adapting it to the data model that supports the Database Management System (DBMS) that will be used for implementation — Logical Design.

In the twentieth century the relational systems dominated the market, and that is why it was selected for the second phase of the process; this model is fundamental for the modern DB technology [19]. It deals with three main information aspects: the data structure, the data manipulation, and the data integrity. The database relational model is based on the mathematical theory of relations, and data is logically structured in a relation represented as a table. Because the Entity/Relation and the Relational models share the same design basic principles, it is possible to apply a set of derived rules that allow transforming the conceptual model to a logical relational model [20].

Finally, the physical design has the purpose of the implementation, as efficiently as possible, of the logical model. For this process, the DBMS sub-language data definition is used.

3. Method for Experiment 1

One of the most used empirical methodologies in the field of SE is experimentation, specifically, experimentation in controlled environments [21]. This methodology helps us to identify and, when appropriate, to understand the possible relationships between factors and dependent variables, both parameters involved in software process. Among the characteristic elements of the controlled studies found in the literature, the use of groups of students as experimental subjects stands out. In [22] the authors indicate that this academic sample allows the researcher to obtain preliminary evidence to confirm or refute hypotheses that can be later contrasted in industrial contexts. The first experiment aims to explore, through the execution of a controlled experiment with students, the influence of the use of Belbin's Theory in the integration of development teams with members who present compatible roles, on the task of Database Conceptual Design.

3.1 Planning

In accordance with the purpose of our study, this experiment aims to comparing metrics related to the quality of the Database Conceptual Design, using integrated teams with Compatible Team (CT), as well as randomly integrated or Traditional Team (TT).

The first pair of statistical hypotheses uses as the dependent variable, a metric related to the software product, the quality of the Database Conceptual Design.

- H_{01}: The mean of the conceptual designs quality (CDQ) generated by the CT is the same as the mean of the quality of the conceptual designs generated by the TT.
- H_{11}: The mean of conceptual designs quality (CDQ) generated by the CT differs from the mean of the quality of the conceptual designs generated by the TT.

A second pair of statistical hypotheses were generated considering effort as the dependent variable, a metric related to the process of Database Conceptual Design.

- H_{02}: The mean of the effort invested by the CT is the same as the mean of the effort made by the TT in the Database Conceptual Design task.
- H_{12}: The mean of the effort made by the CT differs from the mean of the effort made by the TT in the Database Conceptual Design task.

For our study, factorial design with a source of variation, and two treatments for each of the two dependent variables is an appropriate experimental design.

3.2 Execution

The convenience sample used for the experiment consisted of 34 students who participated voluntarily, from the Software Engineering program of the Autonomous University of Yucatan. The participants were enrolled in the subject “Experimentation in Software Engineering” during the August-December 2019 semester. With this group of student volunteers, 17 development teams – experimental subjects – were formed, of which 8 teams were integrated with compatible roles (CT:

Compatible Teams) and the remaining 9 – control teams – randomly or traditionally (TT: Traditional Teams).

At the beginning of the experimental session, the requirements specification document was delivered and read, clarifying doubts regarding the specifications; requested to record the start time of the task. Likewise, instructions were given so that the teams, at the end of the task, record the time of completion, digitize the model and upload the generated Conceptual Model to the Institutional Learning Management System.

Table 1 illustrates the assignment of the teams to the treatments, as well as the data collected through the experiment.

Table 1. Data obtained in the first experiment

Team	Treatment	CDQ	Effort
I	CT	1.86	51
II	CT	1.25	41
III	CT	1.62	39
IV	CT	1.50	40
V	CT	1.56	42
VI	CT	1.42	39
VII	CT	1.47	30
VIII	CT	1.78	53
IX	TT	1.57	17
X	TT	2.32	42
XII	TT	1.62	38
XII	TT	1.31	33
XIII	TT	1.58	54
XIV	TT	1.89	39
XV	TT	1.51	51
XVI	TT	2.29	47
XVII	TT	1.58	38

3.3 Analysis

A descriptive and inferential statistical analysis was carried out with the information collected, using the statistical software "Statgraphics" to describe the behavior of the data, as well as to evaluate the statistical hypotheses previously defined.

Tables 2 and 3 present some of the most important measures of central tendency and variability for the dependent variables Conceptual Design Quality (CDQ) and Effort.

Table 2. Statistical summary to the CDQ variable

Treatment	#	μ	Median	Ơ
CT	8	1.5575	1.53	0.19616
TT	9	1.7411	1.58	0.35243

Table 3. Statistical summary to the effort variable

Treatment	#	μ	Median	Ơ
CT	8	41.875	40.5	7.25923
TT	9	39.888	39.0	10.9367

In order to visually compare the two treatments, boxplots were generated to analyze the dispersion and symmetry of both data sets. In Fig. 1 we can observe a lot of similarity in both treatments, from which there seems to be no difference for the Effort variable. In case of the CDQ variable, although the CT presented a more symmetric behavior, a visual difference cannot be distinguished with respect to the TT (see Fig. 2).

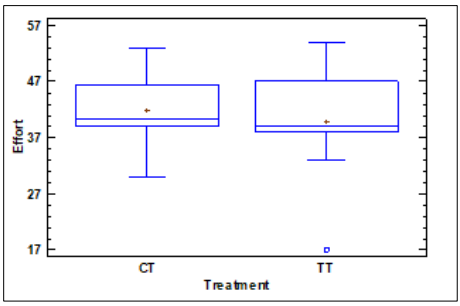

Fig. 1. Box plot for the Effort variable

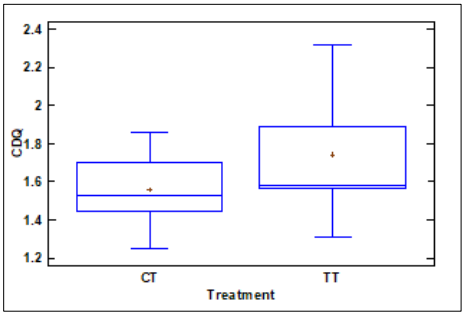

Fig. 2. Box plot for the CDQ variable

For the inferential analysis, the one-way analysis of variance was chosen, because it allows performing hypothesis tests to determine whether or not there are significant differences between the means of the values collected in the dependent variable. The analysis of variance [23] is a technique to build a statistical model that describes the impact of a single categorical factor on a dependent variable. The result of the ANOVA evaluation is illustrated in Table 4.

Table 4. ANOVA result for dependent variables

Dependent variable	F	p-value
CDQ	1.70	0.2125
Effort	0.19	0.6699

In both variables, the p-value of the F test is greater than 0.05, therefore, the null hypotheses H01 and H02 cannot be rejected; that is, in both cases there are no statistically significant differences between the means of the two treatments.

3.4 Model validation

It is important to mention that the ANOVA model has three associated assumptions that must be validated [23]. The assumptions of the model are: (1) experimental errors of its data are normally distributed, (2) there is no difference between the variance of the treatments, and (3) there is independence between the samples. The three assumptions were validated.

3.5 Results

Once the ANOVA model was validated, the results of the experiment showed that it was not possible to demonstrate the existence of significant differences, although on average the TC presents slightly lower metrics in the CDQ, and the values obtained from the TT show greater variability. In the case of the effort variable, the behavior of the data was similar in both treatments, so the null difference between them was verified with the analysis of variance.

4. Method for Experiment 2

The second controlled experiment was contextualized in a task performed individually; The experiment intends to explore the influence of Belbin's role —identified in the student— in the Database Logical Design task.

4.1 Planning

The factor considered in the study is defined as the role played by the subject in the Database Logical Design task. Nine treatments are identified, which correspond to the nine team roles proposed by Belbin. The first pair of hypotheses uses as dependent variable, a metric related to the Database Logical Design Quality (LDQ) generated by students.

- H_{01}: The means regarding the LDQ generated by each of the roles do not present differences.
- H_{11}: The means regarding the LDQ quality generated by each of the roles, differ in at least a two of them.

The second pair of hypotheses uses as dependent variable, the effort to complete the individual task of generating Database Logical Design.

- H_{02}: The effort made to generate the Database Logical Design for each of the roles does not present significant differences.
- H_{12}: The effort made to generate the Database Logical Design for each of the roles shows significant differences in at least one pair of them.

To evaluate the LDQ, Correctness as a factor was selected [24]. Regarding the second dependent variable Effort, operationally the Time – in minutes – used by a student to complete the task will be considered. In this case, it will be obtained through the difference between the task end time and the task start time. Factorial design with one source of variation, and nine treatments for each of the two dependent variables is the most appropriate experimental design for our study.

4.2 Execution

The convenience sample consisted of 33 of the 34 student volunteers who participated in the first experiment, one of the students reported sick for the second experimental session. All the students completed the self-perception instrument proposed by Belbin [5], in order to identify —by the researcher— their main role, thus ruling out that said information could represent a distractor in the execution of the experiment. It should be noted that this instrument does not include the role of Specialist, and in the case of the sample, no student was identified with the role of Resource Researcher.

Table 5 presents the total sample of the 33 experimental subjects distributed in the seven resulting treatments. For the analysis of the LDQ variable, three products were discarded because the digital files were not clear for their analysis. Likewise, for the Effort variable, two subjects with chairman role did not record the completion time of the task, so they were not considered.

Table 5. Sample for the second experiment

Treatment	Sample (#)		
	Treatment	**LDQ**	**Effort**
Plant	2	2	2
Teamworker	4	3	4
Chairman	4	4	2
Completer Finisher	8	7	8
Implementer	5	4	5
Sharper	5	5	5
Monitor Evaluator	5	5	5
Total	*33*	*30*	*31*

Prior to the execution of the experiment, a session was dedicated as a review of the subject related to Database Logical Design, studied in a subject from the immediately prior semester to the one the students were studying.

During the experimental session, the Database Conceptual Design was delivered, doubts about the model were clarified, and it was requested for them to record the start time of the task. Instructions were also given so that the subjects, at the end of the task, record the completion time, digitize the model and upload the generated Logical Model to the institutional Learning Management System.

At the end of the session, the experimental subjects delivered the designed logic model, these documents were the experimental objects used for the experimental analysis phase.

4.3 Analysis

As in experiment 1, a descriptive and inferential statistical analysis was performed with the information collected to describe the behavior of the data, as well as to evaluate the statistical hypotheses previously raised.

Table 6 presents some of the most important measures of central tendency and variability for the LDQ variable. We can see that the Monitor-Evaluator role presents the highest mean and, after the Plant role, it is the second treatment with the least variability. It is worth mentioning that both roles are classified as mental.

Table 6. Statistical summary to the LDQ variable.

Treatment	**#**	**μ**	**Median**	**O̓**
Plant	2	0.6489	0.6489	0.0272
Teamworker	3	0.6857	0.6667	0.1406
Chairman	4	0.7295	0.7253	0.1030
C-F	7	0.6920	0.6296	0.1235
Implementer	4	0.6172	0.6142	0.0400
Sharper	5	0.7574	0.8009	0.1483
M-E	5	0.9484	0.9506	0.0369

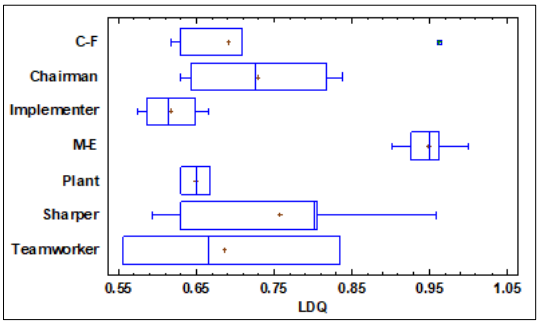

Fig. 3. Box plot for the LDQ variable

To visually analyze the behavior of the data, we generated a box plot. In Fig. 3, we can see the outdated behavior of the Monitor-Evaluator treatment data; this leads us to think that there is a possible difference with the other six treatments.

Table 7 lists some of the most important measures of central tendency and variability for the Effort variable.

Table 7. Statistical summary to the effort variable

Treatment	#	μ	Median	Ơ
Plant	2	14.0	14.0	5.6568
Teamworker	4	22.75	22.0	5.4390
Chairman	2	20.0	20.0	1.4142
C-F	8	19.37	18.9	4.8384
Implementer	5	15.6	15.0	3.5071
Sharper	5	15.4	13.0	4.8270
M-E	5	19.4	20.0	3.0495

We can observe that the social roles Teamworker and Chairman are those with the greatest effort in the task, being the Chairman the role with the least variability among the seven treatments. We also observed that the increased time allocated for the task occupied only one third of the time planned for the task.

The box plot in Fig. 4 illustrates the behavior of the treatments for the effort variable; visual analysis does not allow to identify significant difference in any subset of treatments.

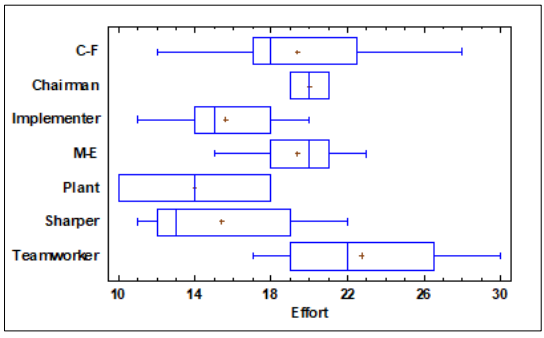

Fig. 4. Box plot for the Effort variable

In order to statistically evaluate the differences between the treatments of the LDQ and Effort variables, the one-way Analysis of Variance was applied. The result of evaluating the ANOVA is illustrated in Table 8.

Table 8. ANOVA result for dependent variables

Dependent variable	F	p-value
LDQ	4.49	0.0030*
Effort	1.85	0.1314

In the case of the LDQ variable, the p-value of the F test is less than 0.05; therefore, the null hypotheses H01 can be rejected. That is, we can affirm that there are at least one pair of treatments that present statistically significant differences between their means, with a 5% significance.

To identify which of the treatments are different, we generate the graph of means with confidence intervals according to the LSD test, which is illustrated in Fig. 5. This graph allows a visual and statistical comparison of the means of the treatments. As we can see, it seems that the only treatment that presents a lag with respect to the other six treatments is the one corresponding to the Monitor-Evaluator role.

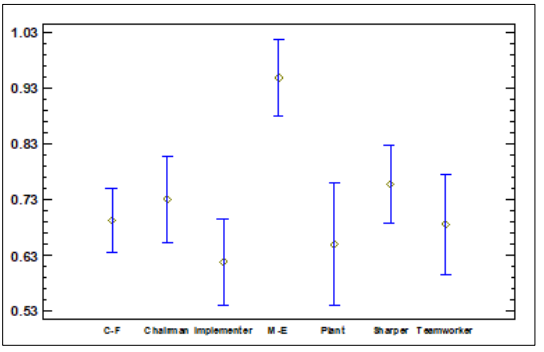

Fig. 5. Means plot for LDQ variable

The multiple-range test for the quality variable, using the LSD method, shows differences between six pairs of treatments, as illustrated in Table 9. This analysis matches with the visual analysis in Fig. 5.

Table 9. Multiple range test for LDQ variable

Contrast	Sig.	Difference	+/- Limits
Plant - ME	*	-0.299537	0.184751
Teamworker - ME	*	-0.262757	0.161264
Chairman - ME	*	-0.218904	0.14813
CF - ME	*	-0.256437	0.129299
Implementer - ME	*	-0.331173	0.14813
Sharper - MA	*	-0.191049	0.139658

In the case of the Effort variable, the p-value of the F test is greater than 0.05; therefore, we should reject the null hypothesis H_{02}. That is, we can affirm that there is no evidence, with a 5% significance, of differences between the means of the treatments.

4.4 Model validation

To correctly interpret what was obtained in the statistical analysis, the three assumptions of the ANOVA model were validated and we can affirm that the comments derived from Table 9 are valid.

4.5 Results

With the controlled experiment we found that for the LDQ variable, the Monitor Evaluator role presents significant differences, with a degree of better quality than the other six treatments (roles). On the other hand, for the task effort variable, the treatments did not show significant differences.

5. Conclusions and Future Work

This study aims to explore the influence of Belbin's role theory on tasks —team and individual— related to software development, particularly to Database Design.

In a first controlled experiment, the Conceptual Design Quality (CDQ) and the effort required for the aforementioned task were considered as dependent variables, such as treatments, integrated development teams based on Belbin's theory, and randomly integrated teams – called traditional equipment. The treatments did not show significant differences in both variables. With what was observed in the experiment raises the question if the task might not require the work of a team, but rather being designed as an individual task.

The second experiment, derived from reflection on the results obtained in the first, considered as dependent variables the Logical Design Quality (LDQ) and the effort required for the aforementioned individual task; the treatments considered in the study were the seven team roles identified in the participating experimental subjects, with the purpose of identifying if there is a particular role better for the task. The results of the experiment allowed to identify that the Evaluator Monitor role presented significant differences in the Logical Design Quality variable, having obtained a better quality degree than the other six participating roles in the experiment.

It is pertinent to comment that the Evaluating Monitor role is one of the two mental roles identified in the experiment. This result partially coincides with the results reported in [10] in the sense that it is one of the three identified roles – Specialist, Monitor-Evaluator and Finalizer – with a good contribution performance to the design task. It also coincides with what was commented in [7], regarding the fact that it is one of the two roles – Plant and Monitor-Evaluator – that present preference for the design phase. In the case of the effort variable, although the Plant role presented an average of lower time required for the task, the ANOVA did not identify significant differences in at least one pair of roles; possibly the task was not complex enough to require longer dedication times. This not allowed us to observe differences in the effort required between the participating roles.

With the results obtained in this study, the authors propose to continue with the development of controlled experiments in other tasks related to the software process, in order to generate knowledge about possible relationships between types of tasks and roles with better performance in them. In the case of database design, the lessons learned in both experiments, particularly in the first one (the team performance), make it possible to identify as future work, considering Database Design as an entire task – Conceptual Design, Logical Design and Physical Design – to explore the influence of Belbin's theory on development teams. This task is more complex, requires a longer period of development, and even within teams, it would allow mixing individual and team activities to achieve it.

References

[1] Juristo N., Moreno A. Basics of Software Engineering Experimentation. Kluwer Academic Publishers, 2001, 418 p.

[2] Morales N., Vega V. Factores Humanos y la Mejora de Procesos de Software. Propuesta inicial de un catálogo que guíe su gestión. Revista Ibérica de Sistemas y Tecnologías de Información, issue 29, 2018, pp. 30-42 (in Spanish).

[3] Humphrey W. Introduction to the Team Software Process. Addison Wesley, 2000, 463 p.

[4] Senior B. Team roles & Team performance: Is there 'really' a link? Journal of Occupational and Organizational Psychology, vol. 70, issue 3, 1997, pp. 85-94.

[5] M. Belbin. Management teams. Why they succeed or fail. John Wiley & Sons, 1981, 179 p.

[6] M. Belbin. Team roles at Work. Butterworth Heinemann, 1993, 152 p.

[7] Schoenhoff P.K. Belbin's Company Worker, The Self-Perception Inventory and Their Application to Software Engineering Teams. Master's thesis. Virginia Polytechnic Institute and State University, 2001, 161 p.

[8] Pollock M. Investigating the relationship between team role diversity and team performance in information systems teams. Journal of Information Technology Management, vol. 20, issue 1, 2009, pp .42-55.

[9] Abdulrahman A., Omar M. et al. An Analysis of Belbin Team Roles in Software Engineering Team. Journal of Engineering and Applied Sciences, vol. 12, issue 5 SI, 2017, pp. 6878-6883.

[10] Henry S., Stevens K. Using Belbin's leadership role to improve team effectiveness: An empirical investigation. The Journal of Systems and Software, vol. 44, issue 3, 1999, pp. 241-250.

[11] Estrada E., Peña A. Influencia de los roles de equipo en las actividades del desarrollador de software. Revista Electrónica de Computación, Informática, Biomédica y Electrónica, issue 1, 2013, 19 p. (in Spanish).

[12] Aguileta A., Ucán J., Aguilar R. Explorando la influencia de los roles de Belbin en la calidad del código generado por estudiantes en un curso de ingeniería de software. Revista Educación en Ingeniería, vol. 12, issue 23, 2017, pp. 93-100 (in Spanish).

[13] Aguilar R., Muñoz M. et al. Explorando la influencia de los roles de Belbin en la especificación de requisitos de software. Revista Ibérica de Sistemas y Tecnologías de la Información, issue 36, 2020, pp. 34-49 (in Spanish).

[14] Aguilar R., Díaz J., Ucán J. Influencia de la Teoría de Roles de Belbin en la Medición de Software: Un estudio exploratorio. Revista Ibérica de Sistemas y Tecnologías de la Información, issue 31, 2019, pp. 50-65 (in Spanish).

[15] Aguilar R., Gómez O. et al. Influencia de la Teoría de Roles en actividades de Gestión: Un experimento controlado con estudiantes de Ingeniería de Software. Revista Ibérica de Sistemas y Tecnologías de la Información, issue 39, 2020, pp. 131-146 (in Spanish).

[16] Aguilar R. A Strategy Assisted by Intelligent Virtual Environments. PhD Thesis. Polytechnic University of Madrid, 2008.

[17] De Miguel A., Piattini M., Marcos E. Diseño de Bases de Datos Relacionales. España, Ra-Ma, 1999, 576 p. (in Spanish).

[18] Silberschatz A., Korth H., Sudarshan S. Database Systems Concepts. 7th ed. McGrawHill, 2020, 1376 p.

[19] C.J. Date. An introduction to database systems. Pearson Education, 2004, 1040 p.

[20] Basili V., Selby R., Hutchens D. Experimentation in Software Engineering. IEEE Transactions on Software Engineering, vol. 12, issue 7, 1996, pp. 733-743.

[21] Genero M., Cruz-Lemus J., Piattini M. Métodos de investigación en ingeniería de software. España; Ra-Ma, 2014, 314 p. (in Spanish).

[22] Aguilar R., Diaz J. Procesos de Evaluación a la Calidad de la Primera Licenciatura en Ingeniería de Software en México. Revista Tecnología Educativa, vol. 3, issue 1, 2016, pp. 43-53 (in Spanish).

[23] Gutiérrez H., De la Vara R. Análisis y Diseño de Experimentos. México: McGraw Hill, 2012, 489 p. (in Spanish).

[24] McCall J.A., Richards P.K., Walters G. F. Factors in Software Quality, Volumes I, II, and III. US Rome Air Development Center Reports, US Department of Commerce, USA, 1977.

[25] Wohlin C., Runeson P. et al. Experimentation in Software Engineering. Springer. 2012, 260 p.

[26] Höst M., Regnell B., Wohlin C. Using Students as Subjects – A Comparative Study of Students and Professionals in Lead-Time Impact Assessment. Empirical Software Engineering, vol. 5, issue 3, 2000. pp. 201-214.

Information about authors / Информация об авторах

Raúl Antonio AGUILAR VERA, Professor of Software Engineering at the Faculty of Mathematics. He obtained a Doctor's degree from the Polytechnic University of Madrid, Spain (European Doctor Mention). His research work includes the areas of Software Engineering and Educational Informatics.

Рауль Антонио АГИЛАР ВЕРА, профессор программной инженерии математического факультета. Он получил степень доктора в Политехническом университете Мадрида, Испания. Его исследовательская работа включает области разработки программного обеспечения и образовательной информатики.

Adriana PEÑA, PhD in Computer Science, Professor. Research interests: Information Technology, Virtual Environments, Nonverbal Communication, Computer Engineering, Collaboration, CSCW, Groupware.

Адриана ПЕНЬЯ, кандидат компьютерных наук, профессор. Научные интересы: информационные технологии, виртуальные среды, невербальная коммуникация, вычислительная техника, совместная работа, групповое ПО.

Julio César DÍAZ-MENDOZA, Professor. Research interests: Relational Databases, SQL Programming, Software Development, Web Development.

Хулио Сезар ДИАС-МЕНДОЗА, профессор. Область научных интересов: реляционные базы данных, программирование на SQL, разработка программного обеспечения, веб-разработка.

Juan Pablo UCÁN PECH, PhD, Researcher. Research interests: Software Engineering, Teamwork, Web Engineering, Virtual Learning Environments.

Хуан Пабло УКАН ПЕЧ, кандидат наук, исследователь. Область научных интересов: программная инженерия, командная работа, веб-инженерия, виртуальные среды обучения.

DOI: 10.15514/ISPRAS-2023-35(1)-6

Scrumlity: A Quality User Story Framework

[1] *C. Tona, ORCID: 0000-0003-4492-3432 <tona.claudia@uabc.edu.mx>*
[2] *S. Jímenez, ORCID: 0000-0003-0938-7291 <samantha.jimenez@tectijuana.edu.mx>*
[1] *R. Juárez-Ramírez, ORCID: 0000-0002-5825-2433 <reyesjua@uabc.edu.mx>*
[3] *R. González Pacheco López, ORCID: 0000-0001-5979-2813 <rgonzalez@sdgku.edu>,*
[2] *Á. Quezada, ORCID: 0000-0001-5706-8047 <angeles.quezada@tectijuana.edu.mx>*
[4] *C. Guerra-García, ORCID: 0000-0002-9290-6170 <cesar.guerra@uaslp.mx>*

[1] *Universidad Autónoma de Baja California,*
Tijuana, México, 22390
[2] *Instituto Tecnológico de Tijuana,*
Tijuana, México, 22430
[3] *San Diego Global Knowledge University,*
San Diego, United States, 92101
[4] *Universidad Autónoma de San Luis Potosí,*
San Luis Potosí, México, 78600

Abstract. Scrum is one of many agile frameworks and is considered the most popular and widely adopted. Although Scrum presents several advantages, process and final product quality continue to be Scrum's main challenges. The quality assessment should be an essential activity in the software development process. Several authors have attempted to improve the quality of Scrum projects, changing some aspects of the framework, such as including new quality practices, a quality role, and quality processes. However, the quantification of quality is still a challenge. For that reason, the authors proposed a framework called Scrumlity, which was defined in a previous study. This framework proposes a change to Scrum, including a quality role and some artifacts to evaluate quality through a complete execution of a Sprint. In this study, the authors add a User Story Quality assessment to the framework. The User Story Quality Assessment includes over 250 analyzed User Stories. Results obtained after this experiment indicate the importance of executing a User Story Quality Assessment and that Scrum Team members are willing to accept adding this to the framework.

Keywords: Scrum; agile frameworks; quality assessment; User Story Quality Assessment; Scrumlity

For citation: Tona C., Jímenez S., Juárez-Ramírez R., González Pacheco López R., Quezada Á., Guerra-García C. Scrumlity: A Quality User Story Framework. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 87-100. DOI: 10.15514/ISPRAS-2023-35(1)-6

Scrumility: фреймворк для оценки качества пользовательских историй

[1] *К. Тона, ORCID: 0000-0003-4492-3432 <tona.claudia@uabc.edu.mx>*
[2] *С. Хименес, ORCID: 0000-0003-0938-7291 <samantha.jimenez@tectijuana.edu.mx>*
[1] *Р. Хуарес-Рамирес, ORCID: 0000-0002-5825-2433 <reyesjua@uabc.edu.mx>*
[3] *Р. Гонсалес Пачеко Лопес, ORCID: 0000-0001-5979-2813 <rgonzalez@sdgku.edu>,*
[2] *А. Кесада, ORCID: 0000-0001-5706-8047 <angeles.quezada@tectijuana.edu.mx>*
[4] *С. Герра-Гарсия, ORCID: 0000-0002-9290-6170 <cesar.guerra@uaslp.mx>*

[1] *Автономный университет Нижней Калифорнии (UABC),*
Мексика, 22390, Нижняя Калифорния, Тихуана
[2] *Тихуанский технологический институт,*
Мексика, 22414, Нижняя Калифорния, Тихуана
[3] *Университет глобальных знаний Сан-Диего,*
США, 92101, Сан-Диего,
[4] *Автономный университет Сан-Луис-Потоси,*
Мексика, 78000, SLP, Сан-Луис-Потоси

Аннотация. Scrum – один из многих гибких фреймворком, наиболее популярным и широко распространенным. Хотя Scrum имеет несколько преимуществ, его главной проблемой остается качество процесса и конечного продукта. Оценка качества должна быть важным элементом в процессе разработки программного обеспечения. Несколько авторов пытались улучшить качество проектов Scrum, изменив некоторые аспекты фреймворка, такие как включение новых методов обеспечения качества, роль качества и процессы обеспечения качества. Однако количественная оценка качества все еще остается проблемой. По этой причине авторы данной статьи предложили фреймворк под названием Scrumlity, который описывался в предыдущем исследовании. В этом фреймворке Scrum расширяется, включая добавление роли качества и некоторых артефактов для оценки качества посредством полного выполнения спринта. В описываемом исследовании авторы добавляют к фреймворку оценку качества пользовательских историй. Оценка качества пользовательских историй включает более 250 проанализированных пользовательских историй. Результаты, полученные после этого эксперимента, указывают на важность выполнения оценки качества пользовательских историй и на то, что члены команды Scrum готовы принять ее добавление во фреймворк.

Ключевые слова: Scrum; гибкие фреймворки; оценка качества; оценка качества пользовательских историй; Scrumlity

Для цитирования: Тона К., Хименес С., Хуарес-Рамирес Р., Гонсалес Пачеко Лопес Р., Кесада А., Герра-Гарсия С. Scrumility: фреймворк для оценки качества пользовательских историй. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 87-100. DOI: 10.15514/ISPRAS-2023-35(1)-6

1. Introduction

According to the *14th Annual State of Agile Report* [1], the most used agile framework is Scrum. Scrum was designed to increase development speed, focusing on creating a product that generates stakeholder value. Although Scrum presents several advantages such as incremental project deliveries, closer contact, constant feedback with stakeholders, and tolerance for changing requirements; however, several authors have suggested that one of the main problems in Scrum, similar to that of other Agile frameworks, is quality throughout the framework as well as product quality [2-4].

According to the authors in [5], software requirements are defined as a statement that describes a particular feature of the software product and is recorded using a specific technique during requirements engineering. The User Story is the most common way of writing requirements when

using agile frameworks and has the following structure: *As a < type of user>, I want <some goal>, so that <some benefit>*.

User Stories, as requirements, have the potential to break down a complex system into small, user-oriented pieces, which can be implemented independently [6]. Its quality affects communication and coordination in a project and therefore plays a critical role when it comes to an understanding of how User Stories impact the daily work of a software team; their structure, granularity, and understanding are interesting aspects [7]. However, Agile requirements are by definition incomplete, not specific, and might be ambiguous when initially specified. User Stories are often incomplete or poorly defined, so misunderstandings or dependencies remain unpredictable [8], which is why the requirements quality assessment should be an essential step in the software development process.

Despite User Stories' popularity in the Agile industry, many methods to assess and improve User Story quality are limited. Some of the existing approaches employ highly qualitative metrics, such as the acronym I.N.V.E.S.T. which helps remember a set of criteria that can be used to assess the quality of a User Story. The meaning of this acronym is [9]: Independent, Negotiable, Valuable, Estimable, Scalable, and Testable. Additionally, good practices for quality in agile requirements established by Heck *et al.* [10] consider three different approaches to a User Story: Completeness, Uniformity, Consistency, and Correctness.

Researchers stated that in most organizations' quality aspects are not considered in the Scrum framework due to constant deliveries [11], and the quantification of quality parameters is still challenging.

The authors proposed an agile framework based on quality assurance as a possible solution. This framework suggests an adaptation to Scrum, called Scrumlity, where the main idea is the incorporation of a quality role and some artifacts which aim to evaluate quality before, during, and after the development process. Scrumlity seeks to improve a project's quality, but the previous study only focuses on describing the methodology's acceptance [12]. Scrumlity includes and promotes the existence of a Scrum Quality Owner role, a modified Definition of Ready artifact, a Quality Burn-up Chart template, a modified Definition of Done artifact, and a modified template for User Stories. The Quality Owner has several responsibilities such as: promoting code quality, defining quality processes, assuring that the Definition of Done considers quality software attributes, collaborating in the construction of the Product Backlog by adding a possible technical solution to each User Story, monitoring and generating the Quality Burn-up Chart based on Quality Points and to approve or deny the Sprint outcome in collaboration with the Product Owner. The authors took it forward in extending Scrumlity by adding a User Story Quality Assessment using the Quality User Story (QUS) framework originally proposed in [13] that considers 13 attributes that determine the quality of User Stories [14].

The rest of this paper is organized as follows: Sections 2 and 3 present background information related to Scrum and User Stories, and Section 4 details the related work. Scrumlity is presented in Section 5. Section 6 describes the experiments, sample, and setup that were performed. Section 7 presents the results. Finally, section 8 concludes the study.

2. Scrum Overview

The framework defines three specific roles within the Scrum Team: The Product Owner, the Scrum Master, and the Developers [6]. The main objective of the Product Owner is to define User Stories and be responsible for what will be developed and in what order. The Scrum Master has the responsibility of eliminating team impediments and embracing Scrum values, principles, and good practices; and the developers' responsibilities are: creating a plan for the Sprint, estimating the Sprint Backlog, instilling quality by adhering to a Definition of Done, and adapting their plan each day toward the Sprint Goal [15].

The Sprint is the heartbeat of Scrum, and it is a container for all other events that are mentioned below. Sprint planning is where the Product Owner determines the set of User Stories that should be worked on in the next print and is where a Sprint goal is defined [16]. The Daily Scrum meeting is a 15-minute event for the development team. This meeting aims to inspect progress toward the Sprint Goal [17]. The purpose of the Sprint Review is to inspect the outcome of the Sprint and determine future adaptations [18]. The Sprint Retrospective is where the team assesses its work and processes, and the Scrum Team generates action items for continuous improvement to increase quality and effectiveness.

Every project has a Product Backlog a prioritized list of User Stories; the Product Owner is the only person who has the authority to manage this artifact [17]. The Sprint Backlog is a subset of User Stories of the Product Backlog that indicates a plan by and for the Developers. It demonstrates the work the developers plan to accomplish during the Sprint to achieve the Sprint Goal [15].

It is essential to mention that when a Product Backlog item or increment is described as "Done," everyone in the Scrum team must understand what "Done" means. That is why a Definition of Done (DoD) artifact is created. This artifact includes code guidelines, team agreements, and criteria used to assess when the sprint outcome is complete.

3. User Stories

A User Story is an independent, negotiable, valuable, estimable, small, and testable requirement [19]. A User Story template is structured in the following way: *"As [the WHO], I want/need/can/would like [the WHAT], so that [the WHY]."* User Stories inherently allow addressing the three fundamental elements of requirement engineering: WHO wants the functionality, WHAT functionality end-users or stakeholders wish the system to provide, and the reason WHY the end-users or stakeholders need the system for [20, 21].

According to the Standish Group [22], the primary problems in requirements engineering were incompleteness, poor requirement specification, poor quality requirements, and communication flaws between the project team and the stakeholders. While the Agile Manifesto [23] and agile frameworks like Scrum try to mitigate these problems by embracing User Stories, it is necessary to ensure that these User Stories are of sufficient quality. However, a User Story should be defined considering team agreements established to provide the definition of a Product Backlog with high quality. A common practice is the creation of a Definition of Ready. This artifact represents the minimum criteria to be considered before a user story can be regarded as fit for work by the developers in the scrum team [24].

4. Related Work

The authors in [25] conducted an online questionnaire survey to collect data to compare Agile methods with software quality affecting factors. They considered three types of software qualities: Quality of Design, Quality of Performance, and Quality of Adoption. As a result, the authors identified 13 software quality affecting attributes. They concluded that the main advantage of agile techniques is customer satisfaction and that it welcomes user requirements changing at any phase.

Hanssen *et al.* [4] propose ScrumSafe. ScrumSafe assumes that the quality of software projects is always affected because a Scrum team is supposed to be self-sustained, not having to rely on an external quality management or assurance function like Quality Assurance (QA) manager. The research suggested integrating a Quality Assurance role and defined specific tasks that this new role should handle.

Lucassen *et al.* [14] suggest a Quality User Story framework (QUS), a set of 13 quality attributes that User Story writers should achieve. Given that User Stories are a controlled language, the QUS framework's attributes are classified into three categories: Syntactic, Semantic, and Pragmatic.

Jimenez *et al.* [19] propose a framework for assessing the internal quality of User Stories to improve the external quality of the project deliveries. The authors evaluated quality from internal and external perspective. Internal quality assessment was based on the grammatical structure of the User Stories, and the external quality considered the functionality and usability of the product deliveries. The findings suggested that the presence of adjectives in User Stories improves the usability and correctness of the product and is related to the developer's experience.

5. Scrumlity

According to [26] quality has four dimensions: specification (QD1), design (QD2), development (QD3), and conformance (QD4). The specification dimension refers to the collection of requirements, the definition of project scope, delimitation of time, identification of safety aspects, and evaluation of security aspects. Design dimension refers to how well the product is designed; it includes software architecture, database design, user interface design, among others. Development dimension includes taking care of the most common software development activities such as screen development, library linking, report development, and unit test plan development, among other activities. Finally, the conformance dimension refers to how well quality is built into a product through the above three dimensions.

Table 1. Software Quality Attributes in Quality Dimensions

Attribute	QD1	QD2	QD3	QD4
Functionality	*	*****	*	*
Reliability				*****
Usability	*	*****	*****	*****
Efficiency			*****	*****
Maintainability		*****	*****	
Portability	*	*****	*****	*****
Verifiability				*****
Integrity		*****	*****	
Testability			*	*
Expandability		*	*	*
Flexibility		*	*	
Reusability		*	*	
Interoperability		*	*	*
Security	*	*	*	*
Compatibility		*	*	

Table 1 demonstrates the relationship of software quality factors in agile frameworks with the quality dimensions. Table 2 indicates the Scrum artifact, the process where this artifact takes place, and the quality dimensions in which the artifact should be considered.

Table 2. Artifacts and Processes in Quality Dimensions.

Artifacts	Process or Event	Quality Dimension
Product Backlog	Sprint Planning	Specifications, Design
Sprint Backlog		
Burndown Chart, Board	Daily meeting	Development
	Sprint	
Partial Product	Sprint Review	Conformance
Action Items	Sprint Retrospective	Specifications, Design

Before a Sprint starts, the Product Owner defines a set of User Stories that need to be implemented to improve or construct a product. During the sprint planning meeting, the Product Owner and the rest of the Scrum Team define the User Stories that should be worked on during the next Sprint and compile the Sprint Backlog. The quality of these two artifacts is part of the specification and design dimensions.

According to the previous table, the factors that should be considered to measure the quality in these dimensions are functionality, usability, portability, and security. The burn-down chart displays team velocity, which in turn represents the story points that have been completed every day during the active sprint. The Scrum board helps the team provide visibility on task status. These task callouts are part of the sprint and daily meetings, which is why the dimension considered is development, and it means that factors considered to ensure quality are functionality, usability, maintainability, portability, integrity, expandability, flexibility, reusability, interoperability, security, and compatibility.

The product increment, a functional prototype delivered to the stakeholders, is part of the Sprint review meeting. The quality of this product increment is the responsibility of the conformance dimension. Lastly, during the Sprint Retrospective, the Scrum team defines an improvement plan with action items.

5.1 Scrum Quality Owner role

This study proposes a role hereafter known as the Scrum Quality Owner (QO). The QO has the responsibility to define and implement quality processes, promote code quality, and ensure that DoD considers quality software attributes. Ideally, this role should require previous experience with systems design, systems architecture, and systems modeling. The QO should collaborate with the PO, adding technical perspective to every User Story. Consequently, the PO and the QO would jointly build the Product Backlog. Also, the QO would be the person who is responsible for monitoring and generating the Quality Burn-up Chart which considers quality attributes by User Story. Lastly, this role would collaborate with the PO to assess Sprint's outcome. In summary, the QO would be the person who embraces the quality practices and principles that ensure quality during an active Sprint.

5.2 Scrumlity Process

Scrumlity workflow starts with the Scrum Team working on the Definition of Ready (DoR) in which the team specifies certain preconditions and agreements that must be met before a User Story can be accepted into a new Sprint [24]. One of the main aspects that must be included in this artifact is the definition of the quality assessment of User Stories. Part of this set of agreements will describe each quality attribute, to make sure that every team member understood them, if necessary, it is possible to consider adding examples of each attribute.

The PO will be responsible for defining User Stories under the quality attributes. In such a way, the PO will assess the User Story quality in order to achieve the 13 attributes before it becomes part of the Sprint Backlog. Ready in this context means the User Story is defined with high quality and is sufficiently prepared that a team can start to work on it.

The workflow continues with the QO and the PO collaborating on the definition of the Product Backlog, but each role focuses on different goals. The PO has the stakeholders' approach to defining User Stories; meanwhile, the QO focuses on software quality, through the definition of systems design, systems modeling, specification of design patterns to implement, sequence diagrams, UML diagrams, or any other technical artifact. The definition of a User Story would include a technical perspective with a focus on quality to support its development. The QO would work together with the PO and the SM to define the DoD and ensure that this artifact includes good practices and principles that consider software quality attributes at a technical level, such as some of the attributes described in Table 1. During an active Sprint, and during each daily meeting, the QO will ask each team member how many of the ten quality factors were considered on each completed User Story.

The QO would be empowered to generate the Quality Burn-up Chart, like how the SM generates the Burn-down Chart to report the results achieved at the end of each Sprint. The SM will focus on

reporting completed Story Points during the last Sprint; meanwhile, the QO will focus on reporting the Quality Story Points achieved by every User Story of the Sprint Backlog.

At the end of the Sprint, the PO and the QO will review and verify the product increment during the Sprint Review. Since one of the responsibilities of the QO is to define the technical aspects of a User Story, Scrumlity empowers QOs to decide if the outcome meets quality expectations at a technical level or not. That is how these two roles will be responsible for approving or denying the delivered product increment.

Figure 1 shows the traditional Scrum workflow with the modifications of Scrumlity, indicating the phases where the QO will interact.

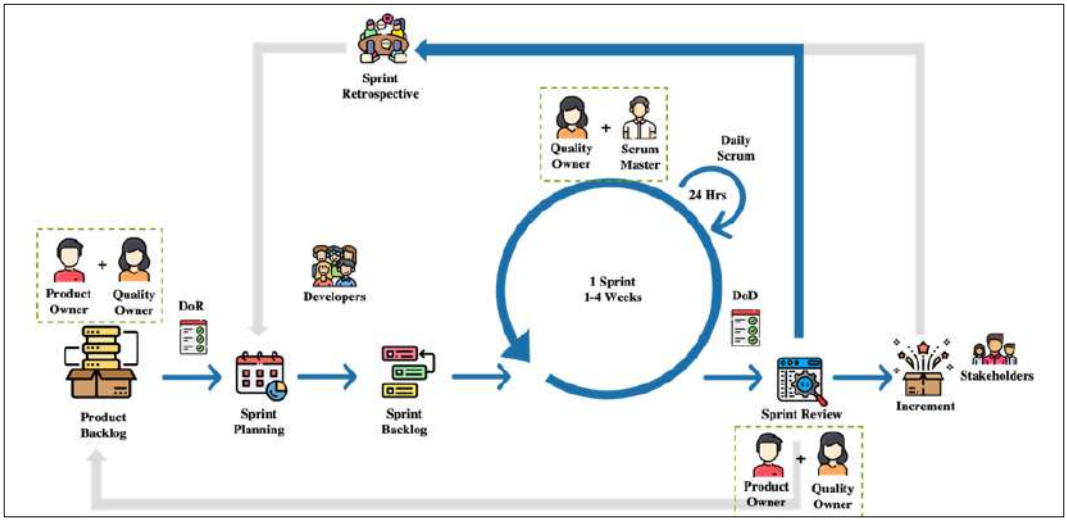

Fig. 1. Scrumlity Framework

5.3 Scrumlity Artifacts

5.3.1 Definition of Ready

Conceptually, the DoR is a checklist of the agreements that the Product Owner is expected to successfully comply with before they can declare the User Story is ready to be added to a Sprint Backlog [27]. The goal of the DoR is to identify defects in the User Story before work has started, thereby reducing the likelihood of defects early on. User Stories that are "ready" are clear, concise, sized appropriately for a Sprint, and most importantly, actionable [27]. The authors propose the DoR as an artifact to define the quality attributes to assess the quality of User Stories, so any team member would have access to this information. However, the PO will oversee the execution of this assessment for every User Story before it becomes part of the Sprint Backlog.

5.3.2 Product Backlog

The authors approach this proposal with two different objectives: the first one is that the User Story considers a technical perspective of a possible solution which might be represented by sequence diagrams, UML diagrams, other software architecture designs, schema designs, API contracts, database designs, etc.

And the second approach is to consider adding a checklist within the template to conduct a quality assessment of the User Story executed by the PO to ensure it meets the DoR. According to the authors in [13], the 13 quality attributes selected to assess a User Story are:

- Well-formed: The core text of the US needs to include at least a role and a goal.
- Atomic: The US should consider only one goal.
- Minimal: The US should contain a role, a goal, and preferably some benefit.
- Conceptually sound: The goal and benefit parts of a US play a specific role.
- Problem-oriented: The US should specify only the problem, not the solution to it.
- Unambiguous: The US avoids terms or abstractions that lead to multiple interpretations.
- Full-sentence: The US should read like a full sentence, without typos or grammatical errors.
- Estimable: As a US grows it increases its complexity, and it becomes more difficult to accurately estimate the required effort.
- Unique: The US is unique when no other US in the same project is (semantically) equal or too similar.
- Conflict free: The US should not conflict with any other US in the Product Backlog.
- Uniform: The US has a format that is consistent with the rest of the USs in the same set.
- Independent: USs should not overlap in concept and should be schedulable and implementable in any order.
- Complete: Implementing a set of USs should lead to a feature-complete application.

Table 3 represents a minimum set of requirements for user stories according to Scrumlity.

Table 3. User Story Template

Story Points	**Priority**									
User Story	As a <*type of user*>, I want <*goal*>, so that <*some benefit*>.									
Acceptance Criteria	Given *[Preconditions or Initial Context]*, When *[Event or Trigger]*, Then *[Expected output]*									
Testing Criteria	*A high-level check of the acceptance criteria.*									
User Story Quality Assessment										
Atomic	☐	**Full sentence**	☐	**Uniform**	☐	**Problem-oriented**	☐			
Minimal	☐	**Estimable**	☐	**Independent**	☐	**Conflict-free**	☐			
Technical description										
Description	*Technical description of how to achieve the User Story goal. Such as, sequence diagrams, UML diagrams, architecture design, schema design, API contract, etc.*									
Main Flow	*The sequence of steps to achieve the objective of the User Story.*									
Alternative Flow	*A different sequence of steps to achieve the objective of the User Story.*									
Exception Flow	*A sequence of events that does not allow to achieve the objective of the User Story.*									

5.3.3 Definition of Done

The DoD should be focused on quality guidelines and regulations. This means, that it is necessary to specify that every User Story should be assessed with each quality attribute to change the status to "done". It is recommended to specify a brief description of every quality attribute in the DoD artifact. That is how the QO collaborates with the PO and the SM to define all the agreements that would be included in the DoD; as consequence, the Scrum team would have a quality perspective to work completed during the Sprint.

Definition of Done and Definition of Ready act as social contracts in agile teams. The development team is responsible for meeting the DoD; while Product Owners are responsible for making sure work items meet the DoR [27].

5.3.4 Quality Burn-up Chart

The Burn-up Chart is a proposal based on quality points. Quality points represent the quality attributes achieved by every User Story.

The maximum attributes to accomplish are ten points by User Story, this means that if there are six User Stories to complete during the current Sprint, the team will have 60 quality points to reach by the end of the Sprint. The considered attributes are the conformance attributes mentioned in Table 1.

As agility allows adjusting to changing technology and needs, and to support this capability, the authors suggest that the Quality Burn-up Chart could be generated through a template. The template could be a spreadsheet with a list of User Stories planned in the current Sprint with a checklist of the ten quality attributes for each User Story. In such a way, it is possible through the checked attributes to automate the calculation of the total quality points achieved at the end of the Sprint and generate the Quality Burn-up Chart The QO will be in charge of updating this template with the Quality Points achieved per day through monitoring with the Scrum team during the daily meeting.

The following equation should be considered to calculate total Quality Points by Sprint.

$$TQP = (PUS)(QA)$$

where:

TQP = Indicates total Quality Points to be completed by the end of the Sprint.

PUS = Indicates the number of planned User Stories in the actual Sprint.

QA = Indicates the number of quality attributes considered and defined in the DoD artifact.

6. Experiments

6.1 Scrumlity Acceptance Experiment

A previous experiment was executed to evaluate the framework's acceptance [12]. The experiment followed a sample (for convenience) of 31 participants. Six of these 31 participants were actively employed in software development companies, and the rest of the participants were undergraduate students enrolled in a Computer Sciences program. The sample was divided into two groups: novices and practitioners. The novice participants attended Scrum training. After training, the participants started working on their projects using the Scrum framework. The practitioner participants did not take any Scrum training because they already had prior experience with Scrum. Both groups received Scrumlity training and executed 2 Sprints with this adapted framework.

The framework's acceptance was evaluated to measure the acceptance of the framework the participants answered a five-point Likert scale instrument. The results suggested that participants accepted the framework satisfactorily. Most participants agreed that the framework benefits their organization and makes software development more efficient, and they would like to use this framework in the future.

The qualitative analysis proposed the implementation of a template for the burn-up chart and a manual or guidebook with the description of quality criteria, to maintain agility and make it easier to adopt the framework. Lastly, most participants rated Scrumlity higher than Scrum.

6.2 User Story Quality Assessment Experiment

For this experiment, the authors considered the suggestions of the experiment implemented in [12] such as more detail on the attributes' description and the implementation of a template to though preserving framework agility.

6.2.1 Sample

This study follows a sample (for convenience) of 78 participants (14 females and 64 male). Members were 22.69 years old in average (min=21, max=38, sd=2.59). The participants were undergraduate students enrolled in a Computer Science program. All members were attending a software engineering course. Twenty-eight of 78 participants have less than 12 months of development experience, 34 of 78 have between one and three years of experience, and 16 of 78 participants have more than three years of software development experience. Related to Agile experience 51 of 78 participants have less than 12 months of experience, 25 of 78 participants have between 12 and 36 months of Agile experience, and only two participants have more than three years of experience. The sample was organized into 14 Scrum teams.

6.2.2 Process

A User Story workshop was imparted to the participants. Once the workshop was over, the 14 teams began generating User Stories. The definition of the User Story was supported by the DoR that included a description of the quality attributes proposed by [14]. Also, the teams used the User Story template mentioned in Section 5.3. The template had a checklist considering the 13 attributes of quality for User Stories, which had to be verified by the PO once the creation of the User Story was complete. To avoid subjectivity in the evaluation of user stories each team was assigned another team's User Stories to evaluate their quality against the proposed criteria. If the User Story meets the quality attribute the team will assign a "1" to this attribute, otherwise, the team will assign a "0". This process repeats until all the quality attributes are evaluated. At the end of the experiment, the team will have two quality assessments of their User Stories. One made by the PO (internal assessment) and an external evaluation carried out by another team.

7. Results

Table 4 shows the results of the quality assessment executed. It shows the number of User Stories defined by the team, the average error margin expressed as a percentage obtained through the evaluation performed by another team in comparison with the internal assessment of each team (AVGE), and the standard deviation of each team's measurements. The AVGE was obtained by averaging the result of the evaluation carried out by the PO during the Product Backlog definition process minus the average of the evaluation carried out by an external team. Five of the 14 teams obtained less than a 10% error margin between the internal assessment executed by the PO (while the team was defining their User Stories), and the assessment executed by another team over the Product Backlog generated. Five of the 14 teams obtained between 10% and 15% of error margin in the quality of their User Stories, and just four teams got more than 15% of error margin between the internal and external evaluation of User Stories.

Another factor that the authors examined was the acceptance of conducting a User Story quality assessment. The results are shown in Table 5. The participants rated the quality assessment of User Stories within the Scrumlity framework on a five-point Likert scale from strongly disagree to strongly agree. The results suggest that 60 participants considered that including a quality assessment of User Stories will benefit the execution of their projects; seven participants gave neutral responses, and just one participant thought that there would be no benefit.

Table 4. User Story Quality Assessment

Team	User Stories Qty.	AVG (%)	Std.
T01	19	16.60%	0.123
T02	20	12.31%	0.094
T03	20	11.15%	0.052
T04	20	11.54%	0.130
T05	25	9.54%	0.055
T06	20	21.92%	0.206
T07	20	3.46%	0.063
T08	20	26.54%	0.133
T09	21	2.93%	0.061
T10	28	11.54%	0.129
T11	5	20.00%	0.116
T12	20	4.62%	0.046
T13	20	3.85%	0.046
T14	20	10.77%	0.063

Table 5. User Story Quality Assessment

	1	2	3	4	5
Assessing the quality of USs helps to write better US.	0	1	13	32	32
Assessing the quality of a US improves the quality of future USs.	0	2	9	35	32
It is important to assess the quality of a US before software development starts.	0	1	6	27	44

8. Conclusion and Future Work

This framework is an evolution of Scrum that includes and promotes the existence of a Quality Owner role, a modified Definition of Ready artifact, a Quality Burn-up Chart template, a modified Definition of Done artifact, and a modified template for User Stories. The Quality Owner has several responsibilities such as: promoting code quality, defining, and implementing quality processes, collaborating in the construction of the Product Backlog by adding a technical perspective solution to each user story, monitoring and generating the Quality Burn-up Chart, and assessing the Sprint outcome in collaboration with the Product Owner. Framework acceptance was evaluated because improving process and software product quality were motivators though preserving framework agility was equally important. The findings suggested that the participants accepted the framework satisfactorily. Most participants agreed that executing a quality assessment of User Stories under Scrumlity benefits their organization and the execution of their projects. In conclusion, the addition of the User Story Quality Assessment to Scrumlity was widely accepted. Further research directions exist that future work should address. As the study was mainly applied to undergraduate students, it is necessary to execute an experiment with Scrum Teams in the industry to gain better feedback on the Scumlity proposal.

References / Список литературы

[1] Digital.ai. 14th Annual State of Agile Report. Available at: https://info.digital.ai/rs/981-LQX-968/images/SOA14.pdf, accessed September 16, 2021.

[2] Khalane T., Tanner M. Software quality assurance in Scrum: The need for concrete guidance on SQA strategies in meeting user expectations. In Proc. of the 2013 International Conference on Adaptive Science and Technology, 2013, pp. 1–6.

[3] Sirshar M., Nadeem T., Abiha U. Software Quality Assurance in SCRUM: Implementing SQA strategies in meeting user expectations. Preprints, 2019, 2019120117, 6 p.

[4] Hanssen G.K., Haugset B. et al. Quality Assurance in Scrum Applied to Safety Critical Software. Lecture Notes in Business Information Processing, vol. 251, 2016, pp. 92-103.

[5] Murtazina M.S., Avdeenko T.V. Ontology-Based Approach to the Requirements Engineering in Agile Environment. In Proc. of the XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE), 2018, pp. 496-501.

[6] Schwaber K., Sutherland J. The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. November 2017. Available at: https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf, accessed September 16, 2021.

[7] Liskin O., Pham R. et al. Why We Need a Granularity Concept for User Stories. Lecture Notes in Business Information Processing, vol. 179, 2014, pp. 110-25.

[8] Lucassen G., Dalpiaz F. et al. The Use and Effectiveness of User Stories in Practice. Lecture Notes in Computer Science, vol. 9619, 2016, pp. 205–222.

[9] Wake B. INVEST in good stories, and SMART tasks. August 17, 2003. Available at: https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/, accessed September 16, 2021.

[10] Heck P., Zaidman A. A Quality Framework for Agile Requirements: A Practitioner's Perspective. arXiv:1406.4692, 2014, 11 p.

[11] Jain P., Sharma A., Ahuja L. A customized quality model for software quality assurance in agile environment. International Journal of Information Technology and Web Engineering, vol. 14, issue 3, 2019, pp. 64-77.

[12] Tona C., Juarez-Ramirez R. et al. Scrumlity: An Agile Framework Based on Quality Assurance. In Proc. of the 9th International Conference in Software Engineering Research and Innovation (CONISOFT), 2021, pp. 88-96.

[13] Lucassen G., Dalpiaz F. et al. Forging high-quality User tories: Towards a discipline for Agile Requirements. In Proc. of the IEEE 23rd International Requirements Engineering Conference (RE), 2015, pp. 126-135.

[14] Lucassen G., Dalpiaz F. et al. Improving agile requirements: the Quality User Story framework and tool. Requirements Engineering, vol. 21, issue 3, 2016, pp. 383-403.

[15] Scrum Revealed Training Book. 2nd ed. International Scrum Institute, 2017, 55 p.

[16] Hart M.A. Agile Product Management with Scrum: Creating Products that Customers Love by Roman Pichler. Journal of Product Innovation Management, vol. 28, issue 4, 2011, pp. 615-615.

[17] Schwaber K., Beedle M. Agile Software Development with Scrum 1st. Pearson, 2001, 176 p.

[18] Srivastava A., Bhardwaj S., Saraswat S. SCRUM model for agile methodology. In Proc. of the International Conference on Computing, Communication and Automation (ICCCA), 2017, pp. 864-869.

[19] Jimenez S., Juarez-Ramirez R. A Quality Framework for Evaluating Grammatical Structure of User Stories to Improve External Quality. In Proc. of the 7th International Conference in Software Engineering Research and Innovation (CONISOFT), 2019, pp. 147-153.

[20] Wautelet Y., Heng S. et al. Unifying and extending user story models. Lecture Notes in Computer Science, vol. 8484, 2014, pp. 211-225.

[21] Durán M., Juárez-Ramírez R. et al. User Story Estimation Based on the Complexity Decomposition Using Bayesian Networks. Programming and Computer Software, vol. 46, issue 8, 2020, pp. 569-583 / Дуран М., Хуарес-Рамирес Р. и др. Оценка пользовательских историй на основе декомпозиции сложности с использованием байесовских сетей. Труды ИСП РАН, том 33, вып. 2, 2021 г., стр. 77-92. DOI: 10.15514/ISPRAS–2021–33(2)–4.

[22] CHAOS Report 2015. The Standish Group International, 2015, 13 p.

[23] Fowler M., Highsmith J. The agile manifesto. Software Development Magazine, vol. 9, issue 8, 2001, pp. 29-30.

[24] Dalton J. Definition of Ready. In Great Big Agile: An OS for Agile Leaders, Apress, 2019, pp. 163-164.

[25] Subih M.A., Malik B.H. et al. Comparison of agile method and scrum method with software quality affecting factors. International Journal of Advanced Computer Science and Applications, vol. 10, issue 5, 2019, pp. 531-535.

[26] Mahnic V., Zabkar N. Measuring Progress of Scrum-based Software Projects. Elektronika ir Elektrotechnika, vol. 18, issue 8, 2012, pp. 73-76.

[27] Power K. Definition of ready: An experience report from teams at Cisco. Lecture Notes in Business Information Processing, vol. 179, 2014, pp. 312-319.

Information about authors / Информация об авторах

Claudia TONA, Master of Science, Professor. Research interests include Software Engineering, Agile Methodologies, Scrum, Agile Frameworks.

Клаудия ТОНА, профессор. Область научных интересов включает разработку программного обеспечения, Agile-методологии, Scrum, Agile Frameworks.

Samantha JIMÉNEZ, Doctor of Science, Full Professor. Research interests include Software Engineering, Usability, Educational Technology, Human-Computer Interaction.

Саманта ХИМЕНЕС, кандидат наук, профессор. Область научных интересов включает разработку программного обеспечения, удобство использования, образовательные технологии, взаимодействие человека и компьютера.

Reyes JUÁREZ-RAMÍREZ, Doctor of Computer Science, Full Professor. Research interests include software Engineering, software uncertainty estimation, and human-computer interaction.

Рейес ХУАРЕС-РАМИРЕС, кандидат компьютерных наук, профессор. Область научных интересов включает разработку программного обеспечения, оценку неопределенности программного обеспечения и взаимодействие человека и компьютера.

Rafael GONZÁLEZ PACHECO LÓPEZ, Researcher. Research interests: Land Use, Energy, Spatial Analysis, Governance, Complexity, Socio-Technical Analysis.

Рафаэль ГОНСАЛЕС ПАЧЕКО ЛОПЕС, исследователь. Научные интересы: землепользование, энергетика, пространственный анализ, управление, сложность, социально-технический анализ.

Ángeles QUEZADA, PhD in Computer Science. Research interests: Neural Networks, Pattern Recognition, Fuzzy Logic, Neural Networks and Artificial Intelligence, Computational Intelligence, Fuzzy Clustering, Computer Vision, Autism Spectrum Disorders, Autism.

Анхелес КЕСАДА, кандидат компьютерных наук. Научные интересы: нейронные сети, распознавание образов, нечеткая логика, нейронные сети и искусственный интеллект, вычислительный интеллект, нечеткая кластеризация, компьютерное зрение, расстройства аутистического спектра, аутизм.

César Arturo GUERRA GARCÍA, Doctor of Computer Science, Full Time Professor. Research interests include Software Engineering, Data and Information Quality, Requirements Engineering.

Сезар Артуро ГЕРРА ГАРСИА, кандидат компьютерных наук, профессор. Область научных интересов включает разработку программного обеспечения, качество данных и информации, разработку требований.

DOI: 10.15514/ISPRAS-2023-35(1)-7

Students' Systems Thinking Competencies Level Identification through Concept Maps Assessment

[1] *J.R. Aguilar-Cisneros, ORCID: 0000-0003-3040-157X <jorge.aguilar@upaep.mx>*
[2] *R. Valerdi, ORCID: 0000-0002-2746-0395 <rvalerdi@arizona.edu>*
[3] *B.P. Sullivan, ORCID: 0000-0002-4646-7277 <b.p.sullivan@utwente.nl>*

[1] *Universidad Popular Autónoma del Estado de Puebla (UPAEP),*
Puebla, Pue., México, 72410
[2] *University of Arizona,*
Tucson, AZ 85721, USA
[3] *University of Twente*
p.o. box 217, 7500 AE Enschede, The Netherlands

Abstract. Systems Thinking Competencies have become extremely important and widely studied due to increasing systems complexity. Because of this, when they are taught, it is extremely useful to identify whether or not students own Systems Thinking Competencies in order to design a specific teaching strategy. This research applied an Adapted Holistic Scoring Method to assess Concept Maps developed by postgraduate and undergraduate engineering students in order to identify Systems Thinking Competencies. It had two phases. At the first one, Students showed an acceptable knowledge of cost estimation drivers, and a certain level of Systems Thinking Competencies. In the second phase, both cost estimation drivers and Systems Thinking Competencies showed an improvement. Mann-Whitney U-test was applied in order to identify if there were significant differences between Phase 1 and Phase 2. Confidence level of 95%, and a significance level of 0.05 was considered.

Keywords: system thinking; student's competencies; Adapted Holistic Scoring Method; Concept Maps; U-test

For citation: Aguilar-Cisneros J.R., Valerdi R., Sullivan B.P. Students' Systems Thinking Competencies Level Identification through Concept Maps Assessment. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 101-112. DOI: 10.15514/ISPRAS-2023-35(1)-7

Определение уровня способности студентов к системному мышлению с помощью оценки концептуальных карт

[1] *Х.Р. Агилар-Сиснерос, ORCID: 0000-0003-3040-157X <jorge.aguilar@upaep.mx>*
[2] *Р. Валерди, ORCID: 0000-0002-2746-0395 <rvalerdi@arizona.edu>*
[3] *Б.П. Салливан, ORCID: 0000-0002-4646-7277 <b.p.sullivan@utwente.nl>*

[1] *Народный автономный университет штата Пуэбла,*
Мексика, 72410, Пуэбла
[2] *Аризонский университет,*
США, AZ 85721, Тусон
[3] *Университет Твенте*
Нидерланды, 7500 AE Энсхеде

Аннотация. Способность к системному мышлению стала чрезвычайно важной и широко изучаемой из-за возрастающей сложности систем. Из-за этого при обучении студентов очень полезно определить, обладают ли они способностями к системному мышлению, чтобы разработать конкретную стратегию

обучения. В нашем исследовании применялся адаптированный целостный метод для оценки концептуальных карт, построенных аспирантами и студентами технических специальностей с целью выявления способностей к системному мышлению. Исследование состояло из двух фаз. На первой фазе студенты показали приемлемое знание факторов оценки затрат и некоторый уровень способности к системному мышлению. На втором этапе оба эти показателя были улучшены. Применялся U-критерий Манна-Уитни, чтобы определить, есть ли существенные различия между фазой 1 и фазой 2. Учитывались уровень достоверности 95% и уровень значимости 0,05.

Ключевые слова: системное мышление; способности студентов; адаптированный целостный метод оценки; концептуальные карты; U-критерий

Для цитирования: Агилар-Сиснерос Х.Р., Валерди Р., Салливан Б.П. Определение уровня способности студентов к системному мышлению с помощью оценки концептуальных карт. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 101-112. DOI: 10.15514/ISPRAS–2023–35(1)–7

1. Introduction

Systems Thinking is a holistic approach to analyze, and solve complex problems and systems [1, 2]. It means looking for an acceptable solution among several potential solutions. These solutions will own relationships of correlation, some of them may involve cause and effect. Systems Thinking emphasizes the complexity of relationships, seeking out webs of causality rather than single, linear causes [3]. In this regard, this study focuses on, firstly, identifying how students change their mind, over time, about cost estimation drivers while they receive an academic course related to this topic. Secondly, how students represent their change of mind through a concept map, and finally how those Concept Maps shed light on Systems Thinking. Particularly, one Systems Thinking Competence defined by [4], was analyzed. The study was applied to undergraduate and postgraduate students. They were enrolled in a cost estimation course during the spring 2020, where, among several concepts, cost estimation drivers were studied. This paper took into account the COCOMO model [5].

After Students' Concept Maps (SCMs) were assessed in phase 2, the outcomes showed an increasing knowledge on estimation cost drivers regarding the first evaluation on phase 1. Additionally, Systems Thinking Competence #3 was identified in the same way at a higher level on phase 2 than phase 1.

2. Background

The term Systems Thinking is known as "the art and science of making reliable inferences about behavior by developing an increasingly deep understanding of an underlying structure" [6]. Systems Thinking has been applied in a wide number of areas [7-9]. Based on these cases, Systems Thinking has demonstrated an ability to manage the complexity of systems, technical and societal, by considering the future implications of decision making and their long-term consequences. Additionally, Systems Thinking can be applied to decision making, and it often involves understanding the complexity of the situation, to see causal relationships, identify dynamic relationships among variables, and so on. In order to identify relationships [10, 11], and complex relationships [12, 13], Conceptual Maps are often used for modeling.

2.1 STC and Skills

There is neither an agreement about competencies definition nor its interpretation. The interpretation ranges from a description of competency in terms of performance and competence in terms of skills acquired by training to a broad view that encompasses knowledge, understanding, skills, abilities and attitudes [14].

This paper uses the term competence instead of competency. Competence is defined as a set of skills acquired by training or teaching. Systems Thinking can be described as a dual ability to understand systems and analyze circumstances, questions, or problems from a systems perspective [15].

Systems can be divided into three aspects, function (utility), structure (form) and behavior (dynamics) [16]. When both terms, Competence and Systems Thinking, are put together. Systems Thinking Competence (STC) arises. STC can be defined as: "Aspect that implies skills, knowledge, attitudes and behavior applied to tasks or activities where Systems Thinking perspective is needed". [1] stated, Systems Thinking can help to develop higher-order thinking skills, such as critical thinking, in order to understand and address complex, interdisciplinary, real-world problems. In this sense, some critical thinking skills (competencies) were defined by [17].

This paper was focused on just one of the eight STC defined by [4]. The competence selected, among eight of them, was the number 3. Competence #3: Ability to see relationships, a system can be understood in the context of relationships.

The competence #3 (STC #3) was selected because this competence is most closely related to cost modeling and allows to isolate a single competence without worrying about the confounding or mediating effects of others.

2.2 Cost Estimation

There are several Effort Estimation Methods, among them, COSMIC [18], User Stories [19]. COCOMO model was used in this research, it estimates the amount of effort in person-months (PM). COCOMO's equations require effort multipliers (EM) and scale drivers/factors (SDF) as inputs. COCOMO defines five Scale Drivers Factors: PREC, FLEX, RESL, TEAM, and PMAT. Additionally, COCOMO defines several Effort Multipliers: RELY, DATA, CPLX, RUSE, DOCU, TIME, STOR, PVOL, ACAP, PCAP, PCON, APEX, LTEX, PLEX, TOOL, SITE, and SCED.

Effort Multipliers and Scale Drive Factors were searched out on each SxCMs (Student x's Concept Map) in order to identify how many of them were used to build their SxCM. Depending on this, a rate was assigned to each SxCM [20].

2.3 CMs and Scoring

Concept Maps were developed by [21] at Cornell University, in order to understand changes in students' knowledge, mainly because CMs are graphical tools for organizing and representing conceptual understanding [22]. Additionally, CMs and Systems Thinking can be used together because they share common characteristics such as structure, dynamism and hierarchy, and some researchers indicate that increase in the number of concepts, connections and diversity in CMs are a reliable parameter for measuring students' systematic thinking [23, 24]. CMs have been used in a wide spectrum of areas due to its advantages [25-29].

Additionally, [26] did an evaluation of six scoring methods: 1) holistic, 2) holistic with master map, 3) relational, 4) relational with master map, 5) structural, and 6) structural with master map. To calculate similarity between subjects' Concept Maps, and the Master Concept Map (MCM), [30] a set of theoretic methods described by [20] can be used.

In general terms, there is a big quantity of research where Concept Maps have been used in order to identify Systems Thinking skills or competencies [31-35].

3. Research Methodology

This section describes the research methodologies' main elements and instruments used to gather and analyze data.

3.1 Research General Background

In [36], authors analyzed whether particular features (medium) of Concept Maps affect the assessment of student's Systems Thinking. They found that the medium rarely influenced the validity of Concept Maps for Systems Thinking. Furthermore, the authors suggest Concept Maps as an appropriate assessment of Systems Thinking.

However, for this research neither specific tool to build Concept Maps was requested nor specific instructions to build them was given. Additionally, for this study an Adapted Holistic Scoring Method (AHSM) was used together with a Master Map Methodology.

The original holistic methodology was adapted because of the type of raw material gathered.

There were two assessments. The first one was applied in January (Phase 1) and the second one in February (Phase 2), both of them in 2020. The assessments consisted, basically, about identifying how many Cost Estimation Drivers terms the students were able to use, and how many Scale Factors and Effort Multipliers they were able to specify when they developed their Concept Maps, additionally the level of Systems Thinking Competence #3 embedded into their Concept Maps was tried to identify. The second assessment analyzed if students used specific cost estimation drivers. System Thinking Competence #3 embedded into their Concept Maps was evaluated. Particularly, we identified if students have reached a better level.

3.2 Research Problem

Solving complex problems is one of the main activities in some industries, where, in the future, students could be hired. In this sense, identifying Systems Thinking Competencies owned by students can be useful. With this identification, training activities to strengthen them could be planned and managed. This research identified one Systems Thinking Competence owned by a group of students (graduated and ungraduated) The Systems Thinking Competence identified was STC #3.

3.3 Research Focus

This research was focused on identifying a STC owned by students, particularly we were focused on how students change their ability to see relationships after they received theory about cost estimation drivers and how they were able to represent it through a Concept Map. Applying the Adapted Holistic Scoring Method together with the assessment rubric, Concept Maps were assessed.

3.4 Research Aim and Research Questions

Research questions guidelines this study:

RQ1: How does ability to see relationships (STC #3) change over time as a result of learning cost modeling?

RQ2: How do students' mental models of the factors that impact project costs change over time?

3.5 Participants

This research was applied to a sample of participants consisting of undergraduate and postgraduate students. All of them were enrolled at different Systems and Industrial Engineering Department careers. The first study was applied to 61 students and the second to 45 students. The survey wasn't mandatory in order to avoid any kind of bias. The study was applied in the spring 2020.

3.6 Instrument and Procedures

The methodology included two phases. The first one (see Figure. 1) assess the degree of similarity between Students' CMs (SxCM) and the Master Concept Map (MCM). High levels of similarity indicate SxCMs own a considerable quantity of Cost Estimation Drivers in their Concept Maps. In order to do this assessment, the Adapted Holistic Scoring Method [30] was used (See Figure 1). According to [30] in order to assess CMs similarity, these have to be constructed using the same set of concepts. However, when SxCMs was requested for this research, and when MCMs were developed, both of them used different sets of concepts because our students did not receive a common set of concepts. Our students did not receive specific information about how to build CMs. During the first assessment, students did not know Cost Estimation Drivers.

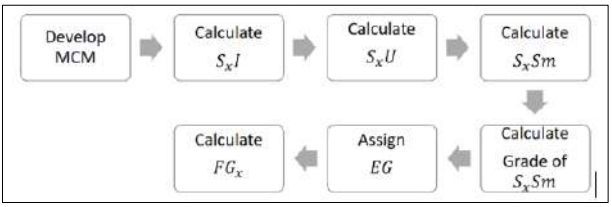

Fig. 1. Adapted Holistic Scoring Method (AHSM)

These equations were used:

a) $S_xI = MCM \cap S_xCM$;

b) $S_xU = \left(\sum_{i=1}^{n} MCM_{term_i} + \sum_{i=1}^{n} S_xCM_{term_i}\right) - S_xI$;

c) $S_xS_m = \frac{S_xI}{S_xU}$;

d) $G_x = S_xS_m * 10$;

e) $EG = ExpertGrage$;

f) $FG_x = (EG + G_x)/2$.

3.7 Assessment Rubric

The assessment was done through a rubric (see Table 1). This rubric was used to evaluate each SxCM, and level of System Thinking Competence #3 (STC#3) was identified. The elements of STM#3 identified indicate whether students saw relationships between the cost estimation drivers (direct costs) and the costs around the project (indirect costs). Regarding relationships, three levels were defined: Low, Medium and High. A SxCMs got low level, if 0 or 1 related element with STC #3 was identified, when 0 elements were identified, it means SxCMs only contains Cost Drivers Estimation related with the cost estimation project itself (direct costs). A SxCMs got medium level, if two related elements with STC #3 were identified. Finally, a SxCMs got high level, if three or more related elements with STC #3 were identified.

Table 1. Rubric to assess SxCMs vs STC #3

SxCM	**Assessment STC #3**			**Identified Level**		
	Low	**Medium**	**High**	**1**	**2**	**3**
	Low. If 1 or 0 external elements were identified	**Medium.** If 2 external elements were identified	**High.** If 3 or more external elements were identified			

The methodology to assess the STC #3 in each SxCMs is shown in figure 3. Figures 11 and 12 represent the outcomes.

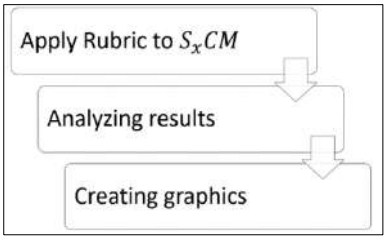

Fig. 2. Methodology to identify STC #3 into SxCMs

3.8 Statistical Test

The Mann-Whitney U-test was used to compare SxCMs developed at the first phase vs SxCM developed at the second phase

Two Mann-Whitney tests were applied, one of them to assess Cost Estimation Drivers. The second test to assess STC#3's level reached.

The first Mann-Whitney test was applied to Cost Estimation Drivers. The Null Hypothesis (H_0), and the alternative hypothesis (H_1) were defined:

$H_0 = NCED_1 >= NCED_2$;

$H_1 = NCED_1 < NCED_2$.

$NCED_1$ represents the number of Cost Estimation Drivers included by students in their CMs in the first phase study, and $NCED_2$ represents the number of Cost Estimation Drivers included by students in their CMs in the second phase study. A confidence level of 95%, and a significance level of 0.05 ($\alpha = 0.05$) were defined.

The size sample was 105, in the first phase study there were 60 students (one of them was eliminated) and in the second phase there were 45 students. It means, 105 Concept Maps were analyzed.

The second Mann-Whitney test was applied to Systems Thinking Competence Level #3 reached by students. The Null Hypothesis (H_0), and the alternative hypothesis (H_1) were defined:

$H_0 = STCL_1 <= STCL_2$;

$H_1 = STCL_1 > STCL_2$.

$STCL_1$ represents the level of Systems Thinking Competence #3 reached by students in their CMs in the first phase study, and $STCL_2$ represents the level of Systems Thinking Competence #3 reached by students in their CMs in the second phase study. A confidence level of 95%, and a significance level of 0.05 ($\alpha = 0.05$) were defined.

The size sample was 95. In the first phase study there were 54 students and in the second phase there were 41 students. Some students were eliminated because they didn't reach any level of STC#3.

Some parameters have to be computed, U and Z_u, according with the following equations:

$$U_1 = n_1 n_1 + \frac{n_1(n_1+1)}{2} - R_1; \quad (1)$$

$$U_2 = n_1 n_1 + \frac{n_2(n_2+1)}{2} - R_2; \quad (2)$$

$$Z_u = \frac{\left|U - \frac{n_1 n_1}{2}\right|}{\sqrt[2]{\frac{n_1 n_1 (n_1+n_2+1)}{12}}}. \quad (3)$$

4. Research Results

This section presents the outcomes regarding degree of similarity between Students' CMs (SxCM) and the Master CMs (MCM). Additionally, the results regarding the level of Systems Thinking Competence (STC #3) reached are shown.

4.1 AHSM and Rubric Results

The first assessment (phase one), maximum and minimum scores reached for SxCMs (see Table 2) were calculated. The SxCM's results showed low grades but it was due to, at this point of time, the

students had not received formal teaching about cost estimation drivers and they had a whole freedom to develop their own Concept Maps.

As it can be seen in Table 2, due to low standard deviation, most of the students got around 4.2 points and an average of 4.15

Table 2. Phase one outcomes (First assessment)

MCMvsS1CMtoS61CM				
Average grade	**Max. grade**	**Min. grade**	**Mean**	**Standard Deviation**
4.15	6.42	2.69	4.18	0.98

Approximately one month later second assessment was applied, and the same task was requested. There was an increase of almost two points, from 6.42 to 8.06 (See Table 3)). Additionally, more specific cost estimation factors were used by students.

Table 3. Phase two outcomes (Second assessment)

MCMvsS1CMtoS45CM				
Average grade	**Max. grade**	**Min. grade**	**Mean**	**Standard Deviation**
4.49	8.06	3.21	4.41	0.88

Additionally, an analysis about Scale Drive Factors and Effort Multipliers was applied. 64.4 % of students included, at least, one SDF on their SxCMs (See Figure 3).

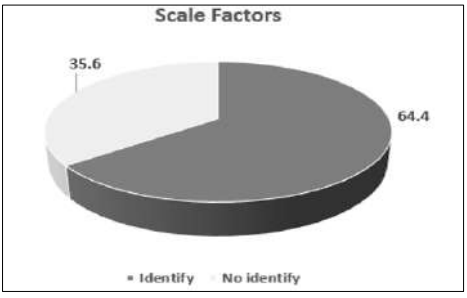

Fig. 3. Scale drive factors

Moreover, EM were identified on SxCMs 73.3% of students included it on their SxCMs. It means almost all students increased their knowledge about specific elements that influence a project cost.

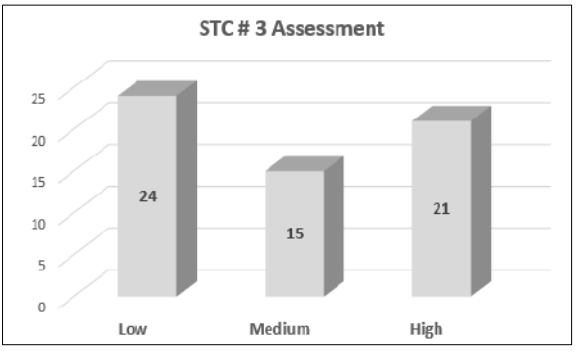

Fig. 4. Level of STC #3 identified. First assessment phase

Furthermore, the assessment rubric (See Table 1) was applied in order to identify aspects about Systems Thinking Competence #3 (STC#3). It was applied in the first and second assessment phase (See Figures 4, and 5).

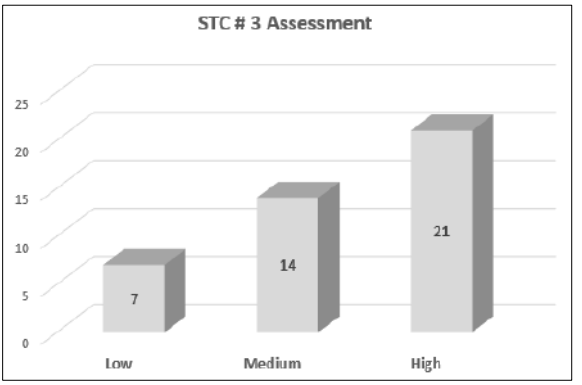

Fig. 5. Level of STC #3 identified. Second assessment phase

Furthermore, the assessment rubric (See Table 1) was applied in order to identify aspects about Systems Thinking Competence #3 (STC#3). It was applied in the first and second assessment phase (See Figures 4, and 5).

The S19CMs (student 19 of 61, first assessment phase), is shown in order to identify what kind of Student Concept Map was built (See Figure 6). This student got a final grade of 4.6 points in the first assessment phase. 4.6 means a similarity of 46% with Master Concept Map, in other words, the student 19 included 5 of 19 cost estimation drivers expected.

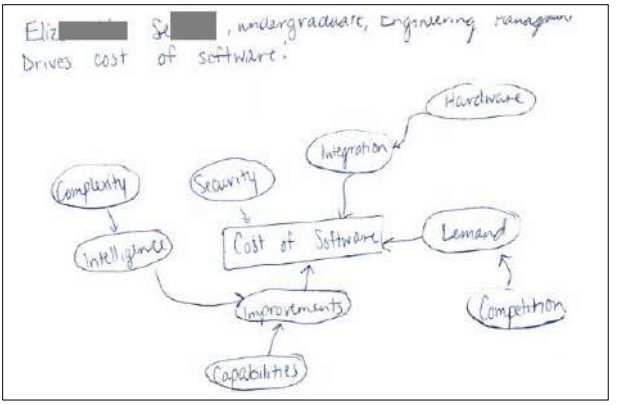

Fig. 6. S19CM (First assessment phase)

The same student (Student 35 = student 19 in phase 1), but in the second assessment phase (Phase 2), got a final grade of 8.1. This grade, 8.1, means a similarity of 81% with Master Concept Map. In other words, the student 35 included 13 of 20 cost estimation drivers. There was an improvement of 3.5 points.

4.2 Mann-Whitney Results

The first Mann-Whitney U-Test applied to Estimation Cost Drivers Included by students in their Concept Maps at first and second study gives us the next results.

The Null Hypothesis (H_0), and the alternative hypothesis (H_1) were defined to the first Mann-Whitney test, where significant Estimation Cost Drivers included at first phase regarding Estimation Cost Drivers included at second phase study was computed.

Phase 1 and Phase 2 were related to $NCED_1$ and $NCED_2$, respectively. The sample size for the first phase and the second phase was n_1 =54, and n_2 =41. The median to Phase 1, and Phase 2 was 3 and 5 respectively (See Figure 7).

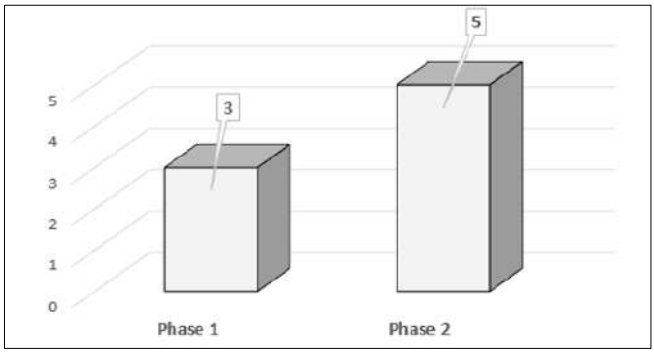

Fig. 7. Medians Phase 1 and Phase 2

Additionally, Figure 8 shows frequency histogram of Estimation Cost Drivers included in students' Concept Maps. This graph represents how many Cost Estimation Drivers were included in the Students' Concept Maps at the first phase, and at the second phase.

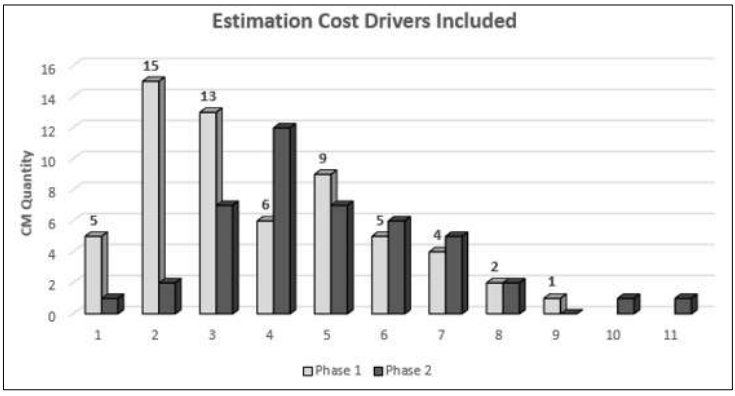

Fig. 8. ECD included at CMs

The rank of Phase 1 and Phase 2 was calculated: $Rank_1$ = 2641, and $Rand_2$ = 2924.

After these calculations, the *U* parameter was calculated for both Groups: U_1 = 1889, and U_2 = 811 (See equation (1), (2)). Hence U = 811.

Z_u was computed because our sample was larger than 20. After applying the equation (3), 3.49 was the value obtained to Z_u.

Since p-value>α (p-value=0.99, α=0.05), the null hypothesis H_0=$STCL_1$ <= $STCL_2$ cannot be rejected. The cost estimation drivers identified by students in Phase 1 are assumed to be less than or equal to the cost estimation drivers identified by students in Phase 2. Additionally, p(x<=Z=0.00095), it means that the chance of error rejecting H_0 is too high:0.999 (99.9%). However, when we estimate the common language effect size $(U/(n_1 * n_2) = 0.30$, this is the probability that a random cost estimation driver from Phase 1 is greater than a random cost estimation driver from Phase 2. Finally, H_0 cannot be rejected.

The second Mann-Whitney test was applied to Systems Thinking Competence Level #3 reached by students. Phase 1 and Phase 2 were related with $STCL_1$ and the second with $STCL_2$. The sample size for the first phase and the second was n_1 =54, and (n_2 =41). The median to Phase 1, and Phase 2 was 2 and 3 respectively.

The U parameter was calculated for both phases. U_1 = 1321.5, and U_2 = 892.5. Hence *U* = 892.5 because the U_2 with fewer scores is selected, in this case U_2 was selected.

Z_u was computed because our sample was larger than 20. After applying the equation, the value obtained to Z_u was 1.61.

Since p-value>α (p-value=0.97, α=0.05), the null hypothesis $H_0=STCL_1 <= STCL_2$ cannot be rejected. The Systems Thinking Competence #3 reached by students at phase 1 are assumed to be less than or equal to the Systems Thinking Competence #3 reached by students at phase 2.

5. Discussion

Regarding RQ1. There was just one month between phase one and phase two, despite those, a relevant and remarkable difference was detected. After students received a little training about cost estimation drivers, they were able to develop Concept Maps where more cost estimation drivers were included, and they were able to see more relationships, it means, they increased the level of Systems Thinking Competence #3 embedded into their Concept Maps.

Regarding RQ2. It is necessary to provide training or teaching where students get information about cost estimation drivers and this information can be internalized for each student in case they work alone and socialized in case they work in a team. During this time, students received information about what factors could impact project costs. The knowledge acquired allowed students to build Concept Maps with more cost estimation drivers included. These outcomes shown it is important include information about what Systems Thinking Competence is, and add information about the specific topic or area to be tackled.

A Mann-Whitney U-test was applied to Cost Estimation Drivers included in Concept Maps developed by students in both Phase 1 and Phase 2. The changes between these Phases is statistically significant. The null hypothesis $H_0=STCL_1 <= STCL_2$ cannot be rejected

An additional Mann-Whitney U-test was applied to Level of Systems Thinking Competence#3 reached by students when they develop their Concept Maps, in both Phase 1 and Phase 2, as a result, we can realize, the changes between Phase 1 and 2 were statistically significant. The null hypothesis $H_0=STCL_1 <= STCL_2$ cannot be rejected.

Limitations and threats. This research was applied to a limited sample of participants, which consisted of undergraduate and postgraduate students, where an optimal sample size wasn't calculated. Additionally, the sample is not heterogeneous, as a result, the outcomes cannot be generalized to different areas of engineering. Hence, the outcomes must be taken with caution, and the outcomes cannot be generalized.

6. Conclusions

This research has shed light on a specific Systems Thinking competence (STC #3). Particularly, it collected evidence about STC #3 owned by undergraduate and postgraduate students. Even when they did not know what a Systems Thinking Competence is.

Collecting this kind of information can be useful when Systems Thinking Competencies must be taught. In other words, before a teacher or trainer will teach Systems Thinking Competencies, it is recommended to apply an initial diagnostic test in order to identify the level of knowledge in each competence, after that, the results can be used in order to design a strategy to teach Systems Thinking Competencies. These actions could save training time, and reach desired objectives more quickly and more efficiently.

Outcomes indicated engineering students own some cost estimation knowledge. This can be understood because they own an engineer profile, and they are aware about aspects that have to be taken into account when a project is developed and when a cost has to be estimated. For instance, they identified aspects that imply time and money.

Outcomes obtained can be useful in order to design an educational strategy when the cost estimation topic is taught.

Additionally, this research has shown that despite engineering students not writing specific cost estimation's names, they identified them, hence just teaching specific cost estimation names will be required, and obviously, detailed theory about it.

Mann-Whitney U Test was useful in order to show that changes between phase 1 and phase 2 were statistically significant.

References / Список литературы

[1] York S., Lavi R. et al. Applications of Systems Thinking in STEM Education. Journal of Chemical Education, vol. 96, issue 12, 2019, pp. 2742-2751.

[2] Maracha V. Systems thinking and collective problem solving practices. In Proc. of the V International Research and Practice Conference–Biennale on System Analysis in Economics, 2018, pp. 269-272.

[3] Amissah M., Gannon T., and Monat J. What is Systems Thinking? Expert Perspectives from the WPI Systems Thinking Colloquium. Systems, vol. 8, issue 1, 2020, 26 p.

[4] Valerdi R., Rouse W.B. When Systems Thinking is Not a Natural Act. In Proc. of the IEEE International Systems Conference, 2010, pp. 184-189.

[5] Boehm B.W., Abts C. et al. Software Cost Estimation with COCOMO II. Prentice Hall Press, 2000, 544 p.

[6] Richmond B. Systems thinking/system dynamics: Let's just get on with it. System Dynamics Review, vol. 10, issues 2-3, 1994, pp. 135-157.

[7] Mehrjerdi Y.Z. Quality function Deployment and its profitability engagement: A Systems Thinking perspective, International Journal of Quality & Reliability Management, vol. 28, issue 9, 2011, pp. 910-928.

[8] Hebel M. Light Bulbs and Change: Systems Thinking and Organizational Learning of New Ventures. The Learning Organization, vol 14, issue 6, 2007, pp. 499-509.

[9] Sankaran S., Hou Tay B., Orr M. Managing Organizational Change by using Soft Systems Thinking in Action Research Projects. International Journal of Managing Projects in Business, vol. 2, issue 2, 2009, pp. 179-197.

[10] Miertschin S.L., Willis C.L. Using Concept Maps to navigate complex learning environments. Proceedings of the 8th Conference on Information Technology Education, 2007, pp. 175-184.

[11] Schwendimann B. Concept Mapping. In Encyclopedia of Science Education, Springer, 2015, pp. 198-202.

[12] Davies M. Concept Mapping, Mind Mapping and Argument Mapping: What are the Differences and Do They Matter? Higher Education, vol. 62, issue 3, 2011, pp. 279-301.

[13] Henderson C., Yerushalmix E. et al. Multi-Layered Concept Maps for the Analysis of Complex Interview Data. Roundtable Discussion, Presented at the AAPT Physics Education Research Conference, 2003, 13 p.

[14] Kennedy D., Hyland A., Ryan N. Learning Outcomes and competencies. Introducing Bologna objectives and tools. B 2.3-3, 2019.

[15] Crawley E., Cameron B., Selva D. Systems architecture: Strategy and product development for complex systems. Pearson, 2015, 480 p.

[16] Lavi R., Dori Y.J. Systems Thinking of pre- and in-service science and engineering teachers. International Journal of Science Education, vol. 41, issue 2, 2018, pp. 248-279.

[17] Richmond B. Systems Thinking: Critical Thinking Skills for the 1990s and beyond. System Dynamics Review, vol. 9, issue 2, 1993, pp. 113-133.

[18] Valdés-Souto F., Naranjo-Albarrán L. Improving the Software Estimation Models Based on Functional Size through Validation of the Assumptions behind the Linear Regression and the Use of the Confidence Intervals When the Reference Database Presents a Wedge-Shape Form. Programming and Computer Software, vol. 47, issue 8, 2021, pp. 673-693.

[19] Durán M., Juárez-Ramírez R. et al. User Story Estimation Based on the Complexity Decomposition Using Bayesian Networks. Programming and Computer Software, vol. 46, issue 8, 2020, pp. 569–583 / Дуран М., Хуарес-Рамирес Р. и др. Оценка пользовательских историй на основе декомпозиции сложности с использованием байесовских сетей. Труды ИСП РАН, том 33, вып. 2, 2021 г., стр. 77-92. DOI: 10.15514/ISPRAS–2021–33(2)–4.

[20] Trendowicz A., Jeffery R. Constructive Cost Model – COCOMO. In Software Project Effort Estimation, Springer, 2014, pp. 277-293.

[21] Novak J.D., Musonda D. A Twelve-Year Longitudinal Study of Science Concept Learning. American Educational Research Journal, vol. 28, issue 1, 1991, pp. 117-153.

[22] Tripto J., Assaraf O.B.-Z., Amit M. Mapping What They Know: Concept Maps as an Effective Tool for Assessing Students' Systems Thinking. American Journal of Operations Research, vol. 3, issue 1A, 2013, pp. 245-258.

[23] White R., Gunstone R. Probing Understanding, Routledge, 1992, 208 p.
[24] Songer C.J., Mintzes J.J. Understanding Cellular Respiration: An Analysis of Conceptual Change in College Biology. Journal of Research in Science Teaching, vol. 31, issue 6, 1994, pp. 621-637.
[25] Turns J., Atman C.J., Adams R. Concept Maps for engineering education: A cognitively motivated tool supporting varied assessment functions. IEEE Transactions on Education, vol. 43, issue 2, 2000, pp. 164-173.
[26] Novak J., Gowin D. Learning How to Learn. New York: Cambridge University Press, 1984, 216 p.
[27] Stewart M. Joined up thinking? Evaluating the use of concept-mapping to develop complex system learning. Assessment & Evaluation in Higher Education, vol. 37, issue 3, 2012, pp 349–368.
[28] Rye J.A., Rubba P.A. Scoring Concept Maps: An Expert Map-Based Scheme Weighted for Relationships. School Science and Mathematics, vol. 102, issue 1, 2002, pp. 33-44.
[29] 12 D. L Darmofal, D. H. Soderholm, and D. R. Brodeur. Using Concept Maps and Concept Questions to Enhance Conceptual Understanding. Frontiers in Education Conference, Boston, MA, November 6–9, 2002.
[30] McClure J.R., Sonak B., Suen H.K. Concept Map Assessment of Classroom Learning: Reliability, Validity, and Logistical Practicality. Journal of Research in Science Teaching, vol. 36, issue 4, 1999, pp. 475-492.
[31] Trochim W.M., Cabrera D.A. et al. Practical Challenges of Systems Thinking and Modeling in Public Health. American Journal of Public Health, vol. 96, issue 3, 2006, pp. 538-546.
[32] Raved L., Yarden A. Developing seventh grade students' Systems Thinking skills in the context of the human circulatory system. Frontiers in Public Health, vol. 2, 2014, article no. 260, 11 p.
[33] Odom A.L., Kelly P.V. Integrating Concept Mapping and the Learning Cycle to Teach Diffusion and Osmosis Concepts to High School Biology Students. Science Education, vol. 85, issue 6, 2001, pp. 615-635.
[34] Markow P.G., Lonning R.A. Usefulness of Concept Maps in College Chemistry Laboratories: Students' Perceptions and Effects on Achievement. Journal of Research in Science Teaching, vol. 35, issue 9, 1998, pp. 1015-1029.
[35] Hu M., Shealy T. Methods for Measuring Systems Thinking: Differences Between Student Self-Assessment, Concept Maps Scores, and Cortical Activation During Tasks About Sustainability. In Proc. of the ASEE Annual Conference & Exposition. 2018, article id. 22718, 12 p.
[36] Brandstädter K., Harms U., GroBschedl J. Assessing System Thinking Through Different Concept-Mapping Practices. International Journal of Science Education, vol. 34, issue 14, 2012, pp. 2147-2170.

Information about authors / Информация об авторах

Jorge Rafael AGUILAR CISNEROS, Ph.D., Research Professor at UPAEP in the Department of Engineering. Research interests include Software Engineering and Software Processes.

Хорхе Рафаэль АГИЛАР СИСНЕРОС, кандидат наук, профессор-исследователь инженерного факультета. Область научных интересов включает программную инженерию и программные процессы.

Ricardo VALERDI, Ph.D., Distinguished Outreach Professor and Interim Department Head, Department of Systems & Industrial Engineering. His research focuses on Cost estimation, systems thinking, sports analytics.

Рикардо ВАЛЕРДИ, кандидат наук, профессор кафедры систем и промышленной инженерии. Его исследования сосредоточены на оценке затрат, системном мышлении, спортивной аналитике.

Brendan Patrick SULLIVAN, Ph.D., Assistant Professor at the University of Twente in the Department of Design, Production and Management (DPM). Research interests include intelligent control systems, decision analysis and changeable engineering systems.

Брендан Патрик САЛЛИВАН, кандидат наук, ассистент кафедры дизайна, производства и управления. Область научных интересов включает интеллектуальные системы управления, анализ решений и изменяемые инженерные системы.

DOI: 10.15514/ISPRAS-2023-35(1)-8

Blockchain and Satisfiability Modulo Theories for Tender Systems

R. Dávila, ORCID: 0000-0002-3174-5748 <photographic_ren@comunidad.unam.mx>
R. Aldeco-Pérez, ORCID: 0000-0002-7003-2724 <raldeco@unam.mx>
E. Bárcenas, ORCID: 0000-0002-1523-1579 <ebarcenas@unam.mx>
Universidad Nacional Autónoma de México
Ciudad Universitaria, Coyoacán, 04510 Mexico City, Mexico

Abstract. A tender process consists in competing offers from different candidate suppliers or contractors. The tender winner is supposed to supply or provide a service in better conditions than competitors. Tenders are developed using centralized unverified systems, which reduce transparency, fairness and trust on the process, it also reduces the ability to detect malicious attempts to manipulate the process. Systems that provide formal verification, decentralization, authentication, trust and transparency can mitigate these risks. Satisfiability Modulo Theories provides a formal analysis to prove correctness of tender offers properties, verified properties ensures system reliability. In addition, one technology that claims to provide decentralization is Blockchain, a chain of distributed and decentralized records linked in a way such that integrity is ensured. This paper presents a formal verified and decentralized proposal system, based on Satisfiability Modulo Theories and Blockchain technology, to make electronic procurement tenders more reliable, transparent and fair.

Keywords: Satisfiability Modulo Theories; Tender verification; Blockchain; e-Procurement

For citation: Dávila R., Aldeco-Pérez R., Bárcenas E. Blockchain and Satisfiability Modulo Theories for Tender Systems. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 113-122. DOI: 10.15514/ISPRAS-2023-35(1)-8

Acknowledgments. This research was supported by the Mexican Council CONACYT (1006953) in collaboration with Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas: Posgrado de Ciencia e Ingeniería de la Computación of the Universidad Nacional Autónoma de México. The work was also supported by UNAM-PAPIIT(IA104122) and UNAM-PAPIIT(TA101021).

Блокчейн и задача выполнимости формул в теориях для тендерных систем

Р. Давила, ORCID: 0000-0002-3174-5748 <photographic_ren@comunidad.unam.mx>
Р. Альдеко-Перес, ORCID: 0000-0002-7003-2724 <raldeco@unam.mx>
Э. Барсенас, ORCID: 0000-0002-1523-1579 <ebarcenas@unam.mx>

Национальный автономный университет Мексики
Мексика, 04510 Мехико, Койоакан, Университетский городок

Аннотация. В тендерном процессе участвуют конкурирующие предложения от разных кандидатов – поставщиков или их контрагентов. Победитель тендера должен поставить или оказать услугу на лучших условиях, чем конкуренты. Тендеры разрабатываются с использованием централизованных непроверенных систем, что снижает прозрачность, справедливость и доверие к процессу, а также снижает возможность обнаружения злонамеренных попыток манипулирования процессом. Системы, которые обеспечивают формальную проверку, децентрализацию, аутентификацию, доверие и прозрачность, могут снизить эти риски. Задача выполнимости формул в теориях обеспечивает формальный анализ для доказательства правильности свойств тендерных предложений, проверенные

свойства обеспечивают надежность системы. Кроме того, одной из технологий, обеспечивающих децентрализацию, является блокчейн, цепочка распределенных и децентрализованных записей, связанных таким образом, что обеспечивается целостность. В нашей статье представлена формальная проверенная и децентрализованная система управления тендерными предложениями, основанная на задаче выполнимости формул в теориях и технологии блокчейн и направленная на то, чтобы сделать электронные тендеры на закупки более надежными, прозрачными и справедливыми.

Ключевые слова: задача выполнимости формул в теориях; проверка тендеров; блокчейн; электронные закупки

Для цитирования: Давила Р., Альдеко-Перес Р., Барсенас Э. Блокчейн и задача выполнимости формул в теориях для тендерных систем. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 113-122. DOI: 10.15514/ISPRAS–2023–35(1)–8

Благодарности. Исследование поддерживалось Национальным советом по науке и технологии CONACYT (1006953) при сотрудничестве с Научно-исследовательским институтом прикладной математики и систем. Работа также поддерживалась грантами IA104122 и TA101021 Программы поддержки научно-исследовательских и технологических инновационных проектов (UNAM-PAPIIT).

1. Introduction

Public tenders are sensitive to fraud and corruption; therefore, the laws of most countries regulate government procurement. One example is the European scheme for public tenders, which is one of the most organized and documented [1]. In this scheme, contracts typically go through competitive processes, following common and local legal guidelines of each member country of the European Union. The purpose of this scheme is to offer a fair process for the participants, including a fair price for the taxpayers of the country issuing the tender. Currently, this scheme handles various types of procedures for tendering, such as open or restricted. These procedures have in common a negotiation about what the participants will supply, but with different rules between each type of procedure.

Although governments have robust legal rules for bidding procedures, these procedures are carried out centrally, where a collective or an individual entity reviews each bid based on the rules established by the corresponding tender. So later, the supplier with the proposal offering the best cost/quality ratio is selected.

This centralization creates different risks for the tendering procedures. Centralized entities might give preferential treatment to some of the participants, thereby, undermining the fairness of the process. There is also the possibility that bids are manipulated to favor a specific participant. In addition, the transparency of the procedures can be compromised, as the results of the tendering process presented to the public are not reliable [2] as malicious manipulations are not published.

This problem has been already identified by some governments that have proposed initiatives for electronic tendering schemes. Some of these schemes are implemented using information systems that carry out bidding procedures through the Internet. One example of these kind of government is presented in [3], where a large-scale implementation was developed.

Despite the advantages offered by these systems, they are still centralized, therefore, managed by selected entities who must comply with the applicable rules. Centralization might hide malicious manipulation.

In addition, it is also not possible to automatically verify if the tender rules are meet by the participants. By doing this, human errors and data manipulation can be reduced. Therefore, systems that provide automated verification, decentralization and transparency can mitigate these risks.

Therefore, by modelling and implementing a system based on Satisfiability Modulo Theories and Permissioned Blockchain, to validate, automate, offer immutable transparency, and ensure fairness in tendering procedures is possible.

Satisfiability Modulo Theories (SMT) [4] is a verification technique to prove correctness of system's properties. Properties are expressed in a formal language and when all given properties are satisfied,

it is said that the system is valid. This technique can be used to implement automatic verification of rules on a system.

Permissioned Blockchain [2] is a type of restricted Blockchain, where access to participants is controlled by having full identification of them. These participants are impartial entities that attest to the records that are generated in the Blockchain. This type of blockchain by having access control, greatly reduces the energy consumption required by public blockchains. In the latter, anonymity requires high resources in terms of hardware and energy, so a permissioned blockchain is more convenient for governments or public institutions.

Considering this, we propose a tendering system based on Satisfiability Modulo Theories and Permissioned Blockchain that supports biding processes.

By using this system, participants' bids will be automatically validated to later be registered in a blockchain, that through consensus of several peers supports decentralization. Therefore, reducing the reliance on a single entity. As consequence, reliability, fairness, integrity, and transparency of a tendering process can be guaranteed.

This paper presents the following contributions:

- Presents a system design for the public tendering procedure, as a reference for investigations of a similar nature;
- Shows the operation of the system model with facilities for its optimization and improvement;
- Validates inputs to the system through the use of a formal verifier;
- Securely and robustly registers the operations carried out in the tender process in a permissioned Blockchain system;
- Offers a proof of concept to set a precedent that implementation is possible.

1.1 Related work

First, work related to verification is shown, from which reference was taken to support the formality of our proposal.

Y. Limón et.al. present a “Mu-Calculus Satisfiability with Arithmetic Constraints” [5]. They study an extension of modal mu logic and Presburger arithmetic constraints, over tree models. They describe a satisfiability algorithm similar to our model.

D. Medina-Martínez et.al. present a “Database Management System Verification with Separation Logics” [6]. They propose to use Separation Logics to verify a database management system, focused on the verification of libraries containing heap data structure manipulation. Inside of the verification they use classical First Order Logic (FOL) reasoners to strength the verification process, in a similar way to our proposal.

In the following sections, we present results in the design, formalization and modeling of such a proposal.

2. Background

In this section, the concepts of Satisfiability Modulo Theories, Blockchain and Smart Contracts are presented.

2.1 Satisfiability Modulo Theories

The Satisfiability Modulo Theories (SMT) problem is a decision problem for logical formulas with respect to combinations of background theories expressed in classical First Order Logic with equality [4].

A decision problem is a problem that can be abstracted as a yes or no question of the input values, while a formal theory is a set of sentences that can be used to restrict the models we wish to consider.

An approach to solve SMT formulae is based on the observation that an SMT can be reduced to a Propositional Satisfiability Problem (SAT) formulae. Reductions can be solved atomically, to finally combine the results, to prove if the input formula is valid. This approach will be useful to validate the inputs during the operation of our proposed model protocol.

2.2 Blockchain

At the end of 2008 [7], along with the invention of cryptocurrencies, decentralized and transparent databases became popular. This is now known as Blockchain. According to NIST "Blockchains are distributed ledgers of cryptographically signed transactions that are grouped into blocks. Each block is cryptographically linked to the previous one (making it tamper evident) after validation and undergoing a consensus decision. As new blocks are added, older blocks become more difficult to modify (creating tamper resistance and strength integrity). New blocks are replicated across copies of the ledger within the network, and any conflicts are solved automatically using established rules [8]".

There exist two types of Blockchain: Permissionless and Permissioned. The former, called Public or Permissionless, it is open to all participants preserving their anonymity and offering full ledger transparency. Everyone in the network can validate transactions and can partake in the process of consensus. However, this type has a high energy consumption and uses consensus algorithms that take considerable time to reach an outcome [2]. An example of an application of this type of Blockchain is the Ethereum platform [9].

The latter, called Private or Permissioned, it is not open to all nodes. The participation of nodes is managed by third parties, usually impartial entities, i.e., they do not belong to the same organization and do not share interests. In this type of Blockchain, not all the nodes in the network can participate in the verification of the transactions. Instead, a selected group of nodes perform such verification, therefore, improving its efficiency. At difference of public blockchains, private blockchains do not provide decentralized security due to restricted access [2]. However, since in a private blockchain a third party assigns the access rights to each participant, the privacy level is increased making this type of blockchain suitable for government sectors. Moreover, their energy consumption is lower as consequence of the used consensus algorithms [2]. An example of this type of Blockchain is the Hyperledger Fabric platform [10].

Blockchain types use consensus protocols. A consensus protocol provides a technique for users or machines to coordinate in a distributed and decentralized setting. It ensures all participants agree on a unified transaction ledger without the help of a central authority. In the case of public blockchains, the consensus is achieved by the validations of the participants in the network, and in the case of private blockchains, the consensus is achieved by the selected entities accepted in the network [9].

3. System model

In this section, we present the high-level design of our system including the actors and its functionality. This functionality is later described through a set of sequence diagrams, as well as the description of the operation of the system's Blockchain network.

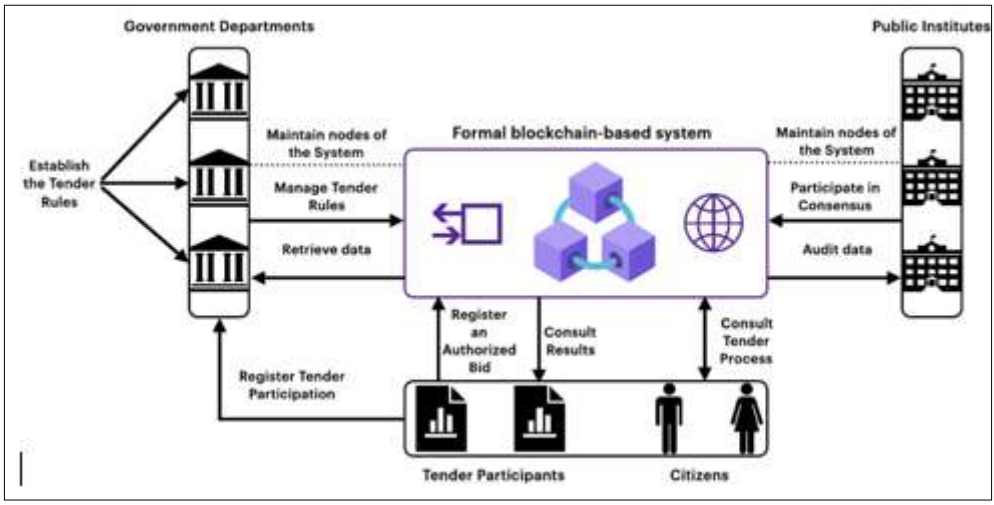

Fig. 1. Architecture of the model

3.1 System overview

The Fig.1 illustrates the architecture of the model. In this, there are 5 blocks that groups the main actors of the system. These blocks are defined as follows.

Formal blockchain-based system. This is the main block of the model that is depicted in purple on Fig.1. Here, the tender rules are established, operations in the Blockchain are registered and the winning offer is determined, all in an automated way. It also offers access to the information registered in the Blockchain to participants and citizens interested in reviewing the procedures carried out within the tender system.

Government Departments. This block represents the public government institutions that issue the calls for bids, establish the tender rules, control access to participants and are constantly managing the operations that occur in the Blockchain.

Public Institutes. This block represents the public institutions that participate in the bidding process as members of the consensus for registering transactions in the Blockchain. They audit the information that is recorded in the system, and also handle the operations that occur within the Blockchain. The participation of these institutions is considered impartial, to strengthen the fairness of the tender process.

Tender participants. This part of the block represents the companies or organizations interested in participating in the tender process. They are obliged to register their participation so that they have control over their access. Once registered, they can send their offers to the system, consult the results of the valid rules that they comply with, or consult the transactions with information on the procedures that were carried out in the tender process in a transparent manner.

Citizens. This part of the block represents citizens interested in reviewing a tender process, to check the procedure was fair and that the use of their taxes will be made according to the legislation.

Once the blocks that represent the actors in the model have been described, the proposed functionality of the system is presented below.

4. Formal model analysis

In this section, we present the results of the formalization of the tender rules and the offers of the participants, that occur in the Formal blockchain-based system (Fig. 1 in Section III).

Following tender rules specified in [1], we have identified the next four types of general rules in a tender process.

Specifications associated to a particular tender entity (several tender entities may form part of the tender), such as antitrust regulations or import or export taxes; Specifications associated to bidders, such as your legal identification or certifications; General specifications, such as a tender registration; and Numerical constraints, such as the price limit of the tender or budget of some offer proposal.

To explain how these rules are used in the tender we present the following set of definitions.

To formalize the tender rules, we propose a hybrid specification based on a rule-based expert system which are non-numerical specifications [11] and a numerical constraint system [12]. The rule-based expert system formalizes the knowledge required to express the type of rules not involving numerical constraints, that is, specifications associated to tender entities and bidders, and general specifications. Numerical constraints are formalized by the corresponding system.

Definition 1 (Non-numerical specifications). Non-numerical specifications are expressed by a set of rules of the following form:

$$\text{IF } antecedent \text{ THEN } consequent$$

where antecedent and consequent may represent a Boolean combination of statements.

Definition 2 (Numerical specifications). Numerical constraints are expressed by an equation system:

$$\begin{aligned} &a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n \leq t_1 \\ &a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n = t_2 \\ &\vdots \\ &a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n \geq t_m \end{aligned}$$

for any positive integers n and m. Notice other relations, such as >, <, ≤, ≥, may also be expressed, for instance $x \leq k$ holds if and only if $x + y = k$ for some positive integer y.

Now, we are going to define a bidder.

Definition 3 (Bidder Offer). A bidder offer is defined by the tuple $(Statements, NumericalEqualities)$, where $Statements$ is a set of fulfilled properties, defined by the tender rules, and $NumericalEqualities$ is a set of equalities between variables and positive real numbers, associated to costs.

We are now ready to define when a bidder satisfies the tender rules.

Definition 4 (Bidder offer fulfillment). Given a set of tender rules, expressed in terms of a rule-based expert system (Definition 1) and a numerical constraints system (Definition 2), we say a bidder offer fulfills the rules, if and only if, the statements and numerical equalities (Definition 3) fulfill all numerical and non-numerical specifications.

Definition 5 (Tender Rules Formalization). Given a set of tender rules, expressed in terms of a rule-based expert system (Definition 1) and a numerical constraints system (Definition 2), and a bidder b, we define the FOL formula $TR(b)$ (b occurs in TR) as follows:

$$TR(b) \coloneqq ES(b) \wedge NS$$

where n rules of the expert system are defined by the formula

$$ES(b) \coloneqq \bigwedge_{i=1}^{n} (Antecedent(b)_i \rightarrow Consequent(b)_i)$$

and m numerical constraints are defined by the formula

$$NS \coloneqq \bigwedge_{j=1}^{m} \sum_{k=1}^{l} a_{j,k}x_l = c_j$$

where c is a value given by the bidder b. Other relations, such as >, <, ≤, ≥, may also be expressed.

Definition 6 (Bidder Offer Formalization). Given a bidder b, expressed in terms of statements and numerical equalities (Definition 3), and his offer, we define the FOL formula $BO(b)$ (b occurs in BO) as follows:

$$BO(b) \coloneqq ST(b) \wedge NE$$

where n statements of the offer are defined by the formula

$$ST(b) \coloneqq \bigwedge_{i=1}^{n} (Statements(b)_i)$$

and m numerical equalities of the offer are defined by the formula

$$NE \coloneqq \bigwedge_{j=1}^{m} (a_j x_j = c_j)$$

where c is a value given by the bidder b.

Based on these definitions, the following theorem is constructed and proved.

Theorem 1 (Bidder offer verification). Given a set of tender rules and a bidder offer b, the FOL formula $TR(b) \wedge BO(b)$ is satisfiable if and only if the bidder offer fulfills the tender rules.

Proof: $[\![TR(b) \wedge BO(b)]\!]_V^S = 1 \Longrightarrow b$ fulfills tender rules.

Induction over the size of $TR(b) \wedge BO(b)$.

Base case:

There is only one rule for $BO(b)$ then there is only one $TR(b)$ rule to be satisfied,

$$ES(b) \wedge NS \wedge ST(b) \wedge NE$$

where

$$ES \coloneqq (Antecedent(b)_1 \rightarrow Consequent(b)_1)$$
$$NS \coloneqq a_{1,1} x_1 = c_1$$
$$ST \coloneqq Statement(b)_1$$
$$NE \coloneqq a_1 x_1 = c_1$$

Assume $BO(b)$ rule satisfies $TR(b)$ rule.

Therefore $\big(TR(b) \wedge BO(b)\big) = 1$ and by Definition 4 in this Section then b fulfills the tender rules.

Induction hypothesis: if there are n rules for $BO(b)$ then there are n $TR(b)$ rules to be satisfied.

Inductive step: proof for $n + 1$ rules for $BO(b)$ over $n + 1$ $TR(b)$ rules.

Case 1:

There is one $ES(b)$ rule and $n + 1$ NS rules

where

$$ES \coloneqq (Antecedent(b)_1 \rightarrow Consequent(b)_1)$$
$$NS \coloneqq \bigwedge_{i=1}^{n+1} \sum_{j=1}^{m+1} a_{i,j} x_m$$
$$ST \coloneqq Statement(b)_1$$
$$NE \coloneqq \bigwedge_{i=1}^{n+1} a_i x_i = c_i$$

Assume $BO(b)$ rules satisfy $TR(b)$ rules.

Therefore $\big(TR(b) \wedge BO(b)\big) = 1$ and by Definition 4 in this Section then b fulfills the tender rules.

Case 2:

There are $n + 1$ $ES(b)$ rules and one NS rule

where

$$ES(b) \coloneqq \bigwedge_{i=1}^{n+1} (Antecedent(b)_i \rightarrow Consequent(b)_i)$$
$$NE \coloneqq a_{1,1}x_1 = c_1$$
$$ST(b) \coloneqq \bigwedge_{i=1}^{n+1} (Statements(b)_i)$$
$$NE \coloneqq a_1 x_1 = c_1$$

Assume $BO(b)$ rules satisfy $TR(b)$ rules.

Therefore $\big(TR(b) \wedge BO(b)\big) = 1$ and by Definition 4 in this Section then b fulfills the tender rules.

Case 3:

There are $n+1$ $ES(b)$ rules and $n+1$ NS rules

where

$$ES(b) \coloneqq \bigwedge_{i=1}^{n+1} (Antecedent(b)_i \rightarrow Consequent(b)_i)$$
$$NS \coloneqq \bigwedge_{i=1}^{n+1} \sum_{j=1}^{m+1} a_{i,j} x_m$$
$$ST(b) \coloneqq \bigwedge_{i=1}^{n+1} (Statements(b)_i)$$
$$NE \coloneqq \bigwedge_{i=1}^{n+1} a_i x_i = c_i$$

Assume $BO(b)$ rules satisfy $TR(b)$ rules.

Therefore $\big(TR(b) \wedge BO(b)\big) = 1$ and by Definition 4 in this Section then b fulfills the tender rules.

The other implication direction is proved in an analogous manner. ■

The demonstration presented gives us the certainty that the rules of a tender process could be formalized correctly. This increases confidence for the participants in the tender, for the governments and for the citizens.

With the tender rules and participant bids formalized, we give the proposal model more confidence, and allows us to understand what the verifier block does precisely.

5. Discussion, Future work & Conclusions

In conclusion, we presented a formal model for verification to provide a more robust solution to a complex problem such as a tender process. Using that model, we created a system that along with a Blockchain network can offer greater confidence in a tender process.

To reach that goal, we also define logical formulas that are the basis for the formalization of offers in a bidding process. Later, we demonstrate the correct operation of the logical formulas, and thus have the confidence that the verification works correctly.

As future work, our prototype can be implemented with an attractive and user-friendly interface (system view) for potential final users. To have a fully automated system, the inputs for the automatic solver can be formatted on a logic-based notation. For that purpose, a procedure with this purpose should be constructed and integrated to our system.

Finally, this is an extension of [13] and short varsion of [14].

References / Список литературы

[1] Public tendering rules in the EU. Available at: https://europa.eu/youreurope/business/selling-in-eu/public-contracts/public-tendering-rules/index_en.htm.

[2] Huynh T.T., Nguyen T.D., Tan H. A survey on security and privacy issues of blockchain technology. In Proc. of the International Conference on System Science and Engineering (ICSSE), 2019, pp. 362-367,.

[3] Road data exchange layer. Available at: https://x-road.global/.

[4] Barrett C., Sebastiani R. et al. Satisfiability Modulo Theories. In Handbook of Satisfiability. IOS Press, 2009, pp. 825-885.

[5] Limón Y., Bárcenas E., et al. Mu-calculus satisfiability with arithmetic constraints. Programming and Computer Soft-ware, vol. 46, no. 8, 2020, pp. 503-510 / Лимон Й., Барсенас Э. и др.. Выполнимость мю-исчисления с арифметическими ограничениями. Труды ИСП РАН, том 33, вып. 2, 2021 г., стр. 191-200. DOI: 10.15514/ISPRAS–2021–33(2)–12.

[6] Medina-Martínez D., Bárcenas E. et al. Database management system verification with separation logics. Programming and Computer Software, vol. 47, no. 8, 2021, pp. 654–672.

[7] Nakamoto S. Bitcoin - A Peer-to-Peer Electronic Cash System. Appendix A. The bitcoin whitepaper by Satoshi Nakamoto. In Antonopoulos A. Mastering Bitcoin: Programming the Open Blockchain, 2nd ed. O'Reilly Media, 2017, pp. 323-334

[8] Yaga D., Mell P. et al. Blockchain technology overview. arXiv:1906.11078, 2019, 68 p,

[9] Ismail L., Materwala H. A review of blockchain archi-tecture and consensus protocols: Use cases, challenges, and solutions, Symmetry, vol. 11, issue 10, 2019, article no. 1198, 44 p.

[10] Hyperledger fabric. Available at: https://hyperledger-fabric.readthedocs.io/en/latest/index.html.

[11] Grosan C., Abraham A. Rule-Based Expert Systems. Intelligent Systems Reference Library, vol. 17, 2011, pp. 149–185.

[12] Bockmayr A., Weispfenning V., Maher M. Solving numerical constraints. In Handbook of Automated Reasoning. Robinson A., Voronkov A. eds. MIT Press, 2001, pp. 751–842.

[13] Dávila R., Aldeco-Pérez R., Bárcenas E. Tender system verification with satisfiability modulo theories.Iin Proc. of the 9th International Conference in Software Engineering Research and Innovation (CONISOFT), 2021, pp. 69–78.

[14] Dávila R., Aldeco-Pérez R., Bárcenas E. Formal Verification of Blockchain Based Tender Systems. Programming and Computer Software, vol. 48, issue 8, 2022, pp. 566-582.

Information about authors / Информация об авторах

René DÁVILA, Master degree in Computer Science and Engineering, PhD Student at Research Institute in Applied Mathematics and Systems (UNAM). Research interests: Decentralised and Distributed Authentication Protocols, Applications of Blockchain to improve services and Privacy of information, Security, Computer Logic, Distributed Systems, Algorithms Analysis.

Рене ДАВИЛА, магистр компьютерных наук и инженерии, аспирант Научно-исследовательского института прикладной математики и систем (UNAM), Научные интересы: протоколы децентрализованной и распределенной аутентификации, применение блокчейна для улучшения услуг и конфиденциальности информации, безопасность, компьютерная логика, распределенные системы, анализ алгоритмов

Rocío ALDECO-PÉREZ, Doctor on Computer Science, Research Associate. Research interests: Privacy of information, Decentralised and Distributed Authentication Protocols, Applications of Blockchain.

Росио АЛЬДЕКО-ПЕРЕС – кандидат компьютерных наук, доцент. Область научных интересов: конфиденциальность информации, децентрализованные и распределенные протоколы аутентификации, применение блокчейна.

Everardo BARCENAS, Ph.D., Assistant Professor. Research interests: modal logics, proof theory, automated reasoning, description logics, model checking, knowledge representation, planning, computer vision.

Эверардо БАРСЕНАС, кандидат наук, доцент. Научные интересы: модальная логика, теория доказательств, автоматические рассуждения, логика описания, проверка моделей, представление знаний, планирование, компьютерное зрение.

DOI: 10.15514/ISPRAS-2023-35(1)-9

Software project estimation using smooth curve methods and variable selection and regularization methods using a wedge-shape form database

F. Valdés-Souto, ORCID: 0000-0001-6736-0666 <fvaldes@ciencias.unam.mx>
L. Naranjo-Albarrán, ORCID: 0000-0002-9078-6363 <lizbethna@ciencias.unam.mx>
Universidad Nacional Autónoma de México
Ciudad Universitaria, Coyoacán, 04510 Mexico City, Mexico

Abstract. *Context:* The impact of an excellent estimation in planning, budgeting, and control, makes the estimation activities an essential element for the software project success. Several estimation techniques have been developed during the last seven decades. Traditional regression-based is the most often estimation method used in the literature. The generation of models needs a reference database, which is usually a wedge-shaped dataset when real projects are considered. The use of regression-based estimation techniques provides low accuracy with this type of database. *Objective:* Evaluate and provide an alternative to the general practice of using regression-based models, looking if smooth curve methods and variable selection and regularization methods provide better reliability of the estimations based on the wedge-shaped form databases. *Method:* A previous study used a reference database with a wedge-shaped form to build a regression-based estimating model. This paper utilizes smooth curve methods and variable selection and regularization methods to build estimation models, providing an alternative to linear regression models. *Results:* The results show the improvement in the estimation results when smooth curve methods and variable selection and regularization methods are used against regression-based models when wedge-shaped form databases are considered. For example, GAM with all the variables show that the R-squared is for Effort: 0.6864 and for Cost: 0.7581; the MMRE is for Effort: 0.1095 and for Cost: 0.0578. The results for the GAM with LASSO show that the R-squared is for Effort: 0.6836 and for Cost: 0.7519; the MMRE is for Effort: 0.1105 and for Cost: 0.0585. In comparison to the R-squared is for Effort: 0.6790 and for Cost: 0.7540; the MMRE is for Effort: 0.1107 and for Cost: 0.0582 while using MLR.

Keywords: Generalized additive models; LASSO; Software estimation; Effort estimation; Cost estimation; Functional size; COSMIC method

For citation: Valdés-Souto F., Naranjo-Albarrán L. Software project estimation using smooth curve methods and variable selection and regularization methods using a wedge-shape form database. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 123-140. DOI: 10.15514/ISPRAS-2023-35(1)-9

Оценка программного проекта с использованием методов гладких кривых и методов выбора переменных и их регуляризации с использованием базы данных клиновидной формы

Ф. Вальдес-Суто, ORCID: 0000-0001-6736-0666 <fvaldes@ciencias.unam.mx>
Л. Наранхо-Альбарран, ORCID: 0000-0002-9078-6363 <lizbethna@ciencias.unam.mx>

Национальный автономный университет Мексики
Мексика, 04510 Мехико, Койоакан, Университетский городок

Аннотация. *Контекст:* влияние правильной оценки на планирование, составление бюджета и контроль делает действия по оценке важным элементом успеха программного проекта. За последние семь десятилетий было разработано несколько методов оценки. В литературе чаще всего используется традиционный метод оценки, основанный на регрессии. Для создания моделей требуется справочная база данных, которая при рассмотрении реальных проектов обычно представляет собой набор данных клиновидной формы. Использование методов оценки на основе регрессии для этого типа базы данных обеспечивает низкую точность. *Цель:* Оценить и предоставить альтернативу общепринятой практике использования моделей на основе регрессии, выяснив, обеспечивают ли методы гладких кривых и методы регуляризации переменных более высокую надежность оценок, основанных на базах данных клиновидной формы. *Метод:* В предыдущем исследовании использовалась эталонная база данных клиновидной формы для построения модели оценки на основе регрессии. В этой статье используются методы гладких кривых, а также методы выбора переменных и регуляризации для построения моделей оценки, которые представляют собой альтернативу моделям линейной регрессии. *Результаты:* результаты показывают улучшение результатов оценки при использовании методов сглаженной кривой и регуляризации переменных по сравнению с моделями на основе регрессии с использованием клиновидных баз данных.

Ключевые слова: обобщенные аддитивные модели, LASSO, оценка программного обеспечения, оценка усилий, оценка стоимости, функциональный размер, метод COSMIC

Для цитирования: Вальдес-Суто Ф., Наранхо-Альбарран Л. Оценка программного проекта с использованием методов гладких кривых и методов выбора переменных и их регуляризации с использованием базы данных клиновидной формы. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 123-140. DOI: 10.15514/ISPRAS–2023–35(1)–9

1. Introduction

Since the appearance of effort estimation in the 50s [1], it has been a relevant topic for researchers in the academy and managers in the industry.

Estimation is one of the crucial activities in software projects [2] It has been identified that inaccurate estimates in the software development industry are one of the most severe problems that cause the failure of software projects [3] because project estimation has an impact on several aspects like planning, budgeting, control, and success of the software projects [4, 5].

Regression-based estimation approaches dominate the literature, as was mentioned by several authors [5-8]. Although other authors have identified a frequent situation in the literature, the regression techniques are not applied correctly, [5, 9-11].

In order to create a regression-based estimation model, a reference database is required; when the database conforms to a broad set of real projects, a wedge-shaped form is presented very often [5, 12] in this type of database, while the x-axis increases, a greater dispersion is observed in the y-axis [12]. This type of dataset was for the first time by [13], presenting high data dispersion, providing low accuracy. Abran [12] mentions that some of the causes that generate the wedge-shaped dataset are, i.e.: “The project data come from organizations with distinct production processes with correspondingly distinct productivity behavior, or the project data represent the development of

software products with major differences, in terms of software domains, nonfunctional requirements, and other characteristics."

This paper explores the use of some smooth curve methods and variable selection and regularization methods like Generalized Additive Models (GAM) and Least Absolute Shrinkage and Selection Operator (LASSO). A comparison of their performance is made, looking to improve the accuracy of the regression-based model developed in the previous study based in the Mexican Software Metrics Association (AMMS) reference database.

The outline of this paper is as follows. Section 2 shows a literature review of software estimation techniques and problems described directly in the models or the database integration. In section 3, the introduction of the fundamental statistical elements used in the paper. Section 4 presents the case study, estimating the effort and cost of the database from AMMS using the Generalized Additive Models (GAM) and Least Absolute Shrinkage and Selection Operator (LASSO). Section 5 discusses the main results of the case study. Finally, the conclusions are discussed in Section 6.

2. Background

2.1 Software Estimation

For more than 70 years since software estimation appeared [1], it has generated interest in the scientific community and the industry as it is a fundamental piece for the success of software projects [1, 2, 14] and has a crucial impact on the planning and budgeting of software projects [15].

After more than seven decades the software estimation research, it is still an open question [16] and presents many difficulties [16]. However, a great variety of estimation techniques [17-19], estimation methods classifications [1, 5, 7, 8, 9], and estimation process topologies [10, 11] have been created. Each statistic technique has specific features that should be considered to make it proper to solve specific problems [11].

The base to create an estimation model is the reference database that should represent the projects to be estimated. Any estimation model possesses a strong relationship with the input data employed to generate the model: "No cost estimation model (or any other model, come to that) will predict well if it is asked to predict effort for projects that are substantially different in nature to the projects on which the model was built" [9]. A lot of weakness in the databases has been identifying in the literature by several authors [6, 9, 15[.

When an estimation model is generated, there is a need to integrate a reliable reference database based on past completed projects. This database allows identifying relationships between different variables [19] (cost drivers) corresponding to the information of the project. According to Carbonera et al. [15], "most studies (71.67%) use multiple cost-drivers rather than priorate a specific one".

Even when several cost drivers are used, several authors identify the functional size as a critical factor to be included in the reference database [1, 20-24]. This situation makes a sound because "nowadays, the only feature of the software that could be defined in a consensual mode and, in consequence, measured in a standard way is the functional size" [25].

It is important that except for the functional size, most of the other drivers are descriptive or qualitative rather than quantitative, p.e. programming language, primary database, primary operating system, software life cycle, etc. In consequence, the estimation-based on functional size does not represent all the cost estimations for the projects and includes an uncertainty degree derived from the other cost drivers.

When a database is integrated over real projects using as independent variables the functional size, a wedge-shaped dataset is usually observed. In a wedge-shaped dataset, a greater dispersion is observed in the y-axis while the x-axis increases (see Fig. 1), some of the causes that generate the wedge-shaped dataset identified by Abran [12] are "The project data come from organizations with distinct production processes with corresponding distinct productivity behavior, or the project data

represent the development of software products with major differences, in terms of software domains, nonfunctional requirements, and other characteristics."

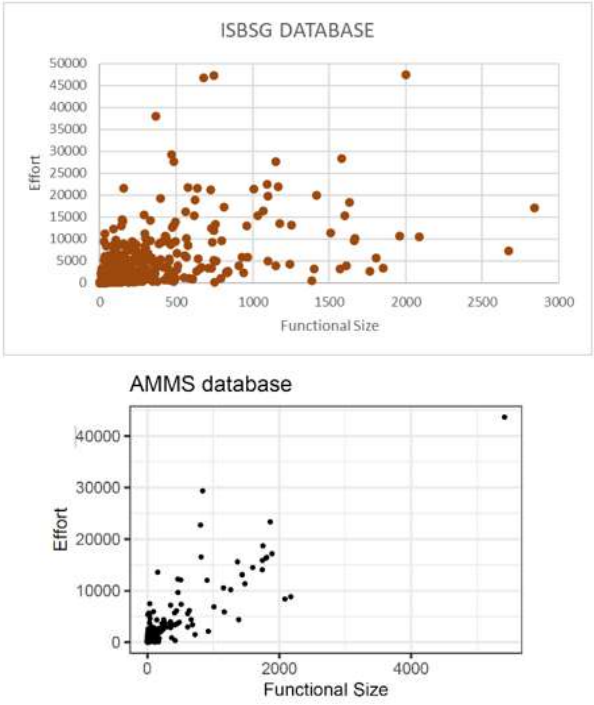

Fig. 1. Wedge-shaped dataset

As there is a high dispersion in a wedge-shaped dataset, the specific features for regression-based models are not accomplished [5]. The use of regression-based estimation techniques may provide low accuracy frequently or may present a cut-off for the accuracy, especially if methods are not adequately applied.

In particular, the software estimation literature reviewed it is not founding the use of smooth curve methods and variable selection and regularization methods. This paper introduces and compares these methods to evaluate their performance with a wedge-shaped form dataset.

2.2 Estimated models performance comparison

In the literature, it has been sought to have a quantitative way of evaluating the performance of estimated models, mainly based on the differences between the real values and the estimated values. Different criteria have been used that determine the confidence of the models used [26-29]. Among the most used criteria in the literature are:

- Coefficient of Determination (R2),
- R2 adjusted,
- Mean Magnitude of Relative Error (MMRE),
- Median Magnitude of Relative Error (MdMRE),
- Standard Deviation of MRE (SDMRE), and
- Prediction level, PRED (x%).

2.2.1 Cross-Validation

A cross-validation framework is considered to validate the results. Specifically, the dataset is randomly split into a training subset composed of 80% of the software projects, and the remaining 20% of the software projects constitute the testing subset. This procedure is repeated independently 500 times, and the results are then averaged.

3. Statistical fundamental elements in estimation

3.1 Smooth Curve Methods

Linear regression models have been studied in Software estimation [5-9], as in many other areas. In Software estimation, the effect of Functional Size on effort or cost is often not linear. In this section, we give a general overview of some statistical methods that allow smooth curve approximations and their properties; we focus on generalized additive models (GAM) and use the regularization and variable selection method LASSO, see [30-33], among others.

3.2 Generalized additive models (GAM)

GLM was proposed by Nelder and Wedderburn (1972) [40], and they extended the multiple linear regression model (MLR) or linear model to include models for binaries and counts data, among others. The GLM is defined by three components [35]:

1) First, the random component Y, with mean $E[Y] = \mu$, where the variable Y has a distribution in the exponential family.
2) Second is the systematic component, where the variables $x_1, x_2, \dots, x_p$ produce a linear predictor $\eta = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$.
3) Third, the link function $g(\cdot)$, that link the random and systematic components, $g(\mu) = \eta$. The required properties of $g(\cdot)$ are strict monotonicity and being twice differentiable in the range of μ.

In GAM, the systematic component η is defined as a sum of smooth functions of the independent variables, $\boldsymbol{x} = (x_1, x_2, \dots, x_p)$:

$$\eta = f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p)$$

Usually, the intercept is included as $f_1(x_1) = \beta_0$ because the f_k are centered for identifiability purposes. The effects of the covariates are assumed additive. The functions f_k are estimated by smoothers.

In the particular case of Y being a random variable with normal distribution, $Normal(\mu, \sigma^2)$, the GAM reduces to the additive model, where the relationship between the mean $E[Y] = \mu$ and the linear predictor $\eta = f_1(x_1) + f_2(x_2) + \cdots + f_p(x_p)$ is defined by the identity link function $\mu = \eta$.

Note that the additive model reduces to the MLR model when the smothers are defined as $f_k(x_k) = \beta_k x_k$.

3.2.1 Smoothing Methods

The smoother functions f_k allow to extend the linear predictor to other sophisticated non-linear curves, the most common are the following, see more details in [36]:

1) *Polynomial regression* extends linear regression and adds extra predictors by raising each one to a power.
2) *Step functions* cut the range of x into k distinct regions producing a quantitative variable, and then fitting a piecewise constant function.
3) *Basis function* consists of having a family of functions or transformations that are applied to x.

4) *Regression splines* involve dividing the range of x into k distinct regions, and within each one, a polynomial function is fitted.
5) *Smoothing splines* are similar to regression splines; they result from minimizing a residual sum of squares criterion subject to a smoothness penalty.
6) *Local regression* is similar to splines, but the regions are allowed to overlap in a smooth way.

3.2.2 Inference and Prediction

In order to fit the generalized additive models, the criterion is to maximize a penalized log-likelihood, or equivalently, minimize a penalized of the least squared errors.

3.3 Least Absolute Shrinkage and Selection Operator (LASSO)

The common selection variable methods retain a subset of the predictor variables and discard the rest; however, this subset selection often exhibits high variance, and it doesn't reduce the prediction error of the full model. Shrinkage methods, consisting of regularization and selection variables, do not suffer as much from high variability.

The LASSO (least absolute shrinkage and selection operator) is a shrinkage method. The LASSO coefficients are defined by

$$\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum\nolimits_{j=1}^{p} \beta_j x_{ij} \right)^2,$$

$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq t.$$

Making t sufficiently small will cause some of the coefficients to be exactly zero, making LASSO like a selection variable method. Choosing t larger than $\sum_{j=1}^{p} |\hat{\beta}_j|$, where $\hat{\beta}_j$ is the least-squares estimates, then the LASSO results in these $\hat{\beta}_j$'s. See [36] and [32] for more details.

When the independent variables belong to predefined groups, for instance, a collection of dummy variables representing the levels of a categorical variable is desirable to shrink and select the group members, to have all coefficients within a group become nonzero or zero simultaneously. The algorithm needed for these cases is the Group LASSO method [37].

In GAM is possible to apply regularization and variable selection methods, see, and particularly to use LASSO, see [38].

4. Case Study

In this section, the analysis of the Effort and Cost estimation models based on the Mexican reference database is described. For detail information about the database conformation see Table 1.

Table 1. Summary of database information.

Variable or Drivers	Effort N = 390
Effort (N = 390)	454.1 (184.8 – 1457.9)
Cost (N = 387, with 3 missing data)	71,067 (28,522 – 226,468)
Functional size	16.70 (6.96 – 125.42)
Type of organization: • private (reference) • governmental	 302 (77.44%) 88 (22.56%)

Development: • maintenance (reference) • new	 323 (82.82%) 67 (17.18%)
Capacity of development: • area inter of systems (reference) • outsourcing or project of key on hand	 315 (80.77%) 75 (19.23%)
Architecture: • client/server (reference) • development web • multilayers • other	 171 (43.85%) 122 (31.28%) 86 (22.05%) 11 (2.82%)
Language: • C# or PHP (reference) • JAVA/J2EE • C++ • other or non-specified • ASP.NET • VisualBasic6	 103 (26.41%) 76 (19.49%) 105 (26.92%) 47 (12.05$) 2 (0.51%) 57 (14.62%)
Operative System: • windows XP or Linux (reference) • UNIX, windows NT, or other • windows 7/8, windows mobile, or windows vista • windows	 215 (55.13%) 59 (15.13%) 46 (11.79%) 70 (17.95%)
Data base: • POSTGRESTSQL, MySQL, or non-specified (reference) • INFORMIX • ORACLE • SQLSERVER	 178 (45.64%) 96 (24.62%) 17 (4.36%) 99 (25.38%)
Process framework: • CMMI (reference) • MAAGTICSI or RUP • other	 326 (83.59%) 9 (2.31%) 55 (14.10%)
Life cycle: • cascade (reference) • Iterative/agile	 328 (84.10%) 62 (15.90%)
Certification of quality model: • yes (reference) • no	 348 (89.23%) 42 (10.77%)
Size of organization: • ≥ 500 employees (reference) • 251-500 employees • Micro and small	 332 (85.13%) 43 (11.02%) 15 (3.85%)

4.1 GAM with LASSO

In the GAM, for the predictor or independent variables we used the functional size, and other categorical variables. As response or dependent variables, we used Effort and Cost in two different analyses. We used the logarithm transformation for the functional size, effort, and cost. Considering

the multicollinearity and the existence of not significant variables, we applied variable selection methods using LASSO looking to integrate some categories.

The models' generation was made using the software R, defining specific code to calculate all the statistic values. The R libraries *mgcv* [38], *grplasso* [39] and *plsmselect* were used for the GAM, the LASSO linear regression for categorical variables, and the GAM with LASSO, respectively.

For the Effort, the results showed that the statistically significant variables are the logarithm of Functional size, Development, Architecture, Language, Operative System, Data Base, Certification of the quality model, and Size of Organization. On the other hand, the variables identified as statistically no significant like: Organization, Capacity of development, Process framework, and Cycle of life, were deleted, i.e., the model is the following:

$$\log(Effort) = \beta_0 + f(\log(Functional\ size)) + \beta_{devlop} Development + \beta_{archi} Architecture + \beta_{lang} Language + \beta_{os} Operative\ system + \beta_{dbase} Data\ base + \beta_{certif} Certification + \beta_{sizeorg} Size\ of\ organization.$$

For the Cost, the results showed that the statistically significant variables are the logarithm of Functional size, Type of organization, Capacity of development, Architecture, Language, Operative system, Data base, and Certification of the quality model. On the other hand, the variables identified as statistically no significant like: Development, Process framework, Cycle of life, and Size of organization were deleted, i.e. the model is the following:

$$\log(Cost) = \beta_0 + f(\log(Functional\ size)) + \beta_{typeorg} Type\ of\ organization + \beta_{capdevelop} Capacity\ of\ development + \beta_{archi} Architecture + \beta_{lang} Language + \beta_{os} Operative\ system + \beta_{dbase} Data\ base + \beta_{certif} Certification.$$

The models' estimated parameters are shown in Tables 2 and 3, respectively, for Effort and Cost. The first columns show the category or variable names associated with the corresponding parameter. Column two shows the estimates parameters. The last column refers to the standard errors. The fourth columns display the p-values related to the test $H_0: \beta = 0$ vs. $H_a: \beta \neq 0$ for each parameter.

Table 2. Summary of the estimated parameters for Effort by using GAM with LASSO.

Response: Effort Coefficients:	**Estimate**	**Standard Error**	**p-value**
Intercept	6.1488	0.1110	<0.0001
Development: new (ref.: maintenance)	0.1854	0.1423	0.1935
Architecture: (reference: client/server or multilayer)	‘’	‘’	‘’
Architecture: development web or other	-0.2226	0.1151	0.0538
Language (reference: C# or PHP or other or non-specified)	‘’	‘’	‘’
Language: JAVA/J2EE or ASP.NET	0.2631	0.1383	0.0579
Language: C++	-0.0418	0.1125	0.7100
Language: Visual Basic 6	-0.1314	0.1746	0.4521
Operative System (reference: windows XP or Linux)	‘’	‘’	‘’
Operative System: UNIX, windows NT, or other	-0.5843	0.1716	0.0007
Operative System: windows 7/8, windows mobile, or windows vista	0.6007	0.1514	<0.0001
Operative System: windows	-0.5830	0.1469	<0.0001
Data base (reference: POSTGRESTSQL, MySQL, SQLSERVER or non-specified)	‘’	‘’	‘’
Data base: INFORMIX	0.5669	0.1206	<0.0001
Data base: ORACLE	0.1568	0.2540	0.5374
Certification of quality model: no (ref: yes)	0.6068	0.2771	0.0291
Size of organization (reference: ≥ 500 employees)	‘’	‘’	‘’
Size of organization: 251-500 employees	0.2664	0.2378	0.2632
Size of organization: Micro and small	-0.1828	0.2549	0.4735

In Table 3, column 2, line 12, the logarithm of the cost decreases by 0.7181 units (estimated parameters equal to -0.7181) if the operative system is UNIX or Windows NT compared to the operative system LINUX or Windows XP, which is the reference for this categorical variable (line 11). In contrast, Table 3, column 2, line 18, increases 0.8341 units (estimated parameter equal to 0.8341) if the certification of the quality model is "No" in comparison to the "Yes" (category of reference for this variable).

Table 3. Summary of the estimated parameters for Cost by using GAM with LASSO.

Response: Cost Coefficients:	**Estimate**	**Standard Error**	**p-value**
Intercept	11.1182	0.1128	<0.0001
Type of organization: governmental (ref.: private)	0.7876	0.2850	0.0060
Capacity of development: outsourcing or project of key on hand (ref.: area inter of systems)	0.6206	0.4121	0.1329
Architecture: (reference: client/server or multilayer)	‘’	‘’	‘’
Architecture: development web or other	-0.4002	0.1179	0.0007
Language (reference: C# or PHP or other or non-specified)	‘’	‘’	‘’
Language: JAVA/J2EE or ASP.NET	0.2504	0.1421	0.0788
Language: C++	-0.0671	0.1166	0.5649
Language: VisualBasic6	-0.0375	0.1834	0.8380
Operative System (reference: windows XP or Linux)	‘’	‘’	‘’
Operative System: UNIX, windows NT, or other	-0.7181	0.2822	0.0113
Operative System: windows 7/8, windows mobile, or windows vista	0.6960	0.1556	<0.0001
Operative System: windows	-0.5297	0.1519	0.0005
Data base (reference: POSTGRESTSQL, MySQL, SQLSERVER or non-specified)	‘’	‘’	‘’
Data base: INFORMIX	0.6397	0.1239	<0.0001
Data base: ORACLE	0.0451	0.2879	0.8753
Certification of quality model: no (ref: yes)	0.8341	0.2341	0.0004

Additionally, there are no differences in the logarithm of the cost for the category C++ of language since the p-value is 0.5649 (Table 3, column 4, line 9), which means that there are significant differences between the language C++ and the language C# or PHP (reference category). However, there are no differences if the architecture development web or other (Table 3, column 4, line 5) in comparison to the architecture client/server or multilayer since the p-value is 0.0007.

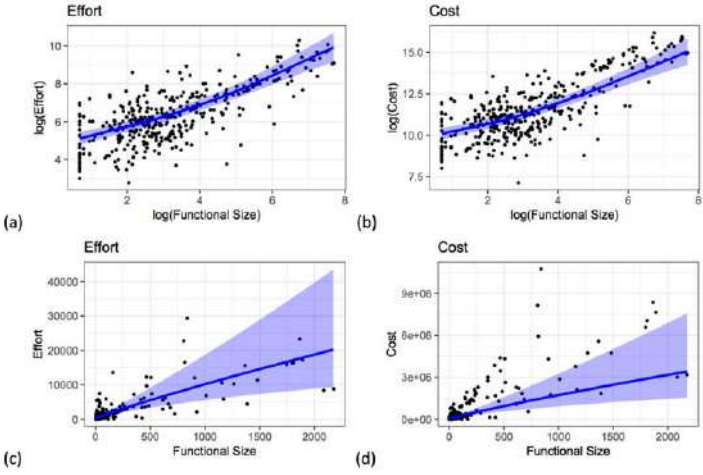

Fig. 2. Fitted line and 95% confidence intervals (shades) for (a) Effort and (b) Cost in logarithmic scale, and (c) Effort and (d) Cost in real scale, by using GAM with LASSO

Fig. 2 shows the fitted lines for reference categories for the categorical independence variables. The shades on the graph provide the 95% pointwise confidence interval for the fitted. To return to the original scale of Y, an exponential function is applied to the predicted values obtained from the model.

From results in Table 2, the estimated model for Effort is:

$$\begin{aligned}\log(Effort) = {} & 6.1488 + s(\log(Functional\ size), 2.46) + 0 * Devlop(maintenance) \\ & + 0.1854 * Devlop(new) + 0 * Archi(client\ server\ \ or\ multilayer) \\ & - 0.2226 * Archi(develpment\ web\ or\ other) + 0 \\ & * Lang(C\#, PHP\ or\ other) + 0.2631 * Lang(JAVA\ J2EE\ or\ ASP.NET) \\ & - 0.0418 * Lang(C++) - 0.1314 * Lang(Visual\ Basic\ 6) + 0 \\ & * OS(windows\ XP\ or\ Linux) - 0.5843 * OS(UNIX, windows\ NT\ or\ other) \\ & + 0.6007 * OS(windows\ 7{,}8, windows\ mobile\ or\ windows\ vista) - 0.5830 \\ & * OS(windows) + 0 \\ & * Dbase(POSTGRESTSQL, MySQL, SQLSERVER\ or\ non\ specified) \\ & + 0.5669 * Dbase(INFORMIX) + 0.1568 * Dbase(ORACLE) + 0 \\ & * Certif(yes) + 0.6068 * Certif(no) + \ 0 * Sizeorg(\geq 500\ employees) \\ & + 0.2664 * Sizeorg(251 - 500\ employees) - 0.1828 \\ & * Sizeorg(Micro\ or\ small).\end{aligned}$$

From results in Table 3, the estimated model for Cost is:

$$\begin{aligned}\log(Cost) = {} & 11.1182 + \ s(\log(Functional\ size), 2.5) + 0 * Typeorg(private) + 0.7876 \\ & * Typeorg(governmental) + 0 * Capdevelop(are\ inter\ of\ systems) \\ & + 0.6206 * Capdevelop(outsourcing\ or\ project\ of\ key\ on\ hand) + 0 \\ & * Archi(client\ server\ or\ multilayer) - 0.4002 \\ & * Archi(development\ web\ or\ other) + 0 * Lang(C\#, PHP\ or\ other) \\ & + 0.2504 * Lang(JAVA\ J2EE\ or\ ASP.NET) - 0.0671 * Lang(C++) \\ & - 0.0375 * Lang(Visual\ Basic\ 6) + \ 0 * OS(windows\ XP\ or\ Linux) \\ & - 0.7181 * OS(UNIX, windows\ NT\ or\ other) + \ 0.6960 \\ & * OS(windows\ 7{,}8, windows\ mobile\ or\ windows\ vista) - 0.5297 \\ & * OS(windows) + 0 \\ & * Dbase(POSTGRESTSQL, MySQL, SQLSERVER\ or\ non\ specified) \\ & + \ 0.6397 * Dbase(INFORMIX) + 0.0451 * Dbase(ORACLE) + 0 \\ & * Certif(yes) + \ 0.8341 * Certif(no).\end{aligned}$$

Table 4 and Table 5 depict the results related to the smooth function $f(\log(Functional\ size))$, in the same format as Tables 2 and 3. Fig. 3 shows the estimated effect of the functional size in the logarithmic scale, as a solid curve, with its 95% confidence limit as dashed lines. Note that the degree of smoothness of the corresponding $f(functional\ size)$ is 2.46 for Effort and 2.5 for Cost. This means that in both cases, the dimension of the smoother is around 2.5.

Table 4. Spline-based smooths for Effort using GAM with LASSO

Approximate significance of smooth terms			
	Effective Degrees of freedom	**F statistic test**	**p-value**
s(log (Functional size))	2.46	100.5	<0.0001

Table 5. Spline-based smooths for Cost using GAM with LASSO

Approximate significance of smooth terms			
	Effective Degrees of freedom	**F statistic test**	**p-value**
s(log (Functional size))	2.5	87	<0.0001

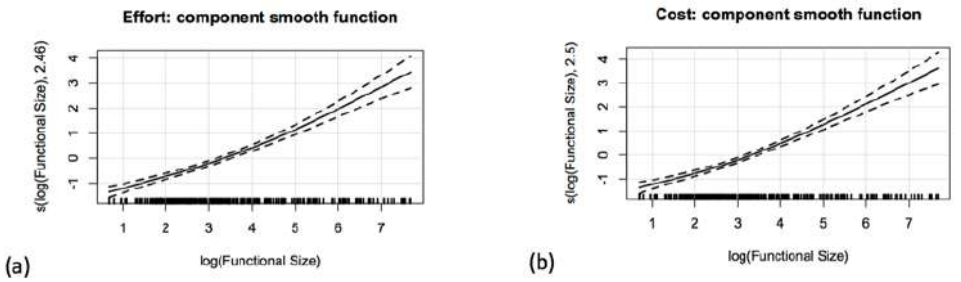

Fig. 3. Component smooth function for the logarithm of Functional Size for the (a) Effort and (b) Cost by using GAM with LASSO.

In GAM, a diagnostic of the residuals, similar to linear regression, must be done. After fitting the model, a diagnostic of the residuals is done to check if the fitted model and assumptions are consistent with the observed data. We used rescaled residuals and graphs to identify homoscedasticity, normality, and influential outliers. Fig. 4 shows the graphs for the residuals for the fitted model for productivity and cost. The quantile-quantile normal plots graphs (a) and (c) visually indicate normality because most dots follow the identity line pattern. The fitted values against the residuals (graphs (b) and (d) in the right) show evidence of constant variance because the dots do not show patterns, which means they show homoscedasticity (constant variance). Moreover, there is no evidence of outliers since there are no residuals with larger values.

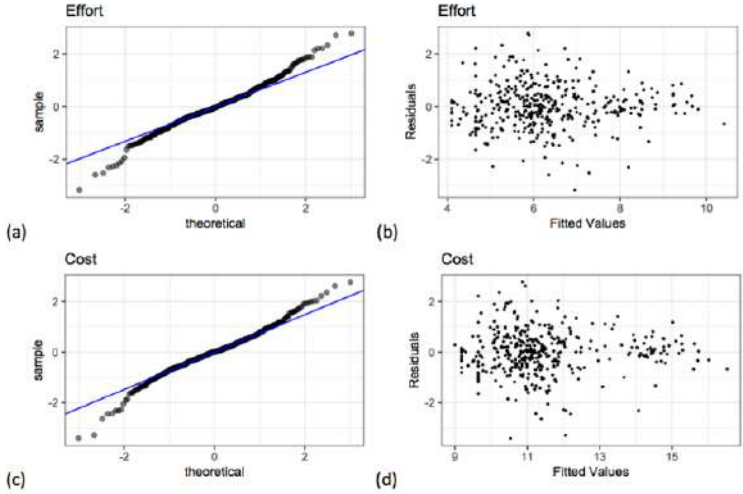

Fig. 4. Residuals for Effort (a, b) and Cost (c, d) by using GAM with LASSO. (a) and (c) quantile-quantile plots to review normal distribution. (b) and (d) residuals vs. fitted values plots to review constant variance

5. Discussion

Three pairs of estimation model techniques were evaluated for comparison purposes, considering the raw data in wedge-shaped form, functional size as the independent variable, and the effort and cost as dependent variables (Fig. 5). The previous study developed the first technique [5], applying a linear regression model (MLR) considering the correct statistical principles and assumptions. The second technique was applying a smooth curve method known as the generalized additive model (GAM). The third technique improved the second approach, using variable selection and regularization methods LASSO (GAM with LASSO), aiming to avoid variables that may be

redundant or irrelevant for predicting the dependent variable, in consequence gathering sparse or simpler models.

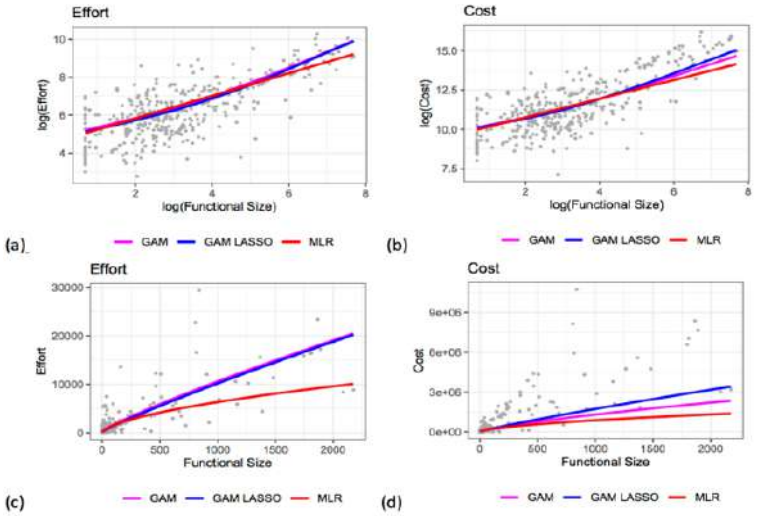

Fig 5. Comparison of the fitted lines for different models. (a) Effort and (b) Cost in logarithmic scale, and (c) Effort and (d) Cost in real scale, by using MLR, GAM, and GAM with LASSO

Tables 6 and 7 show some criteria used to compare the multiple linear regression (MLR), GAM, and GAM with LASSO. Notice that results from GAM and GAM with LASSO are similar. However, GAM with LASSO does not include some variables that are not statistically significant, getting reduced models only with the significant variables.

Table 6. Summary of the criteria for Effort

EFFORT	MLR	GAM	GAM with LASSO
In logarithmic Scale			
R2	0.6790	**0.6864**	0.6836
R2 adjusted	0.6579	0.6645	**0.6705**
MAE	0.6290	**0.6204**	0.6282
MMRE	0.1107	**0.1095**	0.1105
MdMRE	0.0737	0.0743	**0.0730**
SDMRE	0.1356	**0.1340**	0.1346
PRED 25%	0.9025	**0.9051**	**0.9051**
In original Scale			
MAE	773.3	**735.4**	745.0
MMRE	0.8768	0.8594	**0.8557**
MdMRE	0.4353	**0.4345**	0.4434
SDMRE	1.2119	**1.1302**	1.1558
PRED 25%	0.2743	**0.3051**	0.2948

The results for the GAM with all the variables show that the R-squared is for Effort: 0.6864 and for Cost: 0.7581; the R-squared adjusted is for Effort: 0.6645 and for Cost: 0.7413; MAE is for Effort: 0.6204 and for Cost: 0.6355; the MMRE is for Effort: 0.1095 and for Cost: 0.0578. The results for the GAM with LASSO show that the R-squared is for Effort: 0.6836 and for Cost: 0.7519; the R-

squared adjusted is for Effort: 0.6705 and for Cost: 0.7422; MAE is for Effort: 0.6282 and for Cost: 0.6404; the MMRE is for Effort: 0.1105 and for Cost: 0.0585.

Table 7. Summary of the criteria for Cost

COST	MLR	GAM	GAM with LASSO
In logarithmic Scale			
R2	0.7540	**0.7581**	0.7519
R2 adjusted	0.7377	0.7413	**0.7422**
MAE	0.6401	**0.6355**	0.6404
MMRE	0.0582	**0.0578**	0.0585
MdMRE	0.0414	**0.0415**	0.0434
SDMRE	0.0756	**0.0749**	0.0759
PRED 25%	0.9896	**0.9896**	0.9870
In original Scale			
MAE	259498.2	256972.6	**211025.5**
MMRE	0.9113	**0.8993**	0.9575
MdMRE	0.4269	**0.4301**	0.4468
SDMRE	2.2944	2.3511	**1.3845**
PRED 25%	0.2894	**0.3023**	0.2945

5.1 Cross Validation

A cross-validation framework is considered to validate the results. Specifically, the models are fitted using the training subset, and the testing subset is used to predict. Different criteria are computed for the estimated curves in the training subset and then calculated for the predictions in the testing subset. This procedure is independently repeated 500 times, and the results are then averaged.

Table 8. Means and standard deviations of criteria by using MLR, GAM and GAM with LASSO, under the cross-validation scheme for Effort.

Training Data Set			
EFFORT	**MLR**	**GAM**	**GAM with LASSO**
EFFORT	***In logarithmic Scale***	***In logarithmic Scale***	***In logarithmic Scale***
R2	0.6827 ±0.0167	**0.6915 ±0.0169**	0.6869 ±0.0168
R2 adjusted	0.6621 ±0.0178	0.6699 ±0.0180	**0.6703 ±0.0175**
MAE	0.6255 ±0.0153	**0.6145 ±0.0153**	0.6232 ±0.0150
MMRE	0.1100 ±0.0029	**0.1085 ±0.0029**	0.1096 ±0.0028
MdMRE	0.0748 ±0.0032	0.0740 ±0.0034	**0.0740 ±0.0030**
SDMRE	0.1346 ±0.0032	**0.1327 ±0.0033**	0.1337 ±0.0032
PRED 25%	0.9065 ±0.0084	0.9073 ±0.0086	**0.9074 ±0.0088**
EFFORT	***In original scale***	***In original scale***	***In original scale***
MAE	767.2 ±54.4	**717.8 ±51.8**	735.3 ±49.75
MMRE	0.8691 ±0.0407	0.8460 ±0.0416	**0.8447 ±0.0402**
MdMRE	0.4401 ±0.0183	**0.4355 ±0.0185**	0.4402 ±0.0189
SDMRE	1.1930 ±0.1073	**1.0997 ±0.1055**	1.1314 ±0.0996
PRED 25%	0.2934 ±0.0173	**0.3040 ±0.0168**	0.2926 ±0.0180
Testing Data Set			
EFFORT	**MLR**	**GAM**	**GAM with LASSO**
EFFORT	***In logarithmic Scale***	***In logarithmic Scale***	***In logarithmic Scale***
R2	0.6085 ±0.0893	0.6104 ±0.0941	**0.6426 ±0.0728**
R2 adjusted	0.4806 ±0.1185	0.4697 ±0.1302	0.5513 ±0.0925
MAE	0.6819 ±0.0617	0.6750 ±0.0621	**0.6625 ±0.0574**

MMRE	0.1185 ±0.0122	0.1177 ±0.0121	**0.1161 ±0.0118**
MdMRE	0.080 ±0.0107	0.0801 ±0.0109	**0.0793 ±0.0105**
SDMRE	0.1472 ±0.0146	0.1467 ±0.0152	**0.1408 ±0.0125**
PRED 25%	0.8886 ±0.0321	0.8853 ±0.0320	**0.8951 ±0.0314**
EFFORT	***In original scale***	***In original scale***	***In original scale***
MAE	1016.1 ±587.2	946.5 ±384.9	**826.4 ±215.9**
MMRE	1.0057 ±0.2668	0.9834 ±0.2451	**0.9270 ±0.2190**
MdMRE	0.4715 ±0.0547	0.4686 ±0.0576	**0.4678 ±0.0572**
SDMRE	1.8631 ±2.6708	1.6669 ±1.4335	**1.2266 ±0.3749**
PRED 25%	0.2754 ±0.0477	**0.2828 ±0.0478**	0.2758 ±0.0466

Tables 8 and 9 present the summary of the criteria for the Effort and Cost, respectively. Note. that for the training subset, some criteria show better results for GAM, but others show better results for GAM with LASSO; that means that the best estimates result from the GAM or GAM with LASSO compared to the MLR. However, the best results are from the GAM with LASSO for the testing subset. That means GAM with LASSO has the highest predictive capability with this database.

Table 9. Means and standard deviations of criteria by using MLR, GAM and GAM with LASSO, under the cross-validation scheme for Cost

Training Data Set			
COST	**MLR**	**GAM**	**GAM with LASSO**
	In logarithmic Scale	***In logarithmic Scale***	***In logarithmic Scale***
R2	0.7553 ±0.0133	**0.7607 ±0.0132**	0.7536 ±0.0139
R2 adjusted	0.7393 ±0.0141	**0.7439 ±0.0140**	0.7414 ±0.0145
MAE	0.6414 ±0.0162	**0.6347 ±0.0164**	0.6385 ±0.0161
MMRE	0.0582 ±0.0015	**0.0577 ±0.0015**	0.0583 ±0.0015
MdMRE	0.0420 ±0.0019	**0.0420 ±0.0020**	0.0424 ±0.0018
SDMRE	0.0753 ±0.0018	**0.0745 ±0.0019**	0.0756 ±0.0019
PRED 25%	0.9888 ±0.0025	**0.9885 ±0.0027**	**0.9885 ±0.0027**
	In original scale	***In original scale***	***In original scale***
MAE	285293.5 ±34154.5	275634.4 ±33629.2	**210678.5 ±19335.1**
MMRE	0.9194 ±0.0457	**0.9012 ±0.0468**	0.9497 ±0.0550
MdMRE	0.4432 ±0.0194	**0.4468 ±0.0194**	0.4512 ±0.0168
SDMRE	2.4133 ±0.5503	2.3547 ±0.5500	**1.3576 ±0.1892**
PRED 25%	0.2864 ±0.0172	**0.2936 ±0.0174**	0.2928 ±0.0156
Testing Data Set			
COST	**MLR**	**GAM**	**GAM with LASSO**
	In logarithmic Scale	***In logarithmic Scale***	***In logarithmic***
R2	0.6997 ±0.0683	0.7012 ±0.0682	**0.7239 ±0.0603**
R2 adjusted	0.5999 ±0.0911	0.5919 ±0.0936	**0.6583 ±0.0752**
MAE	0.6948 ±0.0656	0.6939 ±0.0667	**0.6712 ±0.0626**
MMRE	0.0625 ±0.0062	0.0625 ±0.0062	**0.0611 ±0.0062**
MdMRE	0.0448 ±0.0065	0.0454 ±0.0065	**0.0449 ±0.0061**
SDMRE	0.0820 ±0.0087	0.0818 ±0.0087	**0.0786 ±0.0076**
PRED 25%	0.9857 ±0.0130	0.9849 ±0.0133	0.9864 ±0.0120
	In original scale	***In original scale***	***In original scale***

MAE	459287.5 ±540631.3	450147.3 ±594499.7	238069.9 ±82057.5
MMRE	1.0683 ±0.3164	1.0532 ±0.3019	**1.0286 ±0.2935**
MdMRE	0.4672 ±0.0591	0.4767 ±0.0615	**0.4754 ±0.0579**
SDMRE	4.0525 ±6.4314	3.8541 ±6.0017	**1.4729 ±0.6324**
PRED 25%	0.2683 ±0.0456	0.2747 ±0.0479	**0.2825 ±0.0481**

6. Conclusions

Software cost/effort estimation has been a relevant topic for more than 60 years in research because of its impact on the industry.

Regression-based estimation approaches have been the more often used technique in the literature, focusing on the estimation model performance comparison. Although, many times, the regression techniques principles are not accomplished.

Additionally, when a database is integrated over real projects and a wedge-shaped form dataset is present, high data dispersion is shown, usually because the project data come from distinct organizations or the project data represent software products with major differences in its characteristics, providing low accuracy in the prediction models generated from the database.

This paper evaluates and provides an alternative to the general practice of using regression-based models. The proposed approach has not been identified in the literature reviewed; it focuses on some smooth curve methods and variable selection and regularization methods like: Generalized Additive Models (GAM) and Least Absolute Shrinkage and Selection Operator (LASSO).

The approach proposed was compared, then the wedge-shaped form database used in a previous study was considered. The performance of the methods generated was evaluated, aiming to improve the accuracy of the MLR model based on the Mexican Software Metrics Association (AMMS) reference database.

A case study is presented to demonstrate how the application of GAM and LASSO over the Mexican Software Metrics Association (AMMS) reference database (wedge-shaped) improves the estimation based on traditional regression-based models (MLR).

In the case of additive models (GAM with normal distribution), the assumptions behind the model are similar to those in multiple linear regression (MLR): residuals must be distributed as Gaussian, being non-correlated and having a constant variance.

This paper used logarithmic transformation to correct problems about normal distribution, constant variance, and influential outliers.

In GAM, the smoother methods extend the linear predictor of generalized linear models (GLM) to other more flexible and non-linear curves, making a more representative model for the data considered in a wedge-shaped database.

The results in this paper show the improvement, providing better accuracy of the generalized additive models (GAM) in comparison to the multiple linear regression (MLR). Moreover, in the cross-validation task, the improvement of the GAM with LASSO on its predictive capability is highest for both dependent variables, Effort, and Cost.

The main contribution of this article is focusing on the generation of estimation models that work better, that is, that offer better precision than those traditionally used, such as simple or multiple linear regression when there are wedge-shaped databases. Additionally, they consider additional drivers, qualitative or quantitative, and optimize them concerning their impact, resulting in simpler models.

Additionally, the explanatory variables should not be correlated to avoid multicollinearity problems and that there are no influential outliers.

Applying the LASSO algorithm, the independent variables are selected, avoiding multicollinearity problems, and choosing those statistically significant, making a sparse model easy to manage and use.

The results show the improvement in the estimation results when smooth curve methods (GAM) and variable selection and regularization methods (LASSO) are used against regression-based models (MLR) when wedge-shaped form databases are considered.

References / Список литературы

[1] Fedotova O., Teixeira L., Alvelos A.H. Software effort estimation with multiple linear regression: Review and practical application. Journal of Information Science and Engineering, vol. 29, issue 5, 2013, pp. 925–945.

[2] Sharma P., Singh J. Systematic literature review on software effort estimation using machine learning approaches. In Proc. of the International Conference on Next Generation Computing and Information Systems (ICNGCIS), 2017: pp. 43-47.

[3] Oliveira A.L.I. Estimation of software project effort with support vector regression. Neurocomputing, vol. 69, issues 13-15, 2006, pp. 1749-1753.

[4] Papadopoulos H., Papatheocharous E., Andreou A.S. Reliable confidence intervals for software effort estimation. In Proc. of the Workshops of the 5th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI-2009), 2009: pp. 211-220.

[5] Valdés-Souto F., Naranjo-Albarrán L. Improving the Software Estimation Models Based on Functional Size through Validation of the Assumptions behind the Linear Regression and the Use of the Confidence Intervals When the Reference Database Presents a Wedge-Shape Form. Programming and Computer Software, vol. 47, issue 8, 2021, pp. 673-693.

[6] Jørgensen M., Shepperd M. A systematic review of software development cost estimation studies. IEEE Transactions on Software Engineering, vol. 33, no. 1, 2007, pp. 33-53.

[7] Braga P.L., Oliveira A.L.I., Meira S.R.L. Software Effort Estimation using Machine Learning Techniques with Robust Confidence Intervals. In Proc. of the 7th International Conference on Hybrid Intelligent Systems (HIS 2007), 2007, pp. 352-357.

[8] Shin M., Goel A.L. Empirical Data Modeling in Software Engineering Using Radial Basis Functions. IEEE Transactions on Software Engineering, vol. 26, no. 6, 2000, pp. 567-576.

[9] Kitchenham B., Mendes E. Why comparative effort prediction studies may be invalid. In Proc. of the 5th International Conference on Predictor Models in Software Engineering, 2009, article no. 4, 5 p.

[10] Bilgaiyan S., Sagnika S. et al. A systematic review on software cost estimation in Agile Software Development. Journal of Engineering Science and Technology Review, vol. 10, issue 4, 2017, pp. 51-64.

[11] Jørgensen M. Regression Models of Software Development Effort Estimation Accuracy and Bias. Empirical Software Engineering, vol. 9, issue 3, 2004, pp. 297-314.

[12] Abran A. Software Project Estimation: The Fundamentals for Providing High Quality Information to Decision Makers, 1st ed. John Wiley & Sons, 2015, 288 p.

[13] Kitchenham B., Taylor N. Software cost models, ICL Technical Journal, vol. 4, issue 1, 1984, pp. 73-102.

[14] Lee T.K., Wei K.T., Ghani A.A.A. Systematic literature review on effort estimation for Open Sources (OSS) web application development, In Proc. of the Future Technologies Conference (FTC), 2016, pp. 1158-1167.

[15] Carbonera C.E., Farias K., Bischoff V. Software development effort estimation: A systematic mapping study. IET Software, vol. 14, issue 4, (2020, pp. 328-344.

[16] Yadav N., Gupta et al. Comparison of COSYSMO Model with Different Software Cost Estimation Techniques. In Proc. of the International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2019, pp. 1-5.

[17] Gray A.R., MacDonell S.G. A Comparison of Techniques for Developing Predictive Models of Software Metrics. Information and Software Technology, vol. 39, issue 6, 1997, pp. 425-437.

[18] Silhavy R., Prokopova Z., Silhavy P. Algorithmic optimization method for effort estimation. Programming and Computer Software, vol. 42, issue 3, 2016, pp. 161-166 / Сильхавы Р., Попова З., Сильхавы П. Алгоритмический метод оптимизации оценки трудозатрат. Программирование, том 42, вып. 3, 2016

г., стр. 64-71.

[19] Durán M., Juárez-Ramírez R. et al. User Story Estimation Based on the Complexity Decomposition Using Bayesian Networks. Programming and Computer Software, vol. 46, issue 8, 2020, pp. 569-583 / Дуран М., Хуарес-Рамирес Р. и др. Оценка пользовательских историй на основе декомпозиции сложности с использованием байесовских сетей. Труды ИСП РАН, том 33, вып. 2, 2021 г., стр. 77-92. DOI: 10.15514/ISPRAS–2021–33(2)–4.

[20] Bourque P., Oligny S. et al. Developing Project Duration Models in Software Engineering. Journal of Computer Science and Technology, vol. 22, 2007, pp. 348-357.

[21] Laird L.M., Brennan M.C. Software Measurement and Estimation: A Practical Approach, Jonh Wiley & Sons, 2006, 280 p.

[22] Koch S., Mitlöhner J. Software project effort estimation with voting rules, Decision Support Systems, vol. 46, issue 4, 2009, pp. 895-901.

[23] De Lucia, Pompella E., Stefanucci S. Assessing effort estimation models for corrective maintenance through e A.mpirical studies, Information and Software Technology, vol. 47, issue 1, 2005, pp. 3-15.

[24] Hill J., Thomas L.C., Allen D.E. Experts' estimates of task durations in software development projects, International Journal of Project Management, vol. 18, issue 1, 2000, pp. 13-21.

[25] ISO/IEC 14143-1:2007 Standard. Information technology — Software measurement — Functional size measurement — Part 1: Definition of concepts. 2007.

[26] Shepperd M., MacDonell S. Evaluating prediction systems in software project estimation. Information and Software Technology, vol. 54, issue 8, 2012, pp. 820-827.

[27] Foss T., Stensrud E. et al, A simulation study of the model evaluation criterion MMRE. IEEE Transactions on Software Engineering, vol. 29, issue 11, 2003, pp. 985-995.

[28] Myrtveit I., Stensrud E., Shepperd M. Reliability and validity in comparative studies of software prediction models. IEEE Transactions on Software Engineering, vol. 31, issue 5, 2005, pp. 380-391.

[29] Jørgensen M., Halkjelsvik T., Liestøl K. When should we (not) use the mean magnitude of relative error (MMRE) as an error measure in software development effort estimation? Information and Software Technology, vol. 143, 2022, article no. 106784, 5 p.

[30] Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning, Data Mining, Inference, and Prediction, 2nd ed. Springer New York, 2009, 745 p.

[31] Yee T.W. Vector Generalized Linear and Additive Models. With an Implementation in R, Springer,, 2015, 613 p.

[32] Hastie T., Tibshirani R., Wainwright M. Statistical Learning with Sparsity The Lasso and Generalizations. Routledge, 2015, 367 p.

[33] Wood S.N. Generalized Additive Models, 2nd ed. Chapman and Hall/CRC, 2017, 476 p.

[34] Hastie T.J., Tibshirani R.J., Sasieni P. Generalized additive models, Routledge, 1990, 352 p.

[35] McCullagh P., Nelder J.A., Enderlein G. Generalized linear models. 2nd ed. Chapman and Hall/CRC, 1989, 532 p.

[36] James G., Witten D. et al, An Introduction to Statistical Learning with Applications in R, 1st ed., Springer, 2013. 440 p.

[37] Yuan M., Lin Y. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society, Series B (Statistical Methodology), vol. 68, issue 1, 2005, pp. 49-67.

[38] Groll A., Hambuckers J. et al. LASSO-type penalization in the framework of generalized additive models for location, scale and shape, Computational Statistics and Data Analysis, vol. 140, 2019, pp. 59-73.

[39] Meier, L., van de Geer S., Bühlmann P., The Group Lasso for Logistic Regression, Journal of the Royal Statistical Society, Series B (Statistical Methodology), vol. 70, issue 1, 2008, pp. 53-7.1

[40] Nelder J., Wedderburn R. Generalized linear models, Journal of the Royal Statistical Society. Series A (General), vol. 135, issue 3, 1972, pp. 370-384.

Information about auth.ors / Информация об авторах

Francisco VALDÉS-SOUTO, Ph.D. in Software Engineering, Associate Professor. Areas of interest: Software Engineering, Software Measurement, Software Estimation, Software, Economy, Software Project Management.

Франсиско ВАЛЬДЕС-СУТО, кандидат наук в области программной инженерии, доцент. Области интересов: разработка программного обеспечения, измерение программного обеспечения, оценка программного обеспечения, программное обеспечение, экономика, управление программными проектами.

Lizbeth NARANJO-ALBARRÁN, Ph.D. in Mathematics, Professor. Areas of interest: Bayesian statistics, biostatistics, generalized linear models, and measurement error models.

Лизбет НАРАНХО-АЛЬБАРРАН, кандидат математических наук, профессор. Области интересов: байесовская статистика, биостатистика, обобщенные линейные модели и модели ошибок измерения.

DOI: 10.15514/ISPRAS-2023-35(1)-10

A Systematic Mapping Study on Process Improvement in Software Requirements Engineering

S. Almeyda, ORCID: 0000-0002-4943-0904 <silvana.almeyda@pucp.edu.pe>
A. Dávila, ORCID: 0000-0003-2455-9768 <abraham.davila@pucp.edu.pe>
Pontificia Universidad Católica del Perú,
Lima, Perú, 15088

Abstract. Software analysis is the process carried out to obtain requirements that reflects the needs of a client's stakeholders and that allows the construction of a software product that meets their expectations. However, it is also known as a process where many defects are injected. In this context, although process improvement has contributed to the software industry, in the case of software requirements it needs to be studied to determine the improvements obtained and established models. In the literature reviewed, a similar mapping study with 4 research question was identified and used as a reference. The objective of this work is to structure the available literature on process improvement in the software requirements engineering (SRE) domain to identify the improvement phases, paradigms, principles, and established models. For this purpose, a systematic mapping study (SMS) was carried out in the most recognized digital databases. The mapping carried out recovered a total of 1,495 studies, and after the process, 86 primary studies were obtained. In this SMS had established and answered 13 research questions. The different models that are applied throughout the software requirements engineering process were identified, and accepted studies were classified and findings on SRE process improvement were collected. The most used models are CMMI, Requirements Engineering Good Practice Guide (REGPG), and ISO/IEC 15504. Also, 62% of accepted studies are of the proposal and evaluation type; that is, they propose a framework and study the implementation of a proposal in one or more case studies respectively. On the other hand, it was found that most of the studies focused on the process improvement analysis phase. Likewise, in contrast with a previous study, proposal and validation type of studies increased in 9 papers each one from 2014 to date. This shows the interest of the scientific community in this domain.

Keywords: software analysis; software requirements engineering; systematic mapping study; software process improvement

For citation: Almeyda S., Dávila A. A Systematic Mapping Study on Process Improvement in Software Requirements Engineering. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 141-162. DOI: 10.15514/ISPRAS-2023-35(1)-10

Acknowledgments. The authors acknowledge Dr. Daniel Méndez for the information shared from his SMS. Authors recognize reviews from members of Grupo de Investigación y Desarrollo en Ingeniería de Software-Pontificia Universidad Católica del Perú (GIDIS-PUCP).

Систематический обзор литературы по совершенствованию процессов разработки требований к программному обеспечению

С. Алмейда, ORCID: 0000-0002-4943-0904 <silvana.almeyda@pucp.edu.pe>
А. Давила, ORCID: 0000-0003-2455-9768 <abraham.davila@pucp.edu.pe>
Папский католический университет Перу,
Перу, 15088, Лима

Аннотация. Анализ программного обеспечения – это процесс, выполняемый для получения требований, которые отражают потребности заказчиков, и позволяющий создать программный продукт, отвечающий их ожиданиям. Однако хорошо известно, что в этом процессе порождается множество дефектов. Хотя усовершенствование процессов разработки внесло свой вклад в индустрию программного обеспечения, процесс разработки требований к программному обеспечению нуждается в дополнительных исследованиях для определения достигнутых улучшений и используемых моделей. В рассмотренных литературных источниках было выявлено и использовано в качестве эталона аналогичное систематическое исследование с четырьмя исследовательскими вопросами. Целью данной работы является структурирование доступной литературы по улучшению процессов в области разработки требований к программному обеспечению для определения этапов совершенствования, парадигм, принципов и моделей. Было проведено систематическое исследование с использованием наиболее признанных баз данных цитирования. В общей сложности было выявлено 1495 исследований, после анализа которых было отобрано 86 основных исследований. Использовались 13 исследовательских вопросов. Были определены различные модели, которые применяются в процессе разработки требований к программному обеспечению, классифицированы выполненные исследования и собраны результаты по улучшению процесса разработки требований. Наиболее часто используемыми моделями являются CMMI, Requirements Engineering Good Practice Guide (REGPG) и ISO/IEC 15504. 62% отобранных исследований относятся к типу предложений и оценок; то есть в них предлагается некоторый фреймворк и изучается возможная реализация предложения в одном или нескольких частных случаях. Было обнаружено, что большинство исследований сосредотачивалось на этапе анализа способов совершенствования процесса. Аналогичным образом, в отличие от предыдущего исследования, с 2014 года по настоящее время количество публикаций типа предложений и валидации увеличилось на 9 статей. Это свидетельствует об интересе научного сообщества к этой области.

Ключевые слова: анализ программного обеспечения; разработка требований к программному обеспечению; систематический структурный обзор литературы; совершенствование процесса разработки требований

Для цитирования: Алмейда С., Давила А. Систематический обзор литературы по совершенствованию процессов разработки требований к программному обеспечению. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 141-162. DOI: 10.15514/ISPRAS-2023-35(1)-10

Благодарности. Авторы признательны д-ру Даниэлю Мендесу за информацию, почерпнутую из его обзора. Благодарим за отзывы членов Группы исследований и разработок в области программной инженерии Папского католического университета Перу.

1. Introduction

The software industry continues to evolve, and new techniques, tools, and good practices are increasingly being applied to improve the software life cycle. Likewise, software continues to revolutionize the world and people's lives, causing a favorable impact on organizations [1]. However, in the software industry, there are many reports of software anomaly related to software requirements [2].

In the software development, there is software requirements engineering process, which is a key stage during the entire software life cycle, since the requirements that reflect the user's needs are obtained [3]. The main measure of the success of a software system is the degree to which it fulfills the purpose for which it was designed [3]. In this context, Software Requirements Engineering (SRE) is the process of discovering that purpose, by identifying the interested parties and their needs,

documenting them in a way that is susceptible to analysis, communication, and subsequent implementation [4]. However, in the industry [5], a few people have had significant experience in requirements management, and many people do not properly distinguish between user requirements and system requirements [5].

According to ISO/IEC/IEEE 29148:2018, requirements engineering is concerned with discovering, eliciting, developing, analyzing, verifying, validating, communicating, documenting and managing requirements [6]. Likewise, it is indicated that the system requirements specification is a structured collection of requirements, that is, it involves functions, performance, design constraints and other attributes for the system and its operational environments and external interfaces [6]. On the other hand, the software requirements specification is also a structured collection of essential requirements that, in this case, involves functions, performance, design constraints, and attributes of the software and its external interfaces [6].

On the other side, Software Process Improvement (SPI) is used in the software industry as a way to move from current inefficient software processes towards processes that achieve the established objectives in terms of quality, time, and productivity [7]. In addition, SPI methodology is defined as a sequence of tasks, tools, and techniques that are performed to plan and implement improvement activities [8].

Software requirements engineering and process improvement have been identified as key processes to improve software quality [7]. In this sense, Méndez's work represents an initial work of the present study [9]. Mendez's study raises 4 research questions about REPI (requirements engineering process improvement): (i) Of what type is the research?, (ii) Which process improvement phases are considered?, (iii) What paradigms do the publications focus on? and (iv) Are the underlying principles of normative or of problem-driven nature? [9].

The objective of this study is to structure the available literature on process improvement in the software requirements engineering (SRE) domain to identify the improvement phases, paradigms, principles, and established models through an SMS in the relevant digital databases such as Scopus, IEEE Xplore, Web of Science, ACM Digital Library, Science Direct, Wiley Online Library, ProQuest, Ebsco, and SpringerLink. It seeks to classify the studies found based on the type of research, process improvement phases, paradigms, principles, and established models of process improvement in SRE. In our study, we have extended and contrasted the study prepared by Méndez et al. [9]. In addition, the problems, factors and metrics that were reported in the SPI implementations in SRE were identified.

The article is organized as follows: in Section 2, the fundamental concepts and related works are presented; in Section 3, the methodology applied to SMS is described; in Section 4, the results found are presented and discussed; in Section 5, the conclusions and future work are established.

2. Background and Related Work

In this section, the concepts of software requirements engineering and software process improvement are presented, and four related studies are presented.

2.1 Software Requirements Engineering

Software requirements engineering (SRE) is the science and discipline related to requirements analysis and management [10], which is an integral part of the software life cycle process connected to other parts through continuous feedback loops [11].

The SRE deals with discovering, developing, tracking, analyzing, qualifying, communicating, and managing requirements that define the system [5], its main objective being to discover the quality requirements that can be implemented in software development [12]. This should make it possible to obtain products that meet customer expectations in terms of functionality and quality [13].

The release of ISO/IEC/IEEE 29148:2011 and its update in 2018 (referred as ISO 29148), represents an important reference since it is articulated to the standards of the system life cycle processes ISO/IEC/IEEE 15288 (referred as ISO 15288) and software life cycle processes ISO/IEC/IEEE 12207 (referred as ISO 12207), and specifies the processes required in engineering activities that result in requirements for systems and software products (including services) throughout the life cycle [6]. These identified requirements must be clear, consistent, modifiable, and traceable to produce a quality product.

2.2 Software Process Improvement

The software process improvement (SPI) is a systematic approach to increase the effectiveness and efficiency of a software development organization and to improve software products [14, 15]. The most used models in the software industry [16] are CMMI, which includes the CMM-Sw and the set of ISO/IEC 15504 with ISO 12207. Also, for the context of small organizations or VSEs (very small entities), the ISO/IEC 29110 family of standards has been published since 2011 [17]. In all cases, it can be observed that they are models that continue to adapt to new contexts, based on previous experiences in the industry of their previous versions.

2.3 Related Works

The works identified as relevant are:

- The Méndez study [9], on software requirements engineering, is an SMS that seeks empirical evidence on existing solutions, their underlying principles, and their research facets, of what he calls REPI (requirements engineering process improvement). One of the results of [9], identifies a research bias on existing proposals instead of SRE improvements according to the individual objectives of the companies. Our study ends up being, in practical terms, an update and extension of the study by [9], as it includes 9 questions and 5 additional databases.
- The study by Kabaale and Kituyi [7], on the design and empirical validation of a theoretical framework to help improve SRE processes in small or medium-sized companies, the key requirements for process improvement being: participation the use of an evolutionary requirements engineering process improvement strategy, change management, training and education, and management engagement.
- The study by Hannola et al. [18] is about the evaluation and improvement of the practices, tools, and techniques used in SRE activities and their problems and needs, carried out through a case study. The authors [18] determine that there is a broad need to improve SRE practices (preparation, analysis, documentation, validation, and management) in the case studied.

3. Systematic Mapping Study

To achieve the objective of the research, a Systematic Mapping Study (SMS) was carried out based on [19], see Fig. 1. The SMS, according to [19, 20], allows defining the general vision for a research area, identifying the amount and type of contribution, as well as the available results.

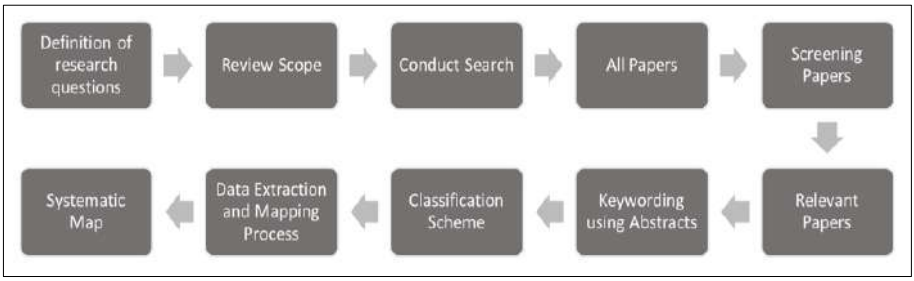

Fig. 1. Systematic mapping process adapted from [19]

3.1 Identification and Scope of the Need

In this study, an SMS is proposed to select and classify the primary studies based on the process improvement phases, paradigms, principles, problems, factors, and metrics that are reported in a requirement engineering process improvement. Likewise, in this study, findings are collected to date and identify the new models that have been developed in software requirements engineering, considering that there is a previous SMS, carried out by [9]. The differences between our work and that of Méndez are: (i) 13 research questions have been established, of which 4 are those of Méndez's previous work; and (ii) the 9 databases (ACM, Scopus, IEEE, Web of Science, Science Direct, Wiley, ProQuest, Ebsco, and SpringerLink) were considered instead of the 5 of Méndez (ACM, SpringerLink, ScienceDirect, Google Scholar, and IEEE Xplore). In addition, the results of Mendéz [9] were taken as a data source.

Table 1. Research question, answer classifier and rationales

Question	Sorter/Rationale
RQ-1. What are the types of study research found?	The classifier proposed by [21] is used, which includes: {Proposal, Evaluation, Validation, Experience, Opinion, Philosophical} and "Exploratory" is added according to [9], which is characterized by studying a problem that is not clearly defined. Identify the type of recurring research has performed in this domain. This RQ is the same to the previous study [9].
RQ-2. What phases of process improvement are considered?	The classifier used in the SMS of [9]: {analysis, construction, validation and SRE process improvement life cycle}. Identify the phase of SPI more studied in this domain. This RQ is the same to the previous study [9].
RQ-3. What paradigms do the studies focus on?	The considered paradigms are taken from [9] and are classified into activities and artifacts. Identify the most studied paradigms in this domain. This RQ is the same as the previous study [9].
RQ-4. Are the principles normative or problem-driven?	The principles considered are taken from [9] and are classified as normative and problem-driven. This RQ is the same as the previous study [9].
RQ-5. What models were used in process improvement?	Identify the most used models in recent years.
RQ-6. What problems have been reported in process improvement projects?	Identify the most recurring problems involved in performing a process improvement in software requirements engineering.
RQ-7. What factors have been reported in SPI implementations in RE?	Identify the reported factors (cultural, organizational, environment, technology, senior management).
RQ-8. What size of the organization is reported in the SPI implementation investigations?	Identify the size of the organizations that carry out improvement implementations
RQ-9. How do you measure the benefit obtained from process improvement?	Identify process improvement metrics in RE.

RQ-10. In which journals or conferences have the publications been made?	The Conference, Journal and Book Chapter classifiers are used. Identify where the authors publish more investigation on the topic.
RQ-11. How has the number of publications on this topic evolved?	Years of publications.
RQ-12. What are the means of publication of the research?	Publication media that concentrate the largest number of studies on this topic.
RQ-13. What are the countries with the greatest contribution from this type of research?	Countries that concentrate the largest number of studies on this topic.

The established research questions and the classifier to be applied to the answers are presented in Table 1.

3.2 Research Strategy

According to [19], this research used the search in relevant digital databases; we worked with PI (Population and Intervention) to build the search chain. The "Population" considered is "Software Process Improvement" and the "Intervention" is "Requirements Engineering" with the aim of covering a greater number of studies related to the research topic, which after finding equivalent terms, remains as presented in Table 2. Before SMS planning, 7 studies of interest from the Scopus database had been identified, which served as a verification mechanism that the search chain can find them and verify if the research questions make sense for those 7 studies.

Table 2. Search string elements

Concept	Terms
Population	"software process improvement" OR SPI
Intervention	"requirements engineering" OR RE OR "software requirement" OR "requirements analysis"
P and I	("software process improvement" OR SPI) AND ("requirements engineering" OR RE OR "software requirement" OR "requirements analysis")

The inclusion (IC) and exclusion (EC) criteria are presented in Table 3. These criteria are applied in the selection process, which is presented in Table 4. It should be noted that, to classify the primary studies by Méndez et al. [9] in the present study, only IC.4 (full-text availability) was applied, so our study includes, as much as possible, Méndez's SMS.

Table 3. Inclusion and exclusion criteria

Id	Criteria
IC.1	They belong to indexed databases.
IC.2	Written in Spanish, English, or Portuguese.
IC.3	Published as Journal Article, Book Chapter, Conference Article.
IC.4	Availability of the full text of the publication.
EC.1	Duplicates or extensions of a study. The less complete version is excluded.
EC.2	Not related to the process improvement field.
EC.3	Not related to the requirements engineering field.

Table 4. Stages and inclusion and exclusion criteria

Stages of the selection process	Criteria
1st. Stage. Extraction of metadata from the considered databases	IC.1, EC.1
2nd. Stage. Title review.	IC.2, IC.3, EC.2, EC.3
3rd. Stage. Review of abstracts	EC.2, EC.3
4th. Stage. Content review.	IC.4, EC.2, EC.3

3.3 Classifiers of Primary Studies

According to what is indicated in the Petersen guide [19], a set of independent indicators of the topic was established. The established classifiers are (i) type of article; (ii) study focus, such as academic, industrial, governmental, project, and organizational; (iii) type of contribution, such as process, method, model, tool, or metric; and (iv) research method, such as case studies, experiment, survey, expert opinion.

In the case of the classifiers of the topic, what is described in [21] and others will be taken into account: (i) solution proposals (ii) solution validation (iii) solution evaluation, (iv) philosophical, (v) experience, and (vi) opinion.

4. Results and Discussion

As defined in the selection and classification process, the search strings were executed in the selected databases, between May and June 2021.

In Fig. 2, the partial results of the selection process are presented. The search was carried out by the first author and the review was carried out by the second author. In addition, of the defined criteria, a general criterion of rejecting only those in which it was very sure to reject was applied, and provisionally accepting, to be resolved in the next stage, any other case. This implied a greater workload in the process, but increased the confidence of not eliminating, in the early stages, some potential primary study. As can be seen in Fig. 2, 1,495 studies were initially obtained, the 58 primary studies from [9] were added and after the process, 86 primary studies were obtained, which are listed in Appendix A. The answers and discussions of the research questions are presented below.

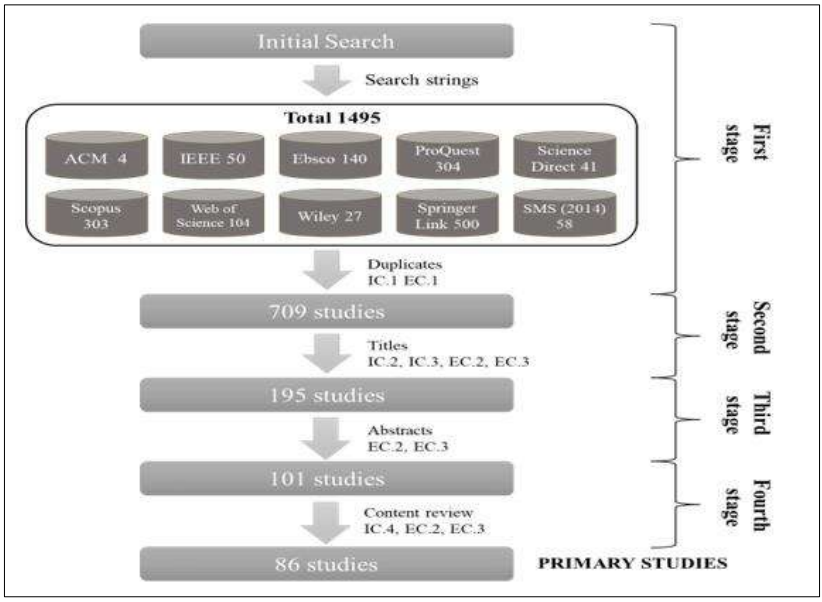

Fig. 2. Study selection results

4.1 RQ-01 What are the types of study research found?

In Table 5, considering the classifiers in Table 1, shows the type of research, the categorized studies and the number of studies, and the percentage ("%") concerning the total (86 studies). The most studied articles are of the proposed type (38,4%) and evaluation (23,3%).

Table 5. Types of research

Types of research	Studies	Quantity	%
Proposal	S01, S02, S06, S07, S08, S15, S16, S26, S28, S30, S32, S35, S39, S46, S48, S49, S50, S52, S55, S56, S59, S61, S66, S69, S76, S78, S79, S80, S81, S82, S83, S84, S85	33	38,4
Evaluation	S04, S05, S22, S23, S27, S34, S37, S38, S40, S44, S45, S47, S51, S53, S54, S64, S67, S68, S71, S72	20	23,3
Validation	S03, S10, S11, S13, S14, S17, S31, S42, S43, S74, S77	11	12,8
Experience	S12, S25, S29, S36, S58, S62, S63, S70, S73, S75	10	11,6
Exploratory	S09, S19, S20, S33, S41, S60, S65, S86	8	9,3
Opinion	S18, S21, S24	3	3,5
Philosophical	S02, S57	2	2,3

The articles of the proposed type, propose frameworks to improve the processes in the software requirements engineering through activities or artifacts. The evaluation-type articles study the implementation of a proposal in one or more case studies to obtain metrics and indicators of the process improvement carried out. Among the least studied are those of opinion (3,5%) and philosophical (2,3%), also considering that the S02 study was classified into two types of research: philosophical and proposed. In addition, in the exploratory type, introduced by [9], 8 studies were found (9,3%).

In Fig. 3, the comparison of the results obtained in the SMS of Méndez [9] and the present study is shown, considering that both take the same classifier (See Table 1). It is observed that the results of each type of research maintain the trend reported in SMS of 2014. Regarding the exploratory type, it is observed that few studies have been published over the years, in the same way as that stated by Méndez [9], this implies that there is little evidence about the problems that organizations face.

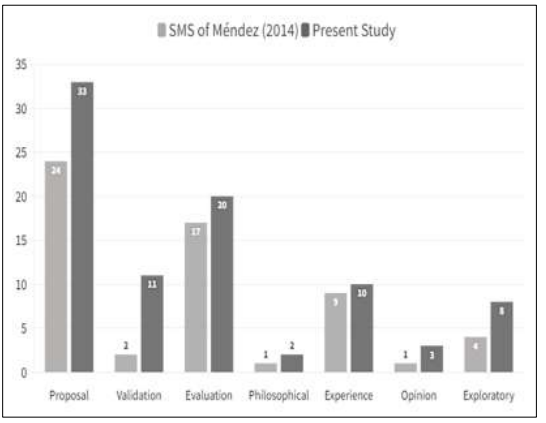

Fig. 3. Comparison of type of research with respect to [9]

4.2 RQ-2 What phases of process improvement are considered?

In Fig 4, the classification of the 86 primary studies distributed in 4 phases is presented. Most focused on the SRE process improvement life cycle and analysis phases with 59 and 23 from primary studies respectively. Based on this, it can be determined that 82 of the primary studies cover the analysis of what happens in the process (or model) of software requirements engineering. Of these, 23 studies are carried out as part of an SRE process improvement life cycle study in a holistic way, which includes all phases, metrics, and general measurements. In addition, it can be observed that there is a second interest, aimed at knowing what happens in a real or realistic context (34%) such as validation (12,8%), experience (11,6%), and exploratory (9,3%). Finally, there are a few studies from the most reflective perspective (5,8%): opinion (3,5%) and philosophical (2,3%).

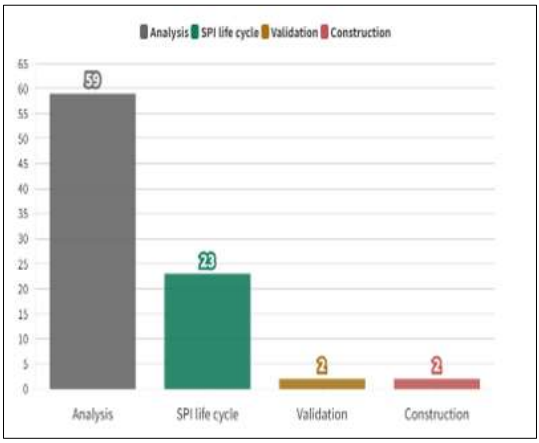

Fig. 4. Process improvement phases

Analogously to the previous question, Table 6 shows the comparison between the SMS of Méndez [9] and the present study. There is an increase in the analysis phases and life cycle of SPI of 21 and 7 studies respectively; while, in the construction and validation phases, the number of studies is maintained. However, from a higher-level perspective, it can be seen that the overall behavior has varied little since the percentage variations are smaller in each phase.

Table 6. Comparison of results on Improvement Phases

Improvement phases	**SMS of Méndez (2014)**		**Present Study (2021)**	
	Quantity	**%**	**Quantity**	**%**
Analysis	38	65,6	59	68,6
Construction	2	3,4	2	2,3
Validation	2	3,4	2	2,3
SPI life cycle	16	27,6	23	26,7
Total	58	100,0	86	100,0

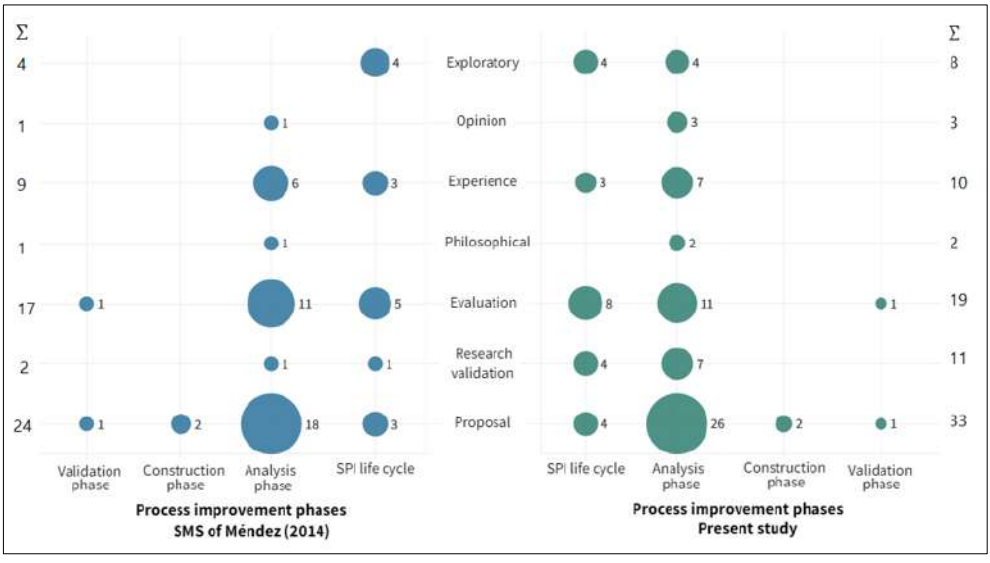

Fig. 5. Process improvement phases vs types of research

In Fig. 5, a diagram of the classification of the process improvement phases and the types of research of the SMS of Méndez [9], left side and the present study (right side) is presented. An increase in studies is observed in the analysis phase of the proposal type (from 18 to 26 studies) and in the research validation (from 1 to 7 studies), which shows the interest of the authors in these aspects. On the other hand, in the construction and validation phases, there is no increase to date.

4.3 RQ-3 What paradigms do the studies focus on?

Paradigms are classified into activities and artifacts, that is, they focus on improving the activities that are part of the SRE process or improving SRE artifacts. In Table 7, the results of the distribution of the primary studies by type of paradigm of the SMS de Méndez of 2014 and the present study (2021) are shown comparatively. It is observed that 81,4% focus on an activity-oriented paradigm, while 10,5% focus on artifacts. Most of the contributions are focused on improvements through models, practices, or strategies focused on SRE activities. Furthermore, in 7 studies not enough information could be found to indicate the paradigm adopted. Furthermore, in contrast, studies, it is observed that, in the intervening 7 years, 22 activity-oriented studies increased, while only 6 oriented artifacts.

Table 7. Comparison of results on activity or artifact-oriented paradigms

Paradigm	SMS of Méndez (2014)		Present Study (2021)	
	Quantity	%	Quantity	%
Activity Orientation	48	82,7	70	81,4
Artefact Orientation	3	5,2	9	10,5
N/A	7	12,1	7	8,1
Total	58	100,0	86	100,0

4.4 RQ-4 Are the principles normative or problem-driven?

The principles were classified, according to [9], as normative or problem-driven (as indicated in Table 1). It is classified as normative when an activity-oriented improvement or SRE artifact is evaluated against an external standard. It is classified as problem-driven when improvement is made against the objectives and problems of an organization. Table 8 shows the distribution of the primary studies according to the principles in a comparative way between the 2014 study by Méndez and the present study (2021). By 2021, it can be seen that 83,7% belong to a normative principle, while 16,3% are driven by problems. Most of the contributions focus on evaluating the activity-oriented or artifact-oriented paradigm against an SRE improvement proposal. In contrast, it can be seen that studies of the type of normative principles have increased by 23 compared to those driven by problems in 5.

Table 8. Comparison of results on normative or problem-driven principles

Principle	SMS de Méndez (2014)		Present Study (2021)	
	Quantity	%	Quantity	%
Normative	49	84,5	72	83,7
Problem-Driven	9	15,5	14	16,3
Total	58	100	86	100

In Fig. 6, the classification of the principles and paradigms is shown. According to the findings, where it is evidenced that, of the activity-oriented studies, 61 are normative and 9 are problem-driven. On the other hand, of those oriented to artifacts, 5 are normative and 4 are problem-driven.

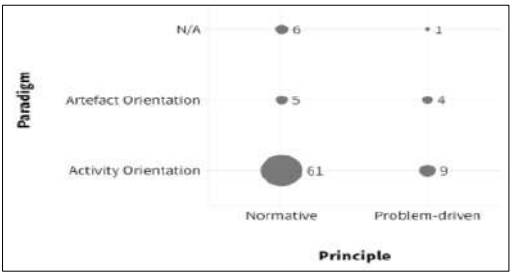

Fig. 6. Principle vs Paradigm

Table 9. Process improvement models in RE

Models	Studies	Quantity
Ad-Hoc	S08, S11, S14, S16, S19, S22, S26, S27, S35, S38, S40, S43, S44, S46, S47, S48, S50, S51, S53, S55, S61, S63, S67, S79, S80, S85	26
CMMI	S01, S03, S04, S07, S32, S33, S49, S58, S66, S70	10
Requirements Engineering Good Practice Guide (REGPG)	S24, S68, S72, S76, S78, S82, S83	7
ISO 15504	S01, S05, S59, S74	4
Requirements Capability Maturity Model (R-CMM)	S30, S52, S54	3
REAIMS	S72, S82, S83	3
Requirements Abstraction Model (RAM)	S42, S45	2
ArtREPI	S10	1
ASAP RE	S71	1
Concern of Requirement Engineering" (CORE)	S64	1
Improvement Framework utilizing light weight assessment and planning (iFLAP).	S37	1
ISO 29110	S13	1
LEGO (Living EnGineering prOcess)	S15	1
Framework of dependent variables	S39	1
Market-driven requirements engineering process model (MDREPM)	S17	1
MESOPYME	S69	1
Method Delphi	S06	1
Modelo descriptivo de RPI	S23	1
NATURE	S84	1
RE maturity measurement framework (REMMF)	S31	1
REPEAT	S81	1
ReqMan	S29	1
Requirements process maturity assessment instrument (RPMAI)	S56	1
Software Requirements Specification (SRS)	S77	1
SRE-MM (software requirements engineering maturity model)	S02	1
Story card Maturity Model (SMM)	S28	1
The QuARS: Quality Analyser for Requirements Specification	S34	1
The Requirements Engineering Process Maturity Model (REPM)	S24	1
University of Hertfordshire model	S24	1

4.5 RQ-5 What models were used in process improvement?

The primary studies were reviewed and it was found that 71 of them presented one or more models of process improvement in software requirements engineering. According to Table 9, it can be observed that 25 studies are of the Ad-Hoc type, that is, they present a process improvement proposal without giving it a specific name, which suggests that many articles proposed their framework is based on software requirements engineering activities. It was also found that the CMMI and Requirements Engineering Good Practice Guide (REGPG) models are the most used with a total of 10 and 7 studies respectively. Other of the most widely found models are ISO 15504, Requirements Capability Maturity Model (R-CMM), and REAIMS with 4, 3, and 3 studies respectively.

4.6 RQ-6 What problems have been reported in process improvement projects?

Of the 86 primary studies, only 9 studies reported problems implementing process improvement in RE. Despite being few studies, these were synthesized to identify the reported problems. Table 10 shows the classified studies.

Table 10. Problems implementing process improvement in RE

Problems	Studies
Process complexity	S51, S53, S58, S73
Cultural change	S37, S72, S73
Lack of authority	S13
Staff turnover	S13
Resistance to change	S19
Informality in the process	S68

4.7 RQ-7 What factors have been reported in SPI implementations in RE?

The studies that reported at least one factor were 15, of which a total of 36 factors could be obtained. According to Table 11, it is observed that the most studied factors in the implementations of SPI in SRE are organizational culture, economic, senior management, and time. Organizational culture involves the way of working of an entire organization, from the defined processes to the principles and values of the workers. The economic factor covers the budgetary part of an organization when making a process improvement. The "top management" factor refers to the commitment of top management when implementing the improvement. On the other hand, 5 studies indicate that, when carrying out a process improvement, time should be considered as a key factor, since there may be cases of delay in improvement activities that could cause the implementation of the activity to take longer than established. Likewise, the column "K&K" was incorporated, which shows the factors categorized and reported by [7], which coincide with the most reported in our study.

Table 11. SPI factors in RE

Factor	Studies	Quantity	K&K
Organizational culture	S10, S12, S13, S60, S72, S73	6	X
Economic	S19, S51, S53, S60, S68	5	X
High direction	S04, S10, S12, S13, S60	5	X
Time	S19, S42, S45, S68, S73	5	--
Study training	S04, S10, S53, S65	4	X
Team engagement	S04, S10, S60	3	X
Organization size	S19, S42, S45	3	--
Soft factors	S10, S14	2	--
Communication	S04	1	--
Stakeholder participation	S65	1	--
Technological	S60	1	--

4.8 RQ-8 What size of the organization is reported in the SPI implementation investigations?

There are 38 primary studies of evaluation, experience, or exploratory type, of which 18 did report an organization size in SRE process improvement implementations. Table 12 shows the size of the organizations as indicated in each study where an improvement was made, from which it can be seen that most involve SMEs.

Table 12. Organization size

Size	Number of Studies	Number of Companies
Small	2	4
Medium	3	4
Large	1	1
SME	9	60
Medium and large	3	11
Not Precise	20	73

4.9 RQ-9 How do you measure the benefit obtained from process improvement?

Of the 86 primary studies, only 8 studies mention any metric used. This suggests that despite having experience and evaluation type studies, an indicator that can measure the benefit obtained after implementation of process improvement in software requirements engineering is not being taken into account. Table 13 lists the set of metrics found.

Table 13. Metrics considered for a process improvement

Metrics	Studies
Process Area Compliance Defects in Software Product Development Validation Area Compliance	S04
Functional errors detected in the product testing and certification stage	S05
Requirements with problems caused by communication problems between distributed teams. Requirements that do not meet customer needs. Non-compliance with the requirements that detail the quality audits of the process.	S36
Process Improvement	S40
Requirements disconnected with the product level. Requirements broken down to function level	S45
Requirements engineering standards used. System requirements that had to be reworked. Time elapsed between system conception and deployment. Project execution time. Effort dedicated to rework. Number of system modifications resulting from RE errors.	S51
It does not require metrics, but used the method based on present value (PV) to perform the financial analysis of this case study.	S34
You don't need metrics, but you used the method Goals-Questions-Metrics (GQM).	S72

4.10 RQ-10 In which journals or conferences have the publications been made?

In Table 14, a list of publications is presented, indicating the type of publication (Column 2 of Type, which can be C = Conference, J = Journal, B = Book Chapter) where the primary studies have been published. Only those publications in which at least two articles have been published are presented. From this Table 14, it is observed that the largest number of studies were published in (i) the

International Conference on Product Focused Software Process Improvement, (ii) International Working Conference on Requirements Engineering: Foundation for Software Quality and (iii) Software Quality Journal. These results allow us to show which conferences or journals generate the greatest interest in the authors of the present study. In addition, 20 studies were published in conferences, 22 in journal and 2 in Book chapter.

Table 14. Publications found

Publication	Type	Studies	Quantity
International Conference on Product Focused Software Process Improvement	C	S09, S10, S11, S14, S18, S19, S23, S32, S33, S49, S56, S73	12
International Working Conference on Requirements Engineering: Foundation for Software Quality	C	S16, S22, S35, S83	4
Software Quality Journal	J	S31, S52, S69, S79	4
IEEE Software	J	S38, S62, S76	3
Requirements Engineering	J	S42, S81, S84	3
IEEE Access	J	S02, S07	2
Journal of Systems and Software	J	S37, S54	2
European Conference on Software Process Improvement	C	S08, S59	2
IEEE Transactions on Software Engineering	J	S27, S74	2
Information and Software Technology	J	S39, S60	2
International Workshop on Database and Expert Systems Applications	C	S75, S77	2
Rationale Management in Software Engineering	B	S46, S47	2
Software Process: Improvement and Practice	J	S34, S82	2
Empirical Software Engineering	J	S58, S85	2
Others (one publication in a journal / conference)	--	Rest of articles	42

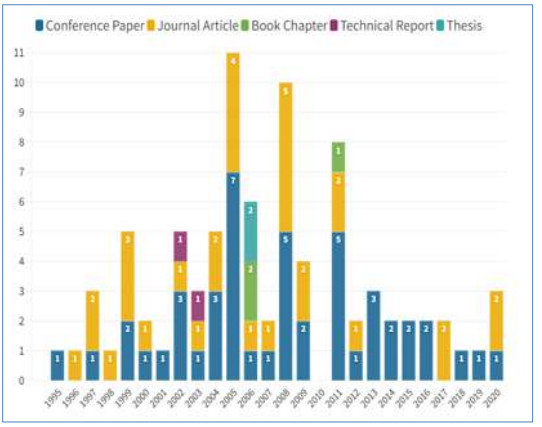

Fig. 7. Publishing media distribution

4.11 RQ-11 How has the number of publications on this topic evolved?

The evolution of the studies was reviewed over time, where publications from 1995 to 2020 were evidenced. It was found that, in 2005, 2008, and 2011, a greater number of publications were made with 11, 10, and 8 studies respectively (See Fig. 7). These years reflect the increased interest of researchers in the subject of study, however, from 2012 onwards there is a stabilization of 2 articles per year on average.

4.12 RQ-12 What are the means of publication of the research?

According to Fig. 7, the crossing of the information of the years and means of publication is presented. It can be seen that, of the 86 primary studies, the most widely used means of publication are conference articles and journal articles with 47 and 32 studies respectively. Likewise, it is observed that 2 technical reports and 2 theses were found.

4.13 RQ-13 What are the countries with the greatest contribution from this type of research?

In Fig. 8, the information crossing of the years and the distribution of the consolidated countries by continent according to the author's affiliation is presented. It should be noted that the countries are counted by author, therefore, in many cases, it is considered 2, 3, or 4 countries per study. Likewise, if there are two or more authors from the same country in a study, it is only counted as one. In total, 31 countries were consolidated and it was detected that those with the greatest contribution are the United Kingdom (18), Germany (11), Canada (8), Australia (7), and Sweden (7). In addition, it is reported that the greatest contribution to this research topic comes from Europe (52%), Asia (20%), and North America (12%).

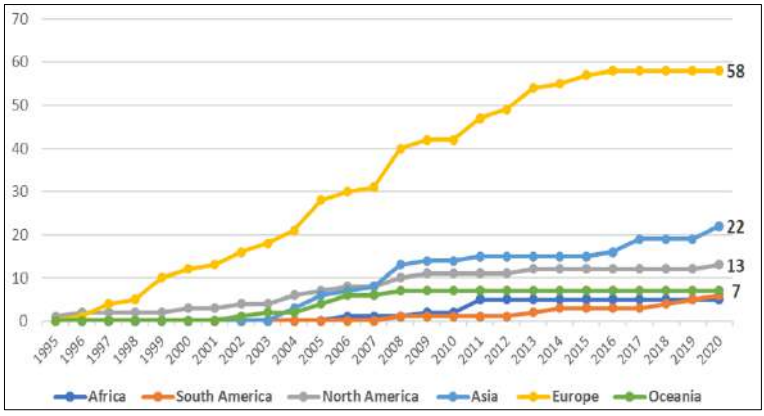

Fig. 8. Distribution of countries by continent based on authors

4.14 Threats to validity

According to Ampatzoglou [22], literature review validity threats are classified into three categories: validity of study selection, the validity of data, and the validity of the research. For the present investigation, the three categories were considered, which are detailed below:

- **Validity of the selection of studies:** The search for studies was carried out in the most relevant digital databases, the search chain was elaborated using the most representative terms of the PI strategy suggested by [19] and established the inclusion and exclusion criteria. In addition, to mitigate the threat of not finding relevant studies, with the search chain, seven studies related to the present research topic were first identified and the chain was executed in Scopus, verifying that the seven studies were in the set of results. Likewise, it was verified that they answered the proposed research questions. Also, it was established to work with the scheme that the doubt about an article was accepted to be resolved in the next stage. Although the work increases, the risk of omitting an article decreases, so a job with greater effort was chosen. However, despite this, there is a possibility that some articles from other repositories that are not being considered in this research will not be found. On the other hand, it should be noted that, in the study selection process, duplicate studies or extensions were identified and the less complete version was excluded.
- **Data validity:** To mitigate this threat, five-stage data extraction, and classification procedure

was developed to ensure the integrity of the investigation. The procedure was first reviewed by the principal investigator and subsequently validated by an experienced investigator. Furthermore, the present study is not threatened by the small sample size, since, in the initial search, 1,495 studies were obtained, leaving 86 primary studies, which were published in various conferences and important journals in the software industry.

- **Research validity:** This research is based on Petersen's methodology [19]. The study defined 10 research questions and 3 bibliometric questions that contribute to the achievement of the study objective. In the study, it was decided not to establish a date range, so findings are collected to date. Likewise, the results obtained from 4 questions were contrasted with those of the SMS of Méndez [9] being essentially similar. In addition, to generalize the results, all process improvements in software requirements engineering were examined, without focusing only on evaluation-type studies (case studies).

5. Conclusion and Future Work

This research presents a systematic mapping study of the literature on software process improvement in requirements engineering. The Petersen methodology was followed, applying a study selection and classification procedure based on inclusion and exclusion criteria. The search for studies was carried out in nine (9) relevant digital databases and; furthermore, after the selection process, the studies selected by Méndez [9] were considered as a special source. Finally, 86 primary studies were obtained. This SMS reports that the most used models are: CMMI, Requirements Engineering Good Practice Guide (REGPG), and ISO 15504. Likewise, 26 Ad-Hoc type studies were found, that is, they presented an improvement proposal based on the SRE activities without giving the framework a specific name.

Regarding the types of research, the proposals and evaluation type works were the most preferred by the authors. Regarding the process improvement phases, it was detected that the majority of studies focused on the analysis phase. On the other hand, the primary studies were classified into paradigms (activity or artifact) and principles (normative or problem-driven), the findings reported that most studies are activity-oriented and normative.

Regarding process improvement projects, it was reported that the factors of organizational culture, economics, senior management, and time are the most studied in the implementations of SPI in RE. In addition, the organizations that participated in an improvement project were classified based on size (small, medium, and large), obtaining as a result that 9 studies involve SMEs.

Since the present study considered the first 4 questions equal to Méndez's SMS [9], the findings reported in the SMS and the present investigation were compared, showing that the results of the types of investigation, paradigms, and principles follow the trend of the SMS carried out in 2014. In the process improvement phases, an increase in studies involved in the analysis phase and SPI life cycle of 21 and 7 studies respectively is reported, while, in the phases of construction and validation, no further studies have been presented since 2014. Likewise, from 2014 to date, proposal and validation type of studies increased in 9 papers each one.

From the SMS, it can be noted that: There is a great variety of models, but only 5 have more than two publications; which reveals that there is no consensus on a model for this domain. Furthermore, it is known that CMMI and ISO/IEC 15504 are not focused on SRE. Analysis is investigated as an improvement phase more than the rest. The approach followed is to implement a (normative) model that seeks a solution to a specific problem. There are few articles reporting problems, measurements and factors. Our study shows that the concern of researchers in this field remains especially from the empirical point of view (validation) and the search for solutions (proposal).

As future work, it is suggested to continue with the investigations of the proposed solution type, taking their studies to the implementation through case studies, to validate if the authors' proposals respond positively in terms of process improvement in the analysis of requirements.

References

[1] Kazman R. Pasquale L. Software Engineering in Society. IEEE Software, vol. 37, issue 1, 2020, pp. 7-9.

[2] Hastie S., Wojewoda S. Standish Group 2015 Chaos Report - Q&A with Jennifer Lynchю. 2015. Available at: https://www.infoq.com/articles/standish-chaos-2015, accessed Apr. 27, 2019.

[3] ISO/IEC/IEEE. ISO/IEC/IEEE 12207:2017 Systems and Software Engineering - Software Life Cycle Processes. 2017.

[4] Nuseibeh B., Easterbrook S. Requirements Engineering : A Roadmap. In Proc. of the Conference on The Future of Software Engineering, 2000, pp. 35-46.

[5] Dick J., Hull E., Jackson K. Requirements Engineering.4th ed. Springer, 2017, 259 p.

[6] ISO/IEC/IEEE. ISO/IEC/IEEE 29148:2018 Systems and Software Engineering - Life Cycle Processes - Requirements Engineering, 2018.

[7] Kabaale E., Kituyi G.M. A Theoretical Framework for Requirements Engineering and Process Improvement in Small and Medium Software Companies. Business Process Management Journal, vol. 21, issue 1, 2015, pp. 80-99.

[8] Aysolmaz B., Demirörs O. A Detailed Software Process Improvement Methodology: BG-SPI. in Systems, Software and Service Process Improvement. Communications in Computer and Information Science, 2011, vol. 172, Springer, pp. 97-108.

[9] Méndez D., Ognawala S. et al. Where do We Stand in Requirements Engineering Improvement Today?: First Results from a Mapping Study. In Proc. of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2014, article no. 58, 4 p.

[10] ISO/IEC/IEEE. ISO/IEC/IEEE 24765:2017 Systems and software engineering – Vocabulary, 2017.

[11] Macaulay L. Requirements Engineering. Springer, 2012, 220 p.

[12] Pandey D., Suman U., Ramani A.K. An Effective Requirement Engineering Process Model for Software Development and Requirements Management. In Proc. of the International Conference on Advances in Recent Technologies in Communication and Computing, 2010, pp. 287-291.

[13] Bjarnason E., Runeson P. et al. Challenges and Practices in Aligning Requirements with Verification and Validation: a Case Study of Six Companies. Empirical Software Engineering, vol. 19, issue 6, 2014, pp. 1809-1855.

[14] Unterkalmsteiner M., Gorschek T. et al. Evaluation and Measurement of Software Process Improvement-A Systematic Literature Review. IEEE Transactions on Software Engineering, vol. 38, issue 2, 2012, pp. 398-424.

[15] O'Regan G. Introduction to Software Process Improvement. Springer, 2011, 270 p.

[16] Von Wangenheim C.G.V., Hauck J.C.R. et al. Systematic Literature Review of Software Process Capability/Maturity Models. In Proc. of the International Conference on Software Process. Improvement And Capability dEtermination (SPICE), 2010, 9 p.

[17] O'Connor R.V., Laporte C.Y. The evolution of the ISO/IEC 29110 set of standards and guides. International Journal of Information Technologies and Systems Approach, vol. 10, issue 1, 2017, article no. 1, 21 p.

[18] Hannola L., Oinonen P., Nikuia U. Assessing and Improving the Front End Activities of Software Development. International Journal of Business Information Systems, vol. 7, issue 1, 2011, pp. 41-59, 2011.

[19] Petersen K., Vakkalanka S., Kuzniarz L. Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update. Information and Software Technology, vol. 64, 2015, pp. 1-18.

[20] Kitchenham B., Charters S. Guidelines for Performing Systematic Literature Reviews in Software Engineering. Version 2.3. EBSE Technical Report EBSE-2007-012007. Keele University, University of Durham, 2007, 65 p.

[21] Wieringa R., Maiden N. et al. Requirements Engineering Paper Classification and Evaluation Criteria: A Proposal and a Discussion. Requirements Engineering, vol. 11, issue 1, 2006, pp. 102-107.

[22] Ampatzoglou A., Bibi S. et al. Identifying, Categorizing and Mitigating Threats to Validity in Software Engineering Secondary Studies. Information and Software Technology, vol. 106, 2019, pp. 201-230.

Appendix A. List of Primary Studies

Table A. Primary Studies

ID	Authors	Year	Title
S01	Gasca-Hurtado G.P., Muñoz M.	2020	A Path for the Implementation of Best Practices for Software Requirements Management Process Using a Multimodel Environment

S02	Akbar M.A., Alsanad A., Mahmood S., Alsanad A.A., Gumaei A.	2020	A Systematic Study to Improve the Requirements Engineering Process in the Domain of Global Software Development
S03	Keshta I.M., Niazi M., Alshayeb M.	2020	Towards the implementation of requirements management specific practices (SP 1.1 and SP 1.2) for small- And medium-sized software development organisations
S04	Bayona-Oré S., Chamilco J., Perez D.	2019	Software process improvement: Requirements management, verification and validation [Mejora de Procesos Software: Gestión de Requisitos, Verificación y Validación]
S05	Allasi D., Dávila A.	2018	Financial impact on the adoption of software validation tasks in the analysis phase: A business case
S06	Iqbal, J; Ahmad, R; Nasir, MHNM; Khan, M	2017	Significant Requirements Engineering Practices for Outsourced Mobile Application Development
S07	Keshta I., Niazi M., Alshayeb M.	2017	Towards Implementation of Requirements Management Specific Practices (SP1.3 and SP1.4) for Saudi Arabian Small and Medium Sized Software Development Organizations
S08	Ito M.	2016	Cardion.spec: An Approach to Improve the Requirements Specification Written in the Natural Language Through the Formal Method
S09	Femmer H., Hauptmann B., Eder,S., Moser D.	2016	Quality Assurance of Requirements Artifacts in Practice: A Case Study and a Process Proposal
S10	Méndez D., Wagner S.	2015	A case study on artefact-based re improvement in practice
S11	Reggio G., Leotta M., Ricca F.	2015	A Method for Requirements Capture and Specification Based on Disciplined Use Cases and Screen Mockups
S12	Khankaew S., Riddle S.	2014	A review of practice and problems in requirements engineering in small and medium software enterprises in Thailand
S13	Alvarez J.J., Hurtado J.A.	2014	Implementing the software requirements engineering practices of the ISO 29110-5-1-1 standard with the unified process
S14	Méndez D., Wieringa R.	2013	Improving requirements engineering by artefact orientation
S15	Buglione, L., Hauck, J. C. R., von Wangenheim, C. G., & McCaffery, F.	2013	Improving Estimates by Hybriding CMMI and Requirement Engineering Maturity Models – A LEGO Application
S16	Bennett-Therkildsen, J., Jørgensen J., Nørskov N., Rubin M.	2013	Redefinition of the Requirements Engineer Role in Mjølner's Software Development Process
S17	Gorschek, T; Gomes, A; Pettersson, A; Torkar, R	2012	Introduction of a process maturity model for market-driven product management and requirements engineering
S18	Frank Houdek	2012	Improving requirements engineering processes: Impressions during one decade of improvement at daimler
S19	Kabaale E., Nabukenya J.	2011	A systematic approach to requirements engineering process improvement in small and medium enterprises: An exploratory study
S20	Shahid M., Ibrahim S., Mahrin M.N.	2011	An evaluation of requirements management and traceability tools
S21	Kelly S., Keenan F., McCaffery F.	2011	Challenges for requirements development: An industry perspective
S22	Markov, G.A. and Hoffmann, A. and Creighton, O.	2011	Requirements engineering process improvement: an industrial case study
S23	Zawedde, A.S.A. and Klabbers, M.D.M. and Williams, D.D. and van den Brand, M.G.J.M.	2011	Understanding the Dynamics of Requirements Process Improvement: A New Approach
S24	Sawyer, P.	2011	Maturing Requirements Engineering Process Maturity Models
S25	Teufl, S. and Khalil, M. and Mou, D. and Geisberger, E.	2011	Experience with content-based requirements engineering assessments
S26	Zawedde, A.	2011	Building a Case for a Dynamic Requirements Process Improvement Model
S27	Napier, NP; Mathiassen, L; Johnson, RD	2009	Combining Perceptions and Prescriptions in Requirements Engineering Process Assessment: An Industrial Case Study
S28	Patel C., Ramachandran M.	2009	Story card Maturity Model (SMM): A process improvement framework for agile requirements engineering practices

S29	Adam, S. and Doerr, J. and Eisenbarth, M.	2009	Lessons Learned from Best Practice-Oriented Process Improvement in Requirements Engineering: A Glance into Current Industrial RE Application
S30	Solemon, B. and Shahibuddin, S. and Abd Ghani, A.A.	2009	Re-defining the Requirements Engineering Process Improvement Model
S31	Niazi, M; Cox, K; Verner, J	2008	A measurement framework for assessing the maturity of requirements engineering process
S32	Niazi M., Hickman C., Ahmad R., Ali Babar M.	2008	A model for requirements change management: Implementation of CMMI level 2 specific practice
S33	Niazi M., Ali Babar M., Ibrahim S.	2008	An empirical study identifying high perceived value practices of CMMI level 2
S34	Raffo, David and Ferguson, Robert and Setamanit, Siri-on and Sethanandha, Bhuricha	2008	Evaluating the impact of requirements analysis tools using simulation
S35	Brinkkemper S., Van De Weerd I., Saeki M., Versendaal J.	2008	Process improvement in requirements management: A method engineering approach
S36	Alves C., Valença G., Sotero T., Mendes J.	2008	Requirements engineering process improvement: A knowledge transfer experience
S37	Pettersson, F.; Ivarsson, M.; Gorschek, T.; Ã–hman, P.	2008	A practitioner's guide to light weight software process assessment and improvement planning
S38	Dörr, J. and Adam, S. and Eisenbarth, M. and Ehresmann, M.	2008	Implementing Requirements Engineering Processes: Using Cooperative Self-Assessment and Improvement
S39	Gorschek, T. and Davis, A.M.	2008	Requirements engineering: In search of the dependent variables
S40	Tripathy S., Mishra S., Shrivastava, A.,Singh, V.K., Darbari, M.	2008	An Efficient Evaluation of Requirements Engineering Process Maturity Assessment and Improvement
S41	Lee E.-S., Bae J.-M.	2007	Design opportunity tree for requirement management and software process improvement
S42	Gorschek, T; Garre, P; Larsson, SBM; Wohlin, C	2007	Industry evaluation of the requirements abstraction model
S43	Napier N.P., Mathiassen L., Johnson R.D.	2006	Negotiating response-ability and repeat-ability in requirements engineering
S44	Palyagar, B. and Moisiadis, F.	2006	Validating Requirements Engineering Process Improvements - A Case Study
S45	Gorschek, T.	2006	Requirements Engineering Supporting Technical Product Management
S46	Hagge, L. and Houdek, F. and Lappe, K. and Paech, B.	2006	Using Patterns for Sharing Requirements Engineering Process Rationales
S47	Palyagar, B. and Richards, D.	2006	Capturing and Reusing Rationale Associated with Requirements Engineering Process Improvement: A Case Study
S48	Yamaç, P.I.	2006	Improvement proposal for a Software Requirements Management Process
S49	Cerón R., Dueñas J.C., Serrano E., Capilla R.	2005	A meta-model for requirements engineering in system family context for software process improvement using CMMI
S50	Jo J.-H., Choi H.-J.	2005	A reflective case study of software process improvement for a small-scale project
S51	Sommerville I., Ransom J.	2005	An empirical study of industrial requirements engineering process assessment and improvement
S52	Beecham S., Hall T., Rainer A.	2005	Defining a requirements process improvement model
S53	Nikula U., Sajaniemi J.	2005	Tackling the complexity of requirements engineering process improvement by partitioning the improvement task
S54	Beecham, Sarah;Hall, Tracy;Britton, Carol;Cottee, Michaela;Austen, Rainer	2005	Using an expert panel to validate a requirements process improvement model
S55	Xu, H. and Sawyer, P. and Sommerville, I.	2005	Requirement Process Establishment and Improvement: From the Viewpoint of Cybernetics
S56	Niazi, M.	2005	An instrument for measuring the maturity of requirements engineering process
S57	Ning, A. and Hou, H. and Hua, Q. and Yu, B. and Hao, K.	2005	Requirements engineering processes improvement: a systematic view

S58	Damian, D. and Chisan, J. and Vaidyanathasamy, L. and Pal, Y.	2005	Requirements Engineering and Downstream Software Development: Findings from a Case Study
S59	Rifaut, A.	2005	Goal-Driven requirements engineering for supporting the ISO 15504 assessment process
S60	Kauppinen, M; Vartiainen, M; Kontio, J; Kujala, S; Sulonen, R	2004	Implementing requirements engineering processes throughout organizations: success factors and challenges
S61	Kamal, Aatif; Ali, Arshad; Anjum, Ashiq; Nazir, Fawad; Ahmad, Hafiz Farooq; Burki, Hamid Abbas; Suguri, Hiroki; Shah, Umair Ali; Tarar, Tallat Hussain	2004	Process maturity for software project outsourcing.
S62	Daneva, M.	2004	ERP Requirements Engineering Practice: Lessons Learned
S63	Doerr, J. and Paech, B. and Koehler, M.	2004	Requirements engineering process improvement based on an information model
S64	Jiang, L. and Eberlein, A. and Far, B.H.	2004	Case studies on the application of the CORE model for requirements engineering process assessment
S65	Niazi M., Shastry S.	2003	Critical Success Factors for the Improvement of Requirements Engineering Process
S66	Beecham, S. and Hall, T. and Rainer, A.	2003	Building a requirements process improvement model
S67	Gorschek, T. and Wohlin, C.	2003	Identification of Improvement Issues Using a Lightweight Triangulation Approach
S68	Kauppinen, M. and Aaltio, T. and Kujala, S.	2002	Lessons Learned from Applying the Requirements Engineering Good Practice Guide for Process Improvement
S69	Calvo-Manzano Villalón J.A., Agustín G.C., Gilabert T.S.F., De Amescua Seco A., Sánchez L.G., Cota M.P.	2002	Experiences in the Application of Software Process Improvement in SMES
S70	Damian, D. and Zowghi, D. and Vaidyanathasamy, L. and Pal, Y.	2002	An Industrial Experience in Process Improvement: An Early Assessment at the Australian Center for Unisys Software
S71	Daneva, M.	2002	Using Maturity Assessments to Understand the ERP Requirements Engineering Process
S72	Kauppinen, M.	2002	A Practical Framework for Systematic Improvement of Requirements Engineering Processes
S73	Kauppinen M., Kujala S.	2001	Starting improvement of requirements engineering processes: An experience report
S74	Emam K.E., Birk A.	2000	Validating the ISO/IEC 15504 measure of software requirements analysis process capability
S75	Houdek, F. and Pohl, K.	2000	Analyzing Requirements Engineering Processes: A Case Study
S76	Sawyer P., Sommerville I., Viller S.	1999	Capturing the benefits of requirements engineering
S77	J. Andrade; J. Ares; O. Dieste; R. Garcia; M. Lopez; S. Rodriguez; L. Verde	1999	Creation of an automated management software requirements environment: A practical experience
S78	Sawyer, P. and Sommerville, I. and Kotonya, G.	1999	Improving Market-Driven RE Processes
S79	Williams, D.W. and Hall, T. and Kennedy, M.	1999	A Framework for Improving the Requirements Engineering Process Management
S80	Williams, D. and Kennedy, M.	1999	A Framework for Improving the Requirements Engineering Process Effectiveness
S81	Regnell, Björn; Beremark, Per; Eklundh, Ola	1998	A market-driven requirements engineering process: Results from an industrial process improvement programme
S82	Sawyer, P. and Sommerville, I. and Viller, S.	1997	Requirements Process Improvement through the Phased Introduction of Good Practice
S83	Sawyer, P. and Sommerville, I. and Viller, S.	1997	Improving the Requirements Process
S84	Grosz, G. and Rolland, C. and Schwer, S. and Souveyet, C. and Plihon, V. and Si-Said, S. and Achour, C.B. and Gnaho, C.	1997	Modelling and engineering the requirements engineering process: An overview of the NATURE approach

S85	El Emam, K.; Madhavji, N.H.	1996	An instrument for measuring the success of the requirements engineering process in information systems development
S86	El Emam, K. and Madhavji, N.H.	1995	Measuring the success of requirements engineering processes

Information about authors / Информация об авторах

Silvia ALMEIDA, Master of Science, Analyst. Research interests: Information Analysis, Project Management, Process Improvement.

Сильвия АЛЬМЕЙДА, магистр наук, аналитик. Область научных интересов: анализ информации, управление проектами, совершенствование процессов.

Abraham DÁVILA is a Principal Professor of the Computer Engineering program and is a Doctoral Candidate in Software Engineering, in the field of process improvement. Field of scientific interests: Software engineering, Software quality process, Software quality product, Education in software engineering, innovations based on software.

Авраам ДАВИЛА – профессор программы компьютерной инженерии и докторант в области программной инженерии. Область научных интересов: программная инженерия, процесс качества программного обеспечения, образование в области программной инженерии, инновации на основе программного обеспечения.

DOI: 10.15514/ISPRAS-2023-35(1)-11

A Systematic Mapping Study on Software Testing in the DevOps Context

[1] *B. Pando, ORCID: 0000-0002-8133-631X <brian.pando@unas.edu.pe>*
[2] *A. Dávila, ORCID: 0000-0003-2455-9768 <abraham.davila@pucp.edu.pe>*

[1] *National Agrarian University of La Selva,*
Tingo María, Huánuco, Peru
[2] *Pontificia Universidad Católica del Perú,*
Lima, Perú, 15088

Abstract. DevOps is a philosophy and framework that allows software development and operations teams to work in a coordinated manner, with the purpose of developing and releasing software quickly and cheaply. However, the effectiveness and benefits of DevOps depend on several factors, as reported in the literature. In particular, several studies have been published on software test automation, which is a cornerstone for the continuous integration phase in DevOps, which needs to be identified and classified. This study consolidates and classifies the existing literature on automated tests in the DevOps context. For the study, a systematic mapping study was performed to identify and classify papers on automated testing in DevOps based on 8 research questions. In the query of 6 relevant databases, 3,312 were obtained; and then, after the selection process, 299 papers were selected as primary studies. Researchers maintain a continuing and growing interest in software testing in the DevOps context. Most of the research (71.2%) is carried out in the industry and is done on web applications and SOA. The most reported types of tests are unit and integration tests.

Keywords: DevOps; software testing; systematic mapping study

For citation: Pando B., Dávila A. A Systematic Mapping Study on Software Testing in the DevOps Context. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 163-188. DOI: 10.15514/ISPRAS-2023-35(1)-11

Acknowledgments. Authors recognize reviews from members of Grupo de Investigación y Desarrollo en Ingeniería de Software – Pontificia Universidad Católica del Perú GIDIS-PUCP).

Систематический обзор литературы по тестированию программного обеспечения в контексте DevOps

[1] *Б. Пандо, ORCID: 0000-0002-8133-631X <brian.pando@unas.edu.pe>*
[2] *А. Давила, ORCID: 0000-0003-2455-9768 <abraham.davila@pucp.edu.pe>*

[1] *Национальный аграрный университет Ла-Сельвы,*
Перу, Уануко, Тинго Мария
[2] *Папский католический университет Перу,*
Перу, 15088, Лима

Аннотация. DevOps – это философия и инфраструктура, которые позволяют группам разработчиков и эксплуатации программного обеспечения работать скоординированно с целью быстрой и дешевой разработки и выпуска программного обеспечения. Однако, как сообщается в литературе, эффективность и преимущества DevOps зависят от нескольких факторов. В частности, было опубликовано несколько результатов исследований по автоматизации тестирования программного обеспечения, которая является краеугольным камнем фазы непрерывной интеграции в DevOps. Эти работы нуждаются в идентификации и классификации. В нашем исследовании консолидируется и классифицируется существующая литература по автоматизированному тестированию в контексте

DevOps. Для исследования было проведено систематическое сопоставление литературных источников на основе 8 исследовательских вопросов. Путем выполнения запросов к шести уместным базам данных было получено 3312 статей. После процесса отбора 299 статей были выбраны в качестве основных. Исследователи сохраняют постоянный и растущий интерес к тестированию программного обеспечения в контексте DevOps. Большая часть исследований (71,2%) проводится в производственной сфере и затрагивают веб-приложения и SOA. Наиболее распространенными типами тестов являются модульные и интеграционные тесты.

Ключевые слова: DevOps; тестирование программного обеспечения; систематический обзор литературы

Для цитирования: Пандо Б., Давила А. Систематический обзор литературы по тестированию программного обеспечения в контексте DevOps. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 163-188. DOI: 10.15514/ISPRAS-2023-35(1)-11

Благодарности. Авторы признательны за отзывы членам Группы исследований и разработок в области программной инженерии Папского католического университета Перу.

1. Introduction

The software market constantly demands strategies that allow it to deal with changes quickly [1], [2]. However, these strategies must maintain quality and avoid the costs of application downtime and failure [3]. Although agile methods are presented as a good alternative; these do not close the cycle until the delivery and operation of the software [4]. In this context, the DevOps philosophy and framework extends the agile methodology to deliver applications quickly and frequently [5], improving performance and costs [6], and taking care of the product quality [7], [8], [9]. So, with the support of top management [10], DevOps can represent a great opportunity for companies of any size to gain a foothold in the market [11]. For this reason, various companies have been adopting it [12] or have adopted plans [13]. Also, DevOps is a key factor in the microservices architecture [14].

In the field of the software industry, the introduction of the term DevOps, in 2008 [15], made it possible to articulate a set of practices that had already been taking place. In particular, the continuous integration practice that is based, among others, on automated tests [16], which represents one of the vital factors for its adoption [17], despite long-standing efforts to resolve this challenge [18], [19]. On the other hand, in the academic field, various literature review studies have been carried out where: (i) it is pointed out that the concept of DevOps is not completely defined [20]; (ii) the definitions, practices and benefits of DevOps are categorized [21]; (iii) the relevant aspects are determined [22], [23]; (iv) the factors that interrupt its adoption are identified [24]; (v) the influence on the product is presented [7]; and, (vi) in [2], a strong need to respond quickly to the market is reported and that DevOps helps to address this problem.

Since software testing is a critical factor for the adoption of DevOps [25], it should be reviewed how it is being applied in the reported cases. For this reason, this paper consolidates and classifies the literature on applied software testing in a DevOps context. The paper is organized as follows: in Section 2, the fundamental aspects of this study are presented; in Section 3, the Systematic Mapping Study (SMS) is described; in Section 4, the results of the SMS are presented; and, in Section 5, the conclusions are established.

2. Background

In this section, DevOps and software testing are briefly presented; as well as the works related to this study.

2.1 DevOps

DevOps integrates the teams that are usually separated (development and operations), focusing on delivering value quickly and continuously, based on 4 dimensions [22]: collaboration, automation, measurement and monitoring. In DevOps [4], it has extended the already known practices of agile

methods, distributing them in 3 phases: construction phase, deployment phase, and operation phase. In addition, it incorporates some existing practices such as: continuous integration [26], continuous deployment [27], continuous delivery [28], and continuous testing [29].

2.2 Software Testing in Agile and DevOps Context

Software testing [30] are activities in the software development process to determine that the software has the expected behavior under a list of test cases. Tests can be categorized, according to [31]: (i) object of the test (unit, integration and system); and (ii) test objective (acceptance, installation, alpha, beta, regression, performance, security, load, recovery, bottom-out, interface, configuration, usability, and interaction).

In the agile context, agile tests have shown their benefits [32], [33], being necessary that the software-testers are present from the collection of requirements [34] and maintain fluid communication, both formal and informal, with the programmers [35].

3. Research Metodology

In this study, a Systematic Mapping Study (SMS) was performed. The SMS proposed by [36] is a research technique to identify and characterize all available studies on a given topic, using a reliable and verifiable methodology.

3.1 Scope and Research Questions

Software testing is one of the pillars to encourage good results in DevOps contexts [5], [8], and on which various publications have been made that require identification, studied and classified. For this reason, an SMS was performed with the purpose of identifying the levels of software tests that are being used in these contexts, as well as the authors, their evolution and the regions where the subject is being investigated, among others. The research questions and considerations for the answers are:

RQ-1 What is the evolution of the publication of papers on software testing in the DevOps contexts? The year of publication was taken as relevant data.

RQ-2 What kind of research has been done in software testing in DevOps? The types of research, adapted from [37], are: (i) survey/interview, (ii) case study, (iii) multiple case study, (iv) replication study, (v) review or literature mapping, and, (vi) background theory.

RQ-3 What kinds of proposals have been presented on software testing in DevOps? The types of proposals are an emerging classification and can be: methods, tools, frameworks.

RQ-4 What levels of software testing are used in DevOps? The possible test levels, depending on the object of the test, are: unit, integration, user, security and load/performance [31].

RQ-5 What programming languages and software testing tools are used in DevOps? Possible answers, at least initially, are: Java, C, PHP, JS, Xunit, Selenium.

RQ-6 In what types of applications are software testing used in the DevOps context? The possible answers, at least initially, are: web, desktop, console, mobile.

RQ-7 What infrastructure tools are used for software testing in DevOps? Possible answers are: Jenkins, Travis, Docker, AWS, Azure.

RQ-8 In what types of activities do software testing occur in DevOps? Possible answers are: Continuous Integration, Continuous Deployment, Continuous Delivery. Also, are security tests mentioned?

3.2 Search Query

Searches were performed according to a generated search string of the population (P) and intervention (I) as suggested [36]. The terms related to (P) are: DevOps, Continuous Integration, Continuous Testing, Continuous Deployment, and Continuous Delivery. The term related to I is: test. Then, the search string stayed as "P and I": "(DevOps OR "continuous integration" OR "continuous deployment" OR "continuous delivery" OR "continuous testing") AND test*". Although a string in English was searched, papers written in Spanish and Portuguese were also considered. Also, to allow for as many results as possible, the date was not restricted. The digital databases are: IEEE Xplore, SCOPUS, ScienceDirect. ACM Digital Library, Web of Science and Willey, selected for their scientific relevance and access to them.

3.3 Data Selection

The selection process was defined in four stages, where the inclusion criteria (IC) and exclusion criteria (EC) are applied (see Table 1); and according to [36] the quality assessment is omitted since relevant digital databases were chosen. The defined selection process has the following stages:

- In the first stage, obtaining the metadata, the EC.1 and IC.2 criteria are used, and the Parsif.al web application to facilitate some operations, such as discarding duplicate papers in the different databases.
- In the second stage, the title is read and EC.2 is applied, to rule out papers that are not related to the subject of software testing in the DevOps contexts.
- In the third stage, reading the summaries, IC.2, IC.3, EC.3 is applied.
- In the fourth stage, a quick reading is made of the content of the study to determine its relevance to the subject of software testing in DevOps contexts and criteria IC.2, IC.3, EC.3 and EC.4 are applied. Likewise, at this stage, the papers to which the full text is not available (EC.5) are withdrawn.

Table 1. Inclusion Criteria (IC) and Exclusion Criteria (EC)

Id	Criteria
IC.1	IC.1 Paper in indexed journals or conferences whose memories are indexed.
IC.2	IC.2 Paper with content in English, Spanish or Portuguese.
IC.3	IC.3 Paper that focuses on software testing in the DevOps context.
EC.1	EC.1 Duplicate article.
EC.2	EC.2 Paper outside the topic of software and DevOps.
EC.3	EC.3 Paper that does not mention software testing levels or strategies.
EC.4	EC.4 Secondary or tertiary articles.
EC.5	EC.5 Paper whose content is not available.

To extract the data, a file was created (see Table 2) to be used in a spreadsheet and collect the data from the papers on it.

Table 2. Structure of the data extraction form

Data	Detail	Question
Id Study	Unique identifier of the study created for the MSL.	General
Title	Title of the paper.	RQ-1
Author	List of authors of the paper.	RQ-1
The year	Year in which the paper was published.	RQ-1
Type of publication	Journal or conference where the paper was published.	RQ-1
Country	Country of affiliation of the authors.	RQ-1
Research type	Categorizes the type of research of the paper.	RQ-2
Context	Categorizes between the academic or industrial context of the paper.	RQ-2
Domain	Categorizes the business domain where the item was applied.	RQ-2

Type of proposal	Categorizes the type of proposal of the paper, if applicable.	RQ-3
Test Level	Categorizes the test levels mentioned in the paper.	RQ-3, RQ-4
Continuous phase	Categorizes the continuous phase mentioned in the paper.	RQ-4
Method	Identifies the method or good development practices.	RQ-4
Testing tool	Identifies the testing tool used.	RQ-5
Version Control	Identifies the tool used for code version management.	RQ-5
Programming language	Programming language mentioned in the paper.	RQ-5, RQ-6
Type App	Type of software developed in the paper.	RQ-6
Architecture type	Type of the architecture of the application developed in the paper.	RQ-6
Infrastructure tool	Collects the infrastructure tools used in the research presented in the paper.	RQ-7
Security	Identifies if the paper mentions the security tests	RQ-8
Teams in DevOps	Identifies if the paper addresses Devs, Ops or both teams.	RQ-8

4. Results

The searches in the considered databases were carried out between June and July 2021. For each database, the search string was adapted according to its own rules (see Table 3). Of the 3,312 papers found, it was processed stage by stage until reaching a total of 299 primary studies. The process was based on the inclusion and exclusion criteria according to the study planning. Table 4 shows the number of papers that remained after each stage. In addition, 15 (5%) papers were withdrawn because the full text was not available, even after having searched different sources. The list of primary studies is available in Appendix A.

Table 3. Database search string

Source	Search string	Quantity
IEEE	(("All Metadata":Devops) OR ("All Metadata":"Continuous Integration") OR ("All Metadata":"Continuous Deployment") OR ("All Metadata":"Continuous Delivery") OR ("All Metadata":"Continuous Testing")) AND (("All Metadata":Test*))	529
Scopus	TITLE-ABS-KEY ((devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND test*)	1,561
ACM	Title: ((Devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND Test*) OR Abstract:((Devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND Test*) OR Keyword:((Devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND Test*)	246
Science Direct	Title-keyword-abstract (Devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND Test	462
Web of Science	TITLE-ABS-KEY ((devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND test*)	432
Willey	TITLE-ABS-KEY ((devops OR "Continuous Integration" OR "Continuous Deployment" OR "Continuous Delivery" OR "Continuous Testing") AND test*)	82
Total		3,312

Table 4. Search results by stage

Procedure	Selection Criteria	Total
First stage	EC.1, IC.1	1,179
Second stage	EC.2	928
Third stage	IC.2, IC.3, EC.3	344
Fourth Stage	IC.2, IC.3, EC.3, EC.4, EC5	299

4.1 RQ1 What is the evolution of the publication of papers on software testing in the DevOps contexts?

From the selected primary studies, from 2011 to Jun-2021 (see Figure 1a), it is observed that the level of publications has been increasing from the beginning, which shows the importance of software testing in DevOps contexts and that coincides with those indicated by [38]. In addition, this growth is expected to continue in the following years.

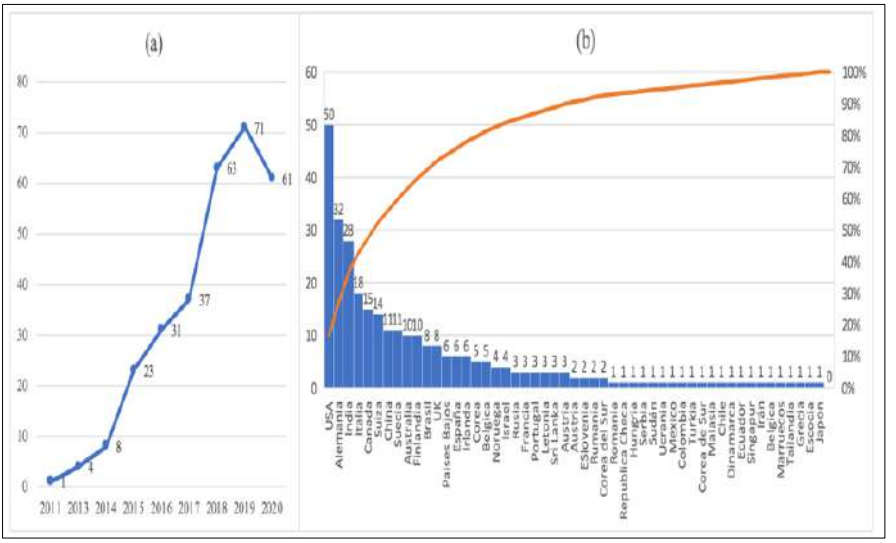

Fig 1. Evolution of publications per year (a), and publications by country (b) in DevOps software testing

Although the topic of DevOps is of global importance, it can be seen (see Figure 1b) that according to the Pareto rule 80% of the studies are concentrated in 16 countries: USA (16.7%), Germany (10.7%), India (9.4%), Italy (6%), Canada (5%), Switzerland (4.7%), China (3.7%), Sweden (3.7%) Australia (3.3%), Finland (3.3%) and Brazil (2.7%), UK (2.7%), the Netherlands (2%), Spain (2%), Ireland (2%), Korea (1.7%) and Belgium (1.7%).

On the other hand, the publication media where they have been published 4 or more primary studies are 14 media and are presented in Table 5.

Table 5. Frequency of primary studies by means of communication, which have 4 or more publications

Venue	Count
Lecture Notes in Computer Science	11
Communications in Computer and Information Science	9
CEUR Workshop Proceedings	9
International Conference on Software Engineering	9
ACM International Conference Proceeding Series	7
International Workshop on Quality-Aware DevOps (QUDOS)	7
IEEE Software	5
Euromicro Conference on Software Engineering and Advanced Application (SEAA)	5
Information and Software Technology	5
IEEE International Conference on Software Maintenance and Evolution (ICSME)	5
Advances in Intelligent Systems and Computing	4
International Conference on Software Testing, Verification and Validation (ICSTW)	4
International Conference on Software Analysis, Evolution, and Reengineering (SANER)	4
Journal of System and Software	4

4.2 RQ2 What types of research have been done on software testing in DevOps?

From the primary studies, on types of research (see Figure 2a), there are two predominant types of research (78.6%): 136 study cases (45.5%) and 99 experiments (33.1%); which are mostly reported in the industry. This orientation, towards the more empirical side, makes sense, since the cases and experiments of integrating Dev and Ops work teams materialize in real projects. This result coincides with the study by [39], who also found a high percentage (20%) of papers at the industry level. Of the remaining group of research types, it can be pointed out that those related to opinion-research allow concepts, ideas, lessons to be proposed when dealing with software testing in DevOps. Likewise, the result of the research context shows that 213 (71.2%) according to Figure 2b, are papers in the industry, compared to 29 (9.7%) are papers in academia; which reinforces the idea of the previous result.

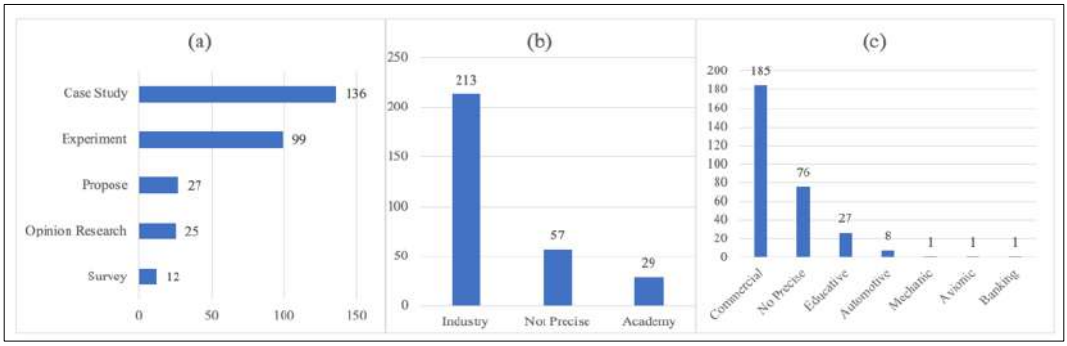

Fig 2. Distribution of primary studies of software testing in the DevOps context, by: (a) research type, (b) research context, and (c) application domain

Finally, from the perspective of the application domain (see Figure 2c), 185 (61.8%) papers have been applied to commercial solutions, that is, applications to sell products, rent services, etc. Likewise, an interesting focus is seen in the education sector, where 27 (9%) primary studies have focused on applications for education (support for the teaching/learning process).

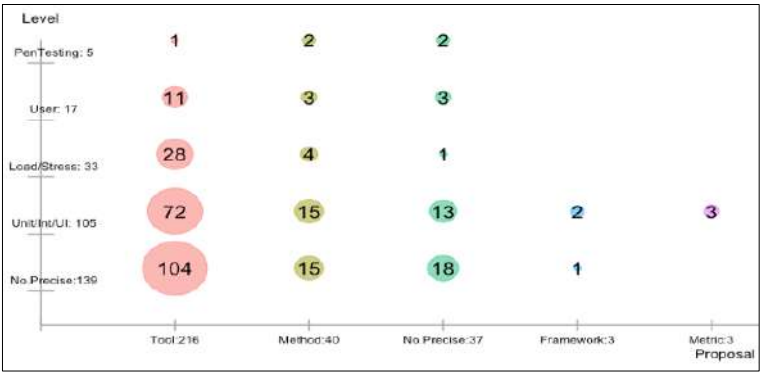

Fig 3. Types of proposals by test levels

4.3 RQ-3 What kinds of proposals have been presented on software testing in DevOps?

In Figure 3, it can be seen that 216 (72.2% primary studies) propose tools to support DevOps contexts, incorporating software testing as part of them. Furthermore, 40 (14%) and 3 (1%) papers propose methods and frameworks respectively to support testing work. These results are in agreement with the results obtained in the study by [40], they point out that tools and frameworks have been proposed and that most are based on unit tests and automated integration.

4.4 RQ-4 What levels of software testing are used in DevOps?

In relation to the levels of software testing used in DevOps (see Figure 4a), the response of "not precise" are 139 papers (46.5%). Despite this, these works do indicate that software testing is a DevOps necessity, but they do not specify the levels of testing in the DevOps context. In the case of the primary studies, which do indicate the levels of proof, it follows that: (i) 122 papers (35.1%) have reported unit and user interface tests; (ii) 33 papers (11%) have reported load and stress; and, (iii) the rest are user tests and penetration testing (pen-testing). The work of [41] and [42] agree that unit and integration tests are among the most studied. Likewise, [41] adds functional, load and stress tests as the most studied with 63.6% of the total studies reviewed; and, they consider that security tests are much less studied with 3.6%. According to reviews from [43] and [44], GUI and accessibility tests are still pending challenges in continuous contexts.

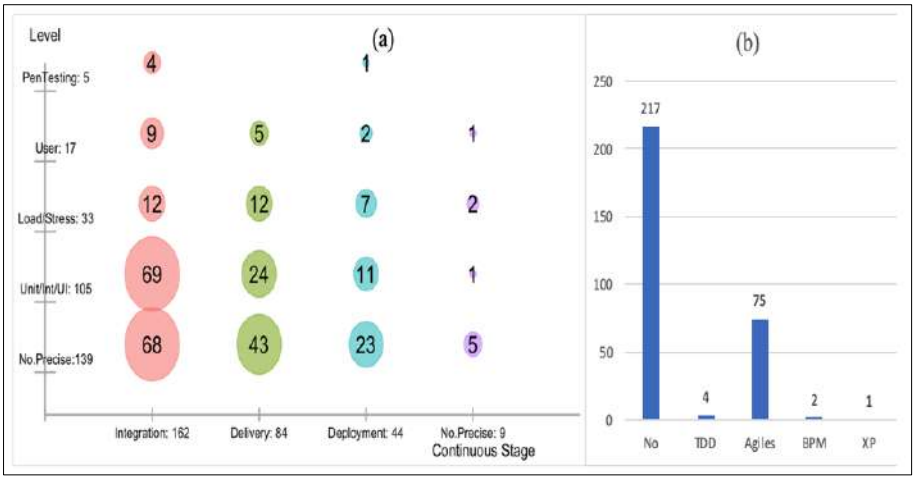

Fig 4. Test levels (a) grouped by continuous phase and (b) methods used in software testing in DevOps

According to this Figure 4a, in relation to the opportunity in the use of software tests in DevOps, it can be pointed out that 162 papers (54.2%) have been applied during continuous integration; which, at first glance, turns out to be the natural space for testing. However, 84 (28.1%) papers have also been identified that have used tests to solve activities in continuous delivery and 44 (14.7%) in continuous deployment, which shows that 42.8% of the tests are outside continuous integration.

According to Figure 4b, in relation to the software development methodology, from the primary studies, it has been determined as "not precise" in 217 (72.6%) papers. In the other cases, it shows 75 (25.1%) papers used agile methodologies, and more explicitly points to TDD and XP with 5 (1.7%) papers, considering both. In particular, in the case of TDD studies, they consider the method important for the success of software testing in DevOps. This suggests that, for now, although TDD is a very good method, there are few studies in this type of context. Similarly, the studies by [43] and [39] consider that TDD would help to better conceptualize testing strategies and mitigate system design errors for help continuous testing.

4.5 RQ-5 What programming languages and software testing tools are used in DevOps?

Due to the nature and objectives of the primary studies, in many cases, programming languages, testing support tools, and version control tools are not required. In the case of programming languages (see Figure 5), it is observed that Java is the most reported language with 90 (30%) papers. In the case of test support tools, Junit with 25 (8.4%) and Selenium with 13 (4.3%) papers are the most reported. Finally, in the case of version control tools, Git is mentioned in 179 (59.9%) of papers.

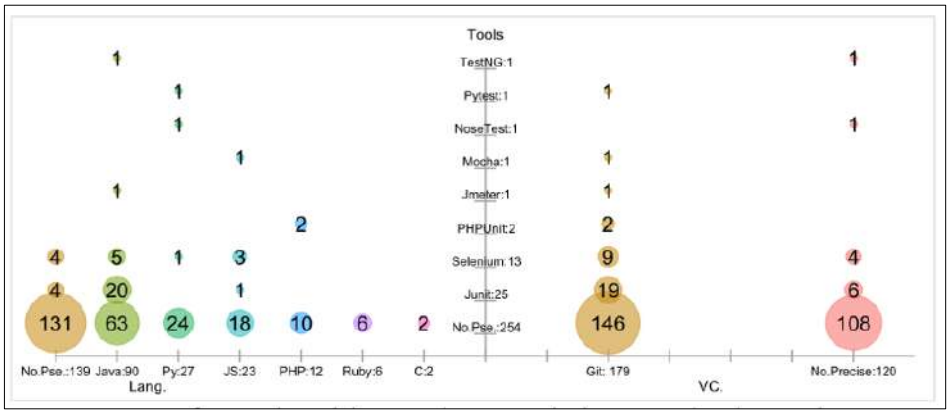

Fig 5. Software testing tools in DevOps by programming languages and version control

In the review of [39], it is agreed that Junit, Selenium and Git are the most frequent tools in the DevOps software testing application. In addition [39], considers NUnit among the most frequent, however, of the selected primary studies, no reference to said tool was found.

According to Figure 6a, Java is the most used language over time with an average of 13 papers per year, while Python has been considered in recent years, with an average of 4 papers per year as presented in Figure 6b.

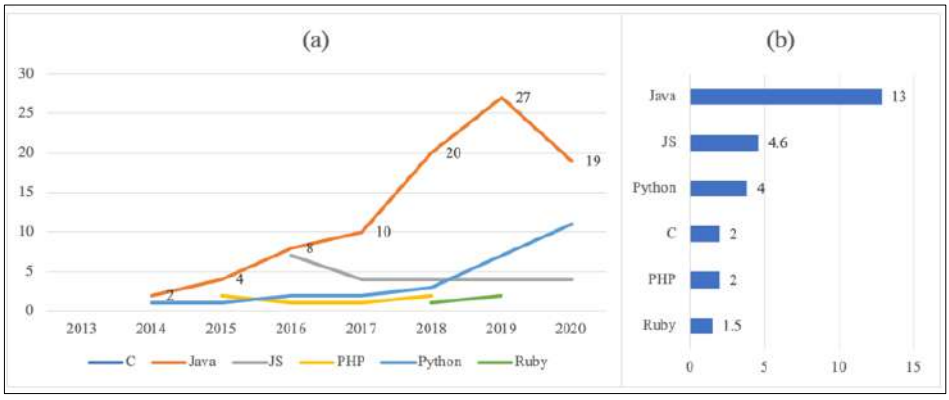

Fig 6. Programming languages in software testing over time (a) and average per year (b)

4.6 RQ-6 In what types of applications and architectures is software testing used in the DevOps context?

In relation to the types of applications where software tests are used in DevOps (see Figure 7a), reported in the primary studies, web applications with 219 (71.9%) papers have to be the most reported applications, and to a lesser extent, mobile applications with 13 (4.3%) papers. The identified console applications are reported for cases in which they apply machine learning concepts and use this type of application to display the results. In relation to the types of architecture (see Figure 7b), the primary studies indicate that 134 (44.8%) are of the MVC type and 52 (17.4%) are of the SOA type, and especially, of the latter, 14 studies report REST as a technology communication. Despite this, 85 (28.4%) papers which represent a high percentage that does not need it.

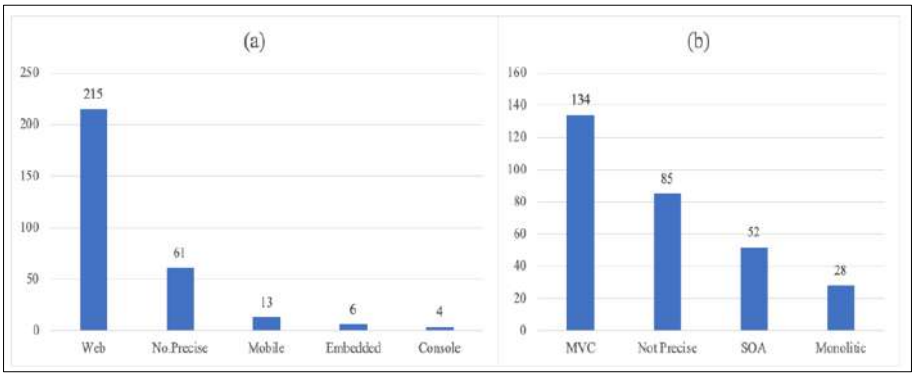

Fig 7. Type of applications (a) and architectures (b) in software testing in DevOps

For [39], 33% of their studies found are web applications, being the most frequent for DevOps software tests; and it also agrees that few researches, that is, 1.6%, are reported on embedded applications.

4.7 RQ-7 What tools are used for software testing in DevOps?

Regarding the tools, it can be pointed out that they are not reported in 111 (37.1%) of the studies (see Figure 8a). In the studies that are reported, Jenkins is present in 92 (30.8%) primary studies. This result coincides with the review by [39] who also found Jenkins to be the most studied tool. In the industry, Jenkins is known as a very versatile tool that allows you to automatically run tests written by the development team, whether they are unit, integration, UI, loading and others. Crossing these results with the years of publication, according to Figure 8b, it can be seen that Jenkins has been increasingly reported in primary studies since 2013. It is also observed, according to Figure 8c, in relation to the average of the publications of papers per year, which Docker has about 6.8 papers/year since 2016, AWS is 3.3 since 2018 and GitLab is 4.8 since 2017. This result shows that Docker is being recurrently reported in the selected primary studies. In the interviews conducted by [42], containerization is mentioned as one of the most studied solutions in continuous delivery.

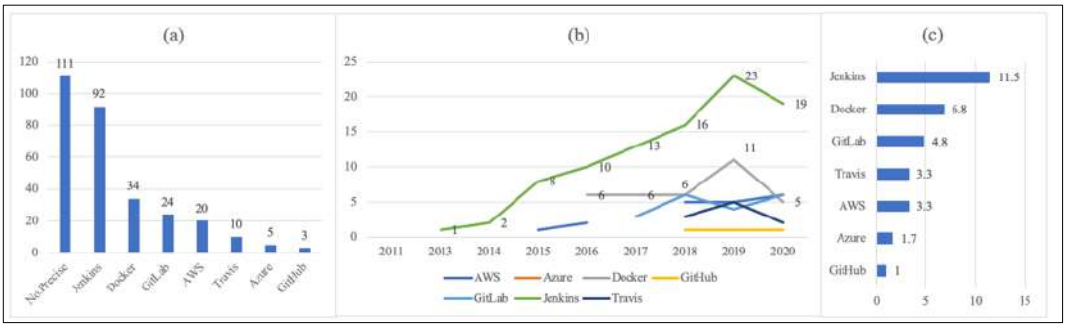

Fig 8. Software testing tools in DevOps (a) by years (b) and, distributed over time and average per year (c)

In Figure 9, it can be seen that Java appears in 40 (13.4%) primary studies, being used in conjunction with Jenkins, becoming the most frequent language for Jenkins. Furthermore, in the case of Java, 19 (21%) papers have been applied in industry and 3 (4%) in the academic context.

Figure 10 shows that 63 (21%) Jenkins primary studies have been studied in the industry and Docker with 34 (7.4%) is behind Jenkins. This shows that Jenkins is the most studied software testing tool in DevOps contexts.

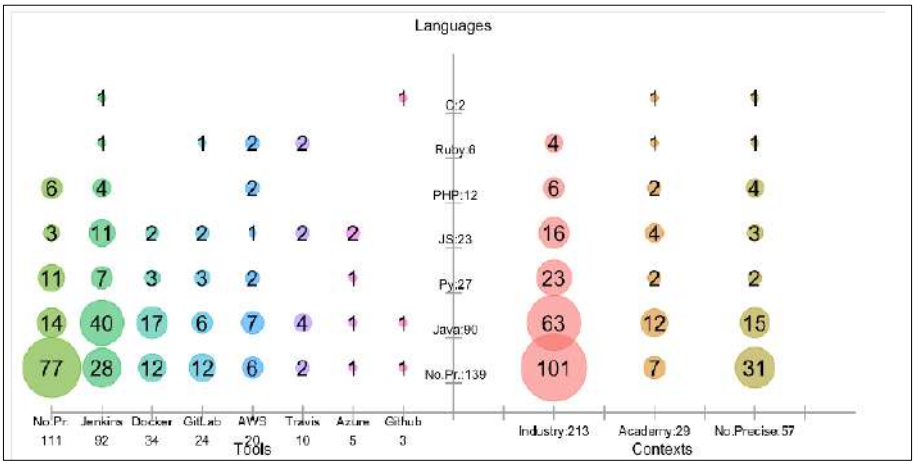

Fig 9. Programming languages and tools in DevOps software testing

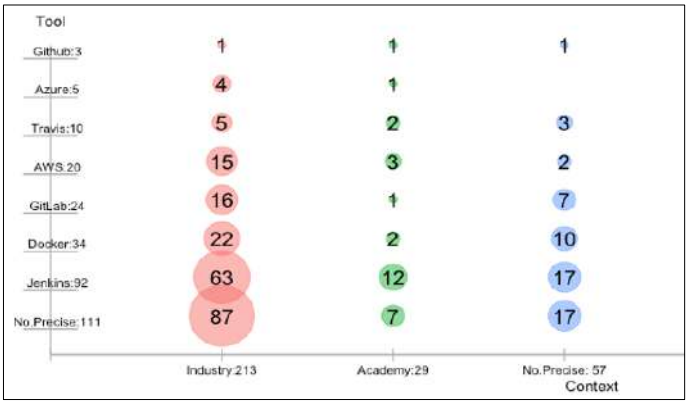

Fig 10. Tools in DevOps for software testing according to its context

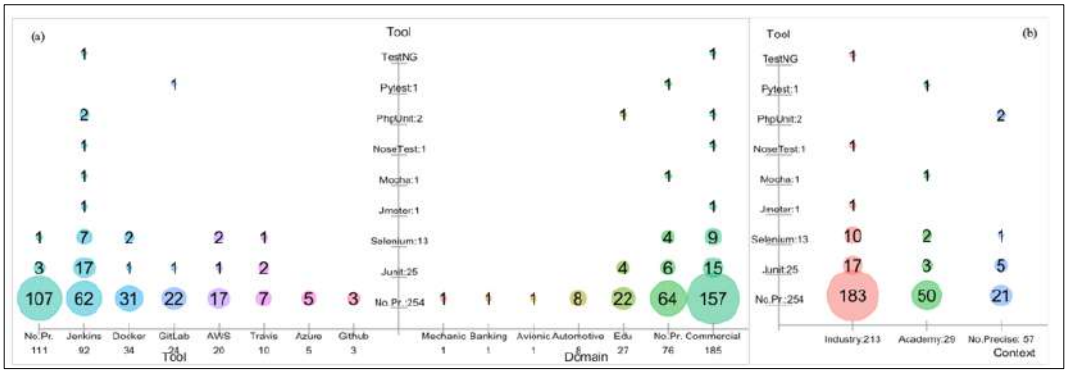

Fig 11. Test tools, infrastructure in DevOps (a) and application context (b)

Figure 11a shows that although Java was often used as a programming language, Junit was not necessarily mentioned in these studies. However, Junit does appear as the most mentioned testing tools in the primary studies. In addition, these, for the most part, 185 (61.8%) papers have been applied in commercial business domains. Figure 11b confirms that Junit is also applied in the industrial context.

4.8 RQ-8 In what types of activities do software testing occur in DevOps? Also, are safety tests mentioned?

According to Figure 12, the selected primary studies show that more than 230 (75%) have concerned themselves with both what is needed in development and in operation, be it with tools, methods, frameworks or suggestions. 60 (20%) papers have studied the specific activities of development teams. Finally, only 9 (3%) have focused solely on operating activities.

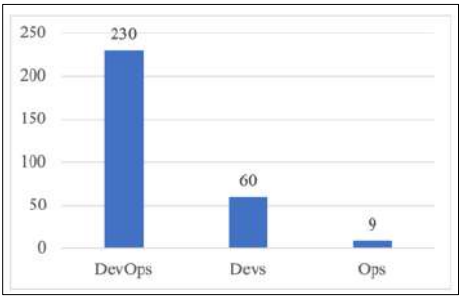

Fig 12. Software testing in DevOps phases

According to Figure 13, more than half of the papers found, that is 169 (56.6%), mention application security as an important factor in the DevOps contexts, despite the fact that there are only 15 application testing papers. penetration (see Figure 4a). These findings are in the same direction as that indicated by [45], [46] and [39], about the need to study more about the security issues in Devops contexts, also known as DevSecOps. This allows you to integrate these types of tests into your development tools.

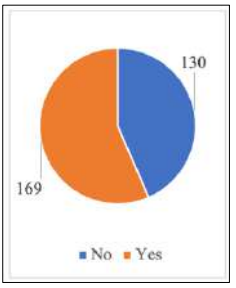

Fig 13. Mention of security in software testing in DevOps

4.9 Threats to Validity

The analysis of the threats to validity was based on the work and questions proposed by [47].

- **Study Selection Validation**. During the planning of the research, in order to ensure the proper identification of all relevant studies, the following was carried out: (i) a preliminary search to identify a relevant set of 20 "test" papers that allowed validating the research questions research, the search chain and selection process; then, (ii) Population and Intervention was used, according to [36], to structure a convenient search chain, actually an iterative task; (iii) a chain test was carried out with the "test" papers, and a check was made if the data obtained from said "test" papers allowed to answer the research questions; and (iv) it was established to work with 6 relevant digital databases.

  The selection was made using the methodology proposed by [36]. Duplicate papers were filtered in the exclusion criteria by DOI, title, authors and year. Inclusion/exclusion criteria were discussed by the authors based on similar research. At each stage, a general criterion was applied, that, when in doubt of acceptance or rejection, acceptance is chosen so that the paper is

subsequently evaluated. This reloads the next stage, but reduces the risk of deleting relevant papers.

- **Data Validation**. Taking into account what was indicated in [36], it was decided to only work with relevant digital databases. These databases usually already have evaluation schemes for the journals and reports of events that they incorporate. In this context, it was decided not to make a quality assessment in the selection process.

  In the first 100 primary studies, a first consolidation was performed, and these studies were discussed between both authors. The evaluation also made it possible to note the relationship of the results with the subject under research. The classification schemes were proposed during the planning of the SMS and were refined, in some cases, during the data extraction. Additionally, the verification of the selection was carried out by the second author in a sample manner.
- **Research Validation**. Both authors are related to the research topic and the second author has more experience in secondary studies. The work carried out is replicable since all the data collected during the research are publicly accessible, phase by phase, as well as the general search string and the personalized ones for each database. At the beginning of the research, it was determined by the research questions and the results of the first stages, that the research would be a systematic mapping of literature due to the need to classify software tests in DevOps contexts. The research can be generalized to all DevOps contexts because it collects the information without considering specific regions, places or periods. In addition, it considers primary studies from both industry and academia.

5. Conclusions

This research presents a Systematic Mapping Study (SMS) on software testing in the DevOps context. The SMS is based on the proposal of [36]. In the selection process, 3,312 studies were obtained and at the end of the process, 299 were selected as primary studies. Based on the data obtained from the primary studies, it was possible to answer the 8 research questions raised.

The interest of research on software testing in the DevOps context is current and continuously growing since 2011. It is also appreciated that it is a global interest, in particular, considering that there are 16 countries from 3 regions (America, Europe and Asia) who have published 239 (80%) of the studies. In accordance with the origin and empirical nature of DevOps, the majority of primary studies, which mean 235 (78.6%) are of the type of case studies and experiments. Likewise, 213 of these studies have been carried out in industry contexts (71.2%) and 185 in commercial applications (61.8%). In addition, 216 (72.2%) primary studies have proposed tools that support test automation. The results also indicate that software testing is considered an important factor in DevOps issues, but what levels of testing are being used are not specified. But, in those that do specify, unit and integration tests are the most studied, and to a lesser extent, user, load and stress and security tests.

In relation to technology, such as programming language and test support tools, it can be noted that these issues are not explicitly reported in primary studies. In the cases that do report, it is pointed out that Java is the most reported language with 90 (30%) both in academic and industrial environments; and in the case of test development tools, 25 papers, that is means, more than 8.3% have been reported to Junit. Other reported programming languages are: Python, Js and PHP respectively. Furthermore, it has to be mentioned that Java is the most reported language in primary studies over time, with an average of 13 papers per year.

The most studied types of applications are those of the Web type with 216 (72.2%), based on both SOA and MVC. One of the most reported tools is Jenkins for both continuous integration, continuous deployment and continuous delivery. In addition, tools such as: Travis, Docker, GitLab, Github and AWS are also reported, showing that the studies carried out are applied to current market tools.

The results of this research show research opportunities in software testing for the DevOps contexts. Likewise, it is clear that training in automated software testing skills could help small companies to compete in the world market with quality.

References / Список литературы

[1] Samarawickrama S.S., Perera I. Continuous scrum: A framework to enhance scrum with DevOps. In Proc. of the 17th International Conference on Advances in ICT for Emerging Regions, 2017, pp. 19-25.

[2] Nicolau de França B.B., Jeronimo H., Travassos G.H. Characterizing DevOps by hearing multiple voices. In Proc. of the XXX Brazilian Symposium on Software Engineering, 2016, pp. 53-62.

[3] Elliot S. DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified. IDC, 2015, 13 p.

[4] Ebert C., Gallardo G. et al. «DevOps», IEEE Software, vol. 33, issue 3, 2016, pp. 94-100.

[5] Riungu-Kalliosaari L, Mäkinen S. et al. DevOps Adoption Benefits and Challenges in Practice: A Case Study. Lecture Notes in Computer Science, vol. 10027, 2016, pp. 590-597.

[6] Stillwell M., Coutinho J.G.F. A DevOps approach to integration of software components in an EU research project. In Proc. of the 1st International Workshop on Quality-Aware DevOps, 2015, pp. 1-6.

[7] Céspedes D., Angeleri P. et al. Software Product Quality in DevOps Contexts: A Systematic Literature Review. Advances in Intelligent Systems and Computing, vol. 1071, Springer, 2020, pp. 51-64.

[8] Perera P., Silva R., Perera I. Improve software quality through practicing DevOps. In Proc. of the 17th International Conference on Advances in ICT for Emerging Regions, 2017, pp. 13-18.

[9] Elberzhager F., Arif T. et al. From Agile Development to DevOps: Going Towards Faster Releases at High Quality – Experiences from an Industrial Context. Lecture Notes in Business Information Processing, vol. 269, 2017, pp. 33-44.

[10] Jones S., Noppen J., Lettice F. Management challenges for devops adoption within UK SMEs. In Proc. of the 2nd International Workshop on Quality-Aware DevOps, 2016, pp. 7-11.

[11] Soni M. End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery. In Proc. of the IEEE International Conference on Cloud Computing in Emerging Markets, 2015, pp. 85-89.

[12] Senapathi M., Buchan J., Osman H. DevOps capabilities, practices, and challenges: Insights from a case study. In Proc. of the International Conference on Evaluation and Assessment in Software Engineering, 2018, pp. 57-67.

[13] Chen L. Continuous Delivery: Overcoming adoption challenges. Journal of Systems and Software, vol. 128, 2017, pp. 72-86.

[14] Valdivia J.A., Lora-González A. et al. Patterns Related to Microservice Architecture: a Multivocal Literature Review. Programming and Computer Software, vol. 46, issue 8, 2020, pp. 594-608 / Вальдивия Х.А., Лора-Гонсалес А и др. Паттерны микросервисной архитектуры: многопрофильный обзор литературы. Труды ИСП РАН, том 33, вып. 1, 2021 г., стр. 81-96. DOI: 10.15514/ISPRAS-2021-33(1)-6.

[15] Debois P. Agile Infrastructure & Operations. In Proc. of the Agile 2008 Conference, 2008, pp. 202-207.

[16] Virmani M. Understanding DevOps & bridging the gap from continuous integration to continuous delivery. In Proc. of the 5th International Conference on Innovative Computing Technology, 2015, pp. 78-82.

[17] Mullaguru S.N. Changing Scenario of Testing Paradigms using DevOps--A Comparative Study with Classical Models. Global Journal of Computer Science and Technology, vol. 15, issue 2, 2015, pp. 23-27.

[18] Chernonozhkin S.K. Automated Test Generation and Static Analysis. Programming and Computer Software, vol. 27, issue 2, 2001, pp. 86-94 / Черноножкин С.К. Задача автоматического построения тестов и статистический анализ. Программирование, том 27, вып. 2, 2001 г., стр. 47-59.

[19] Kuliamin V.V., Petrenko A.K. et al. The UniTesK Approach to Designing Test Suites. Programming and Computer Software, vol. 29, issue 6, 2003, pp. 310-322 / Кулямин В.В., Петренко А.К. и др. Подход UniTesK к разработке тестов. Программирование, том 29, вып. 6, 2003 г., стр. 25-43.

[20] Jabbari R., Ali N., Petersen K. What is DevOps?: A Systematic Mapping Study on Definitions and Practices. In Proce. of the Scientific Workshop of XP2016, 2016, article no. 12, 11 p.

[21] Ghantous G.B., Gill A. DevOps: Concepts, Practices, Tools, Benefits and Challenges. In Proc. of the 21st Pacific Asia Conference on Information Systems (PACIS), 2017, article no. 96, 13 p.

[22] Lwakatare L.E., Kuvaja P., Oivo M. Dimensions of DevOps. Lecture Notes in Business Information Processing, vol. 212, 2015, pp. 212–217.

[23] Katal A., Bajoria V., Dahiya S. DevOps: Bridging the gap between development and operations. In Proc. of the 3rd International Conference on Computing Methodologies and Communication, 2019, pp. 1-7.

[24] Kamuto M.B., Langerman J.J. Factors inhibiting the adoption of DevOps in large organisations: South African context. In Proc. of the 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, 2017, pp. 48-51.

[25] Zimmerer P. Strategy for Continuous Testing in iDevOps. In Proc. of the IEEE/ACM 40th International Conference on Software Engineering, 2018, pp. 532-533.

[26] Fowler M. Continuous Integration. 2006. Available at: https://www.martinfowler.com/articles/continuousIntegration.html, accessed 28-nov-2020.

[27] Parnin C., Helms E. et al. The Top 10 Adages in Continuous Deployment. IEEE Software, vol. 34, issue 3, 2017, pp. 86-95.

[28] Fowler M. Continuous Delivery. 30-may-2013. Available at:: https://martinfowler.com/bliki/ContinuousDelivery.html, accessed 28-nov-2020.

[29] Fitzgerald B., Stol K.J. Continuous software engineering and beyond: Trends and challenge. In Proc. of the 1st International Workshop on Rapid Continuous Software Engineering, 2014, pp. 1-9.

[30] ISO/IEC/IEEE, «ISO/IEC/IEEE 24765:2017 Systems and software engineering – Vocabulary. Geneva, 2017.

[31] Guide to the Software Engineering Body of Knowledge (SWEBOK), Version 3.0. IEEE Computer Society, 2014, 339 p.

[32] Gupta R.K., Manikreddy P., Gv A. Challenges in adapting agile testing in a legacy product. In Proc. of the 11th IEEE International Conference on Global Software Engineering, 2016, pp. 104-108.

[33] Jeeva Padmini K.V., Kankanamge P.S. et al. Challenges faced by agile testers: A case study. In Proc. of the Moratuwa Engineering Research Conference, 2018, pp. 431-436.

[34] Coutinho J.C.S., Andrade W.L., Machado P.D.L. Requirements engineering and software testing in agile methodologies: A systematic mapping. In Proc. of the XXXIII Brazilian Symposium on Software Engineering, 2019, pp. 322-331.

[35] Cruzes D.S., Moe N.B., Dyba T. Communication between developers and testers in distributed continuous agile testing. In Proc. of the 11th IEEE International Conference on Global Software Engineering, 2016, pp. 59-68.

[36] Petersen K., Vakkalanka S., Kuzniarz L. Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology, vol. 64, 2015, pp. 1-18.

[37] Kuhrmann M., Diebold P., Münch J. Software process improvement: A systematic mapping study on the state of the art. PeerJ Computer Science, issue 5, 2016, article no. 62, 38 p.

[38] Pinto G., Castor F. et al. Work practices and challenges in continuous integration: A survey with Travis CI users. Software: Practice and Experience, vol. 48, issue 12, 2018, pp. 2223-2236.

[39] Shahin M., Ali Babar M., Zhu L. Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. IEEE Access, vol. 5, 2017, pp. 3909-3943.

[40] Alnafessah A., Gias A.U. et al. Quality-Aware DevOps Research: Where Do We Stand. IEEE Access, vol. 9, 2021, pp. 44476-44489.

[41] Mascheroni M.A., Irrazábal E. Continuous Testing and Solutions for Testing Problems in Continuous Delivery: A Systematic Literature Review. Computación y Sistemas, vol. 22, issue 3, 2018, pp. 1009-1038.

[42] Shahin M., Babar M.A. et al. Beyond Continuous Delivery: An Empirical Investigation of Continuous Deployment Challenges. In Proc. of the International Symposium on Empirical Software Engineering and Measurement, 2017, pp. 111-120.

[43] Laukkanen E., Itkonen J., Lassenius C. Problems, causes and solutions when adopting continuous delivery – A systematic literature review. Information and Software Technology, vol. 82, 2017, pp. 55-79.

[44] Sane P. A Brief Survey of Current Software Engineering Practices in Continuous Integration and Automated Accessibility Testing. In Proc. of the International Conference on Wireless Communications, Signal Processing and Networking, 2021, pp. 130-134.

[45] Rajapakse R.N., Zahedi M. et al. Challenges and solutions when adopting DevSecOps: A systematic review. Information and Software Technology, vol. 141, 2021, article no. 106700, 27 p.

[46] Daoudagh S., Lonetti F., Marchetti E. Continuous Development and Testing of Access and Usage Control. In Proc. of the European Symposium on Software Engineering, 2020, pp. 51-59.

[47] Ampatzoglou A., Bibi S. et al. Identifying, categorizing and mitigating threats to validity in software engineering secondary studies», Information and Software Technology, vol. 106, 2019, pp. 201-230.

Appendix A. List of Primary Studies

Table A. Primary Studies

ID	Authors	Year	Title
S01	K. Priyadarsini and E. Fantin Irudaya Raj and A. Yasmine Begum and V. Shanmugasundaram	2020	Comparing DevOps procedures from the context of a systems engineer
S02	Casola V., De Benedictis A., Rak M., Salzillo G.	2020	A cloud secdevops methodology: From design to testing
S03	Fehlmann T., Kranich E.	2020	A Framework for Automated Testing
S04	Amaral C.J., Kampik T., Cranefield S.	2020	A framework for collaborative and interactive agent-oriented developer operations
S05	C. Klammer; J. Gmeiner	2020	A Lightweight Customized Build Chain Visualization Approach Applied in Industry
S06	Casola V., De Benedictis A., Rak M., Villano U.	2020	A methodology for automated penetration testing of cloud applications
S07	R. Guntha; S. N. Rao; H. Muccini; M. Vinodini Ramesh	2020	A Novel Paradigm for Rapid Yet Robust Continuous Delivery of Software for Disaster Management Scenarios
S08	Hsu W., Lin J.-S., Chen Y.-C., Wang C.-Y., Huang C.-T.	2020	An Automatic Software Quality and Function Assurance Case Study for Agile
S09	Cai Y.X., Shang Y.F., Tan Y.X., Tang Z.W., Zhao B.	2020	An Effective Solution for Application Orchestration
S10	R. W. Macarthy; J. M. Bass	2020	An Empirical Taxonomy of DevOps in Practice
S11	A. Kanchana; C. Murthy B.N.	2020	Automated Development and Testing of ECUs in Automotive Industry with Jenkins
S12	Avritzer A.	2020	Automated scalability assessment in devops environments
S13	Rakshith M.N., Shivaprasad N.	2020	Build Optimization Using Jenkins
S14	Karlaš B., Interlandi M., Renggli C., Wu W., Zhang C., Mukunthu Iyappan Babu D., Edwards J., Lauren C., Xu A., Weimer M.	2020	Building Continuous Integration Services for Machine Learning
S15	G. Ambrosino; G. B. Fioccola; R. Canonico; G. Ventre	2020	Container Mapping and its Impact on Performance in Containerized Cloud Environments
S16	S. H. Reiterer; S. Balci; D. Fu; M. Benedikt; A. Soppa; H. Szczerbicka	2020	Continuous Integration for Vehicle Simulations
S17	L. Gota; D. Gota; L. Miclea	2020	Continuous Integration in Automation Testing
S18	Gorsky S.A.	2020	Continuous integration, delivery, and deployment for scientific workflows in Orlando Tools
S19	T. Rangnau; R. v. Buijtenen; F. Fransen; F. Turkmen	2020	Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines
S20	M. Johnson; D. Cummings; B. Leinwand; C. Elsberry	2020	Continuous Testing and Deployment for Urban Air Mobility
S21	Angara J., Prasad S.	2020	Continuous testing real-time health analytics dashboard
S22	Doležel M.	2020	Defining testops: Collaborative behaviors and technology-driven workflows seen as enablers of effective software testing in devops
S23	Török M., Pataki N.	2020	DevOps dashboard with heatmap
S24	Yang D., Wang D., Yang D., Dong Q., Wang Y., Zhou H., Daocheng H.	2020	DevOps in practice for education management information system at ECNU
S25	Laaber C., Würsten S., Gall H.C., Leitner P.	2020	Dynamically reconfiguring software microbenchmarks: Reducing execution time without sacrificing result quality
S26	Al-Sabbagh K.W., Staron M., Ochodek M., Meding W.	2020	Early prediction of test case verdict with bag-of-words vs. word embeddings
S27	Couto L.D., Tran-Jørgensen P.W.V., Nilsson R.S., Larsen P.G.	2020	Enabling continuous integration in a formal methods setting

S28	Karakasis V., Manitaras T., Rusu V.H., Sarmiento-Pérez R., Bignamini C., Kraushaar M., Jocksch A., Omlin S., Peretti-Pezzi G., Augusto J.P.S.C., Friesen B., He Y., Gerhardt L., Cook B., You Z.-Q., Khuvis S., Tomko K.	2020	Enabling Continuous Testing of HPC Systems Using ReFrame
S29	Vassallo C., Proksch S., Zemp T., Gall H.C.	2020	Every build you break: developer-oriented assistance for build failure resolution
S30	Luzar A., Stanovnik S., Cankar M.	2020	Examination and comparison of tosca orchestration tools
S31	Meinicke J., Wong C.-P., Vasilescu B., Kästner C.	2020	Exploring differences and commonalities between feature flags and configuration options
S32	Demeyer S., Parsai A., Vercammen S., van Bladel B., Abdi M.	2020	Formal Verification of Developer Tests: A Research Agenda Inspired by Mutation Testing
S33	M. Mazkatli; D. Monschein; J. Grohmann; A. Koziolek	2020	Incremental Calibration of Architectural Performance Models with Parametric Dependencies
S34	Shin J.-S., Kim J.	2020	K-one playground: Reconfigurable clusters for a cloud-native testbed
S35	P. Batra; A. Jatain	2020	Measurement Based Performance Evaluation of DevOps
S36	Eismann S., Bezemer C.-P., Shang W., Okanović D., Van Hoorn A.	2020	Microservices: A performance tester's dream or nightmare?
S37	van den Heuvel W.-J., Tamburri D.A.	2020	Model-driven ml-ops for intelligent enterprise applications: vision, approaches and challenges
S38	Shahin M., Babar M.A.	2020	On the role of software architecture in DevOps transformation: An industrial case study
S39	Mirhosseini S., Parnin C.	2020	Opunit: Sanity Checks for Computing Environments
S40	Gias A.U., Van Hoorn A., Zhu L., Casale G., Düllmann T.F., Wurster M.	2020	Performance engineering for microservices and serverless applications: The RADON approach
S41	J. Chen	2020	Performance Regression Detection in DevOps
S42	Raj P., Sinha P.	2020	Project management in era of agile and devops methodolgies
S43	Cheriyan A., Gondkar R.R., Babu S.S.	2020	Quality Assurance Practices and Techniques Used by QA Professional in Continuous Delivery
S44	M. Huang; W. Fan; W. Huang; Y. Cheng; H. Xiao	2020	Research on Building Exploitable Vulnerability Database for Cloud-Native App
S45	C. Fayollas; H. Bonnin; O. Flebus	2020	SafeOps: A Concept of Continuous Safety
S46	Vishnu Vardhan Reddy B.S., Swamy B.K., Sai S.P.S., Kiran K.V.D.	2020	Securing web application by using qualitative research methods for detection of vulnerabilities in any application of DevSecOps
S47	Petrovic N., Tosic M.	2020	SMADA-Fog: Semantic model driven approach to deployment and adaptivity in fog computing
S48	Orviz Fernández P., David M., Duma D.C., Ronchieri E., Gomes J., Salomoni D.	2020	Software Quality Assurance in INDIGO-DataCloud Project: a Converging Evolution of Software Engineering Practices to Support European Research e-Infrastructures
S49	Wang Y., Mäntylä M.V., Demeyer S., Wiklund K., Eldh S., Kairi T.	2020	Software test automation maturity: A survey of the state of the practice
S50	E. Bernard; F. Ambert; B. Legeard	2020	Supporting efficient test automation using lightweight MBT
S51	R. Li; X. Liu; X. Zheng; C. Zhang; H. Liu	2020	TDD4Fog: A Test-Driven Software Development Platform for Fog Computing Systems
S52	Y. Wang; M. Pyhäjärvi; M. V. Mäntylä	2020	Test Automation Process Improvement in a DevOps Team: Experience Report
S53	Hasan M.M., Bhuiyan F.A., Rahman A.	2020	Testing practices for infrastructure as code
S54	Marlowe T.J., Kirova V., Chang G.	2020	The state of agile: Changes in the world of change
S55	Klemets J., Storholmen T.C.B.	2020	Towards Super User-Centred Continuous Delivery: A Case Study
S56	Ding Z., Chen J., Shang W.	2020	Towards the use of the readily available tests from the release pipeline as performance tests. Are we there yet

S57	Leotta M., Cerioli M., Olianas D., Ricca F.	2020	Two experiments for evaluating the impact of Hamcrest and AssertJ on assertion development
S58	K. Gallaba; S. McIntosh	2020	Use and Misuse of Continuous Integration Features: An Empirical Study of Projects That (Mis)Use Travis CI
S59	Y. Zhou; Y. Su; T. Chen; Z. Huang; H. C. Gall; S. Panichella	2020	User Review-Based Change File
S60	Yu L., Alégroth E., Chatzipetrou P., Gorschek T.	2020	Utilising CI environment for efficient and effective testing of NFRs
S61	Van Rossem S., Tavernier W., Colle D., Pickavet M., Demeester P.	2020	VNF Performance modelling: From stand-alone to chained topologies
S62	Bertolino, Antonia and Angelis, Guglielmo De and Guerriero, Antonio and Miranda, Breno and Pietrantuono, Roberto and Russo, Stefano	2019	DevOpRET: Continuous reliability testing in DevOps
S63	Jacobsen, Douglas M. and Kleinman, Randy and Longley, Harold	2019	Managing a Cray supercomputer as a git branch
S64	B. Meyers; K. Gadeyne; B. Oakes; M. Bernaerts; H. Vangheluwe; J. Denil	2019	A Model-Driven Engineering Framework to Support the Functional Safety Process
S65	F. Zampetti; G. Bavota; G. Canfora; M. D. Penta	2019	A Study on the Interplay between Pull Request Review and Continuous Integration Builds
S66	D. Chhillar; K. Sharma	2019	ACT Testbot and 4S Quality Metrics in XAAS Framework
S67	M. K. A. Abbass; R. I. E. Osman; A. M. H. Mohammed; M. W. A. Alshaikh	2019	Adopting Continuous Integeration and Continuous Delivery for Small Teams
S68	M. Guerriero; M. Garriga; D. A. Tamburri; F. Palomba	2019	Adoption, Support, and Challenges of Infrastructure-as-Code: Insights from Industry
S69	T. Durieux; R. Abreu; M. Monperrus; T. F. Bissyandé; L. Cruz	2019	An Analysis of 35+ Million Jobs of Travis CI
S70	T. Vasile; S. Cane; C. Bertram; F. Jakob	2019	Applying Security Concepts to Continuous Integration for the Purpose of Testing Embedded Systems
S71	C. Vassallo; S. Proksch; H. C. Gall; M. Di Penta	2019	Automated Reporting of Anti-Patterns and Decay in Continuous Integration
S72	A. Janes; B. Russo	2019	Automatic Performance Monitoring and Regression Testing During the Transition from Monolith to Microservices
S73	Krym T., Poniszewska-Marańda A., Markl E., Dupas R.	2019	Automatic Process of Continuous Integration of Web Application
S74	Najafi A., Rigby P.C., Shang W.	2019	Bisecting commits and modeling commit risk during testing
S75	D. A. Tomassi; N. Dmeiri; Y. Wang; A. Bhowmick; Y. Liu; P. T. Devanbu; B. Vasilescu; C. Rubio-González	2019	BugSwarm: Mining and Continuously Growing a Dataset of Reproducible Failures and Fixes
S76	Satyal S., Weber I., Paik H.-Y., Di Ciccio C., Mendling J.	2019	Business process improvement with the AB-BPM methodology
S77	R. K. Gupta; M. Venkatachalapathy; F. K. Jeberla	2019	Challenges in Adopting Continuous Delivery and DevOps in a Globally Distributed Product Team: A Case Study of a Healthcare Organization
S78	Judvaitis J., Nesenbergs K., Balass R., Greitans M.	2019	Challenges of DevOps ready IoT testbed
S79	Nogueira A.F., Sergeant E., Ribeiro J.C.B., Zenha-Rela M.A., Craske A.	2019	Collecting data from continuous practices: An infrastructure to support team development
S80	C. Singh; N. S. Gaba; M. Kaur; B. Kaur	2019	Comparison of Different CI/CD Tools Integrated with Cloud Platform
S81	I. M. A. Jawarneh; P. Bellavista; F. Bosi; L. Foschini; G. Martuscelli; R. Montanari; A. Palopoli	2019	Container Orchestration Engines: A Thorough Functional and Performance Comparison

S82	M. Grambow; F. Lehmann; D. Bermbach	2019	Continuous Benchmarking: Using System Benchmarking in Build Pipelines
S83	Glein R., Perloff A., Ulmer K.	2019	Continuous integration of FPGA designs for CMS
S84	W. Felidré; L. Furtado; D. A. da Costa; B. Cartaxo; G. Pinto	2019	Continuous Integration Theater
S85	Johanssen, JO; Kleebaum, A; Paech, B; Bruegge, B	2019	Continuous software engineering and its support by usage and decision knowledge: An interview study with practitioners
S86	L. G. Gușeilă; D. Bratu; S. Moraru	2019	Continuous Testing in the Development of IoT Applications
S87	Lescisin M., Mahmoud Q.H., Cioraca A.	2019	Design and implementation of SFCI: A tool for security focused continuous integration
S88	O. Veres; N. Kunanets; V. Pasichnyk; N. Veretennikova; R. Korz; A. Leheza	2019	Development and Operations - the Modern Paradigm of the Work of IT Project Teams
S89	R. A. K. Jennings; G. Gannod	2019	DevOps - Preparing Students for Professional Practice
S90	C. Heistand; J. Thomas; N. Tzeng; A. R. Badger; L. M. Rodriguez; A. Dalton; J. Pai; A. Bodzas; D. Thompson	2019	DevOps for Spacecraft Flight Software
S91	L. Georgeta Gușeilă; D. -V. Bratu; S. -A. Moraru	2019	DevOps Transformation for Multi-Cloud IoT Applications
S92	P. Agrawal; N. Rawat	2019	Devops, A New Approach To Cloud Development & Testing
S93	Embury S.M., Page C.	2019	Effect of continuous integration on build health in undergraduate team projects
S94	C. Vassallo	2019	Enabling Continuous Improvement of a Continuous Integration Process
S95	K. Baral; R. Mohod; J. Flamm; S. Goldrich; P. Ammann	2019	Evaluating a Test Automation Decision Support Tool
S96	H. Huijgens; E. Greuter; J. Brons; E. A. van Doorn; I. Papadopoulos; F.	2019	Factors Affecting Cloud Infra-Service Development Lead Times: A Case Study at ING
	Morales Martinez; M. Aniche; O. Visser; A. van Deursen		
S97	T. Suk; J. Hwang; M. F. Bulut; Z. Zeng	2019	Failure-Aware Application Placement Modeling and Optimization in High Turnover DevOps Environment
S98	Bezemer C.-P., Eismann S., Ferme V., Grohmann J., Heinrich R., Jamshidi P., Shang W., Van Hoorn A., Villavicencio M., Walter J., Willnecker F.	2019	How is performance addressed in DevOps? A survey on industrial practices
S99	B. Chen	2019	Improving the Software Logging Practices in DevOps
S100	S. Carturan; D. Goya	2019	Major Challenges of Systems-of-Systems with Cloud and DevOps ‚Äì A Financial Experience Report
S101	J. A. Shah; D. Dubaria	2019	NetDevOps: A New Era Towards Networking DevOps
S102	J. Haavisto; M. Arif; L. Lov√©n; T. Lepp√§nen; J. Riekki	2019	Open-source RANs in Practice: an Over-The-Air Deployment for 5G MEC
S103	Keahey K., Anderson J., Ruth P., Colleran J., Hammock C., Stubbs J., Zhen Z.	2019	Operational lessons from chameleon
S104	K. Hakimzadeh; J. Dowling	2019	Ops-Scale: Scalable and Elastic Cloud Operations by a Functional Abstraction and Feedback Loops
S105	E. Salinas	2019	Pat Helland on Failure and Resilience in Distributed Systems
S106	Al-Sabbagh K.W., Staron M., Hebig R., Meding W.	2019	Predicting test case verdicts using textual analysis of committed code churns
S107	A. Nuriddinov; W. Tavernier; D. Colle; M. Pickavet; M. Peustery; S. Schneidery	2019	Reproducible Functional Tests for Multi-scale Network Services
S108	Wiedemann A., Forsgren N., Wiesche M., Gewald H., Krcmar H.	2019	Research for practice: The Devops phenomenon

S109	P. K. Sidhu; G. Mussbacher; S. McIntosh	2019	Reuse (or Lack Thereof) in Travis CI Specifications: An Empirical Study of CI Phases and Commands
S110	Mäkinen S., Puonti M., Lehtonen T., Mikkonen T., Kilamo T., Männistö T.	2019	Revisiting continuous deployment maturity: A two-year perspective
S111	Siewruk G., Mazurczyk W., Karpiński A.	2019	Security assurance in Devops methodologies and related environments
S112	Vera-Rivera F.H., Vera-Rivera J.L., Gaona-Cuevas C.M.	2019	Sinplafut: A microservices - Based application for soccer training
S113	S. M. Naik; M. Fernandes; G. Pushpakumar; R. Pathak	2019	Smart Grid Communication Protocol Test Automation along with Protection Test Automation
S114	Risdianto A.C., Usman M., Kim J.W.	2019	SmartX box: Virtualized hyper-converged resources for building an affordable playground
S115	K. Czarnecki	2019	Software Engineering for Automated Vehicles: Addressing the Needs of Cars That Run on Software and Data
S116	Keskin Kaynak İ., Çilden E., Aydin S.	2019	Software Quality Improvement Practices in Continuous Integration
S117	Cunningham S., Gambo J., Lawless A., Moore D., Yilmaz M., Clarke P.M., O'Connor R.V.	2019	Software Testing: A Changing Career
S118	Rahman A., Williams L.	2019	Source code properties of defective infrastructure as code scripts
S119	Kapoor S., Sagar K., Reddy B.V.R.	2019	Speedroid: A novel automation testing tool for mobile apps
S120	Arulkumar V., Lathamanju R.	2019	Start to Finish Automation Achieve on Cloud with Build Channel: By DevOps Method
S121	Figalist I., Biesdorf A., Brand C., Feld S., Kiermeier M.	2019	Supporting the DevOps Feedback Loop using Unsupervised Machine Learning
S122	G. Lim; M. Ham; J. Moon; W. Song; S. Woo; S. Oh	2019	TAOS-CI: Lightweight Modular Continuous Integration System for Edge Computing
S123	Cruzes D.S., Melsnes K., Marczak S.	2019	Testing in a DevOps Era: Perceptions of Testers in Norwegian Organisations
S124	D. Martin; S. Panichella	2019	The Cloudification Perspectives of Search-Based Software Testing
S125	Fazayeli H., Syed-Mohamad S.M., Md Akhir N.S.	2019	Towards auto-labelling issue reports for pull-based software development using text mining approach
S126	Meixner K., Winkler D., Biffl S.	2019	Towards combined process & tool variability management in software testing
S127	R. Pietrantuono; A. Bertolino; G. De Angelis; B. Miranda; S. Russo	2019	Towards Continuous Software Reliability Testing in DevOps
S128	F. Giorgi; F. Paulisch	2019	Transition Towards Continuous Delivery in the Healthcare Domain
S129	C. Paule; T. F. Düllmann; A. Van Hoorn	2019	Vulnerabilities in Continuous Delivery Pipelines? A Case Study
S130	M. Chwalisz; K. Geissdoerfer; A. Wolisz	2019	Walker: DevOps Inspired Workflow for Experimentation
S131	B. Benni; M. Blay-Fornarino; S. Mosser; F. Précisio; G. Jungbluth	2019	When DevOps Meets Meta-Learning: A Portfolio to Rule them all
S132	Daoudagh, Said and Lonetti, Francesca and Marchetti, Eda	2019	An automated framework for continuous development and testing of access control systems
S133	Luz, Welder Pinheiro and Pinto, Gustavo and Bonifácio, Rodrigo	2018	Building a Collaborative Culture: A Grounded Theory of Well Succeeded Devops Adoption in Practice
S134	Osses, Felipe and Márquez, Gastón and Astudillo, Hernán	2018	Exploration of Academic and Industrial Evidence about Architectural Tactics and Patterns in Microservices
S135	Schulz, Henning and Angerstein, Tobias and van Hoorn, André	2018	Towards Automating Representative Load Testing in Continuous Software Engineering
S136	K. Kuusinen; V. Balakumar; S. C. Jepsen; S. H. Larsen; T. A. Lemqvist; A. Muric; A. Ø. Nielsen; O. Vestergaard	2018	A Large Agile Organization on Its Journey Towards DevOps

S137	Sandobalin J.	2018	A Model-Driven Approach to Continuous Delivery of Cloud Resources
S138	Sandobalin J., Insfran E., Abrahao S.	2018	A smart provisioning approach to cloud infrastructure
S139	Baudry B., Harrand N., Schulte E., Timperley C., Tan S.H., Selakovic M., Ugherughe E.	2018	A spoonful of DevOps helps the GI go down
S140	J. Shah; D. Dubaria; J. Widhalm	2018	A Survey of DevOps tools for Networking
S141	H. Li; T. Chen; A. E. Hassan; M. Nasser; P. Flora	2018	Adopting Autonomic Computing Capabilities in Existing Large-Scale Systems
S142	Zykov S.V.	2018	Agile services
S143	Akman S., Aksuyek E.B., Kaynak O.	2018	ALM Tool Infrastructure with a Focus on DevOps Culture
S144	Wiedemann A., Wiesche M.	2018	Are you ready for Devops? Required skill set for Devops teams
S145	I. Rubasinghe; D. Meedeniya; I. Perera	2018	Automated Inter-artefact Traceability Establishment for DevOps Practice
S146	R. V. Rosa; C. E. Rothenberg	2018	Automated VNF Testing with Gym: A Benchmarking Use Case
S147	V. Debroy; L. Brimble; M. Yost; A. Erry	2018	Automating Web Application Testing from the Ground Up: Experiences and Lessons Learned in an Industrial Setting
S148	M. J. Kargar; A. Hanifizade	2018	Automation of regression test in microservice architecture
S149	V. Mohan; L. ben Othmane; A. Kres	2018	BP: Security Concerns and Best Practices for Automation of Software Deployment Processes: An Industrial Case Study
S150	A. Rahman; L. Williams	2018	Characterizing Defective Configuration Scripts Used for Continuous Deployment
S151	Rahman A., Agrawal A., Krishna R., Sobran A.	2018	Characterizing the influence of continuous integration: Empirical results from 250+ open source and proprietary projects
S152	A. Agarwal; S. Gupta; T. Choudhury	2018	Continuous and Integrated Software Development using DevOps
S153	X. Bai; M. Li; D. Pei; S. Li; D. Ye	2018	Continuous Delivery of Personalized Assessment and Feedback in Agile Software Engineering Projects
S154	S. A. I. B. S. Arachchi; I. Perera	2018	Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management
S155	L. Williams	2018	Continuously Integrating Security
S156	Alshahwan N., Gao X., Harman M., Jia Y., Mao K., Mols A., Tei T., Zorin I.	2018	Deploying search based software engineering with sapienz at facebook
S157	Marijan D., Sen S.	2018	Devops enhancement with continuous test optimization
S158	D. Marijan; M. Liaaen; S. Sen	2018	DevOps Improvements for Reduced Cycle Times with Integrated Test Optimizations for Continuous Integration
S159	Angara J., Gutta S., Prasad S.	2018	DevOps with continuous testing architecture and its metrics model
S160	Park S., Huh J.-H.	2018	Effect of cooperation on manufacturing IT project development an for successful industry 4.0 Project: Safety management for security
S161	Mårtensson T., Ståhl D., Bosch J.	2018	Enable more frequent integration of software in industry projects
S162	Casale G., Li C.	2018	Enhancing Big Data Application Design with the DICE Framework
S163	T. F. Düllmann; C. Paule; A. van Hoorn	2018	Exploiting DevOps Practices for Dependable and Secure Continuous Delivery Pipelines
S164	Loseva E., Obeid A., Richter H., Backes R., Eichhorn D.	2018	FIXIT - A semi-automatic software deployment tool for arbitrary targets
S165	Jiang H., Chen X., He T., Chen Z., Li X.	2018	Fuzzy clustering of crowdsourced test reports for apps
S166	D. Widder; B. Vasilescu; M. Hilton; C. Kästner	2018	I'm Leaving You, Travis: A Continuous Integration Breakup Story
S167	Fernandes, TCM; Costa, I; Salvetti, N; de Magalhaes, FLF; Fernandes, AA	2018	Influence of DevOps practices in IT management processes according to the COBIT 5 model

S168	Soenen T., van Rossem S., Tavernier W., Vicens F., Valocchi D., Trakadas P., Karkazis P., Xilouris G., Eardley P., Kolometsos S., Kourtis M.-A., Guija D., Siddiqui S., Hasselmeyer P., Bonnet J., Lopez D.	2018	Insights from SONATA: Implementing and integrating a microservice-based NFV service platform with a DevOps methodology
S169	S. van Rossem; W. Tavernier; D. Colle; M. Pickavet; P. Demeester	2018	Introducing Development Features for Virtualized Network Services
S170	Asha N., Mani P.	2018	Knowledge-based acceptance test driven agile approach for quality software development
S171	F. L. Eickhoff; M. L. McGrath; C. Mayer; A. Bieswanger; P. A. Wojciak	2018	Large-scale application of IBM Design Thinking and Agile development for IBM z14
S172	Staron M., Meding W., Söder O., Bäck M.	2018	Measurement and Impact Factors of Speed of Reviews and Integration in Continuous Software Engineering
S173	L. Chen	2018	Microservices: Architecting for Continuous Delivery and DevOps
S174	H. Alipour; Y. Liu	2018	Model Driven Deployment of Auto-Scaling Services on Multiple Clouds
S175	M. Wurster; U. Breitenbücher; O. Kopp; F. Leymann	2018	Modeling and Automated Execution of Application Deployment Tests
S176	D'Ambrogio A., Falcone A., Garro A., Giglio A.	2018	On the importance of simulation in enabling continuous delivery and evaluating deployment pipeline performance
S177	Zhang Y., Vasilescu B., Wang H., Filkov V.	2018	One size does not fit all: An empirical study of containerized continuous deployment workflows
S178	A. Cheriyan; R. R. Gondkar; T. Gopal; S. B. S.	2018	Quality Assurance Practices in Continuous Delivery - an implementation in Big Data Domain
S179	R. Mijumbi; K. Okumoto; A. Asthana; J. Meekel	2018	Recent Advances in Software Reliability Assurance
S180	Kerzazi N., EL Asri I.	2018	Release engineering: From structural to functional view
S181	G. Marquez; F. Osses; H. Astudillo	2018	Review of Architectural Patterns and Tactics for Microservices in Academic and Industrial Literature
S182	Satyal S., Weber I., Paik H.-Y., Di Ciccio C., Mendling J.	2018	Shadow Testing for Business Process Improvement
S183	Limoncelli T.A.	2018	SQL is no excuse to avoid DevOps
S184	P. Zimmerer	2018	Strategy for Continuous Testing in iDevOps
S185	K. K. Luhana; C. Schindler; W. Slany	2018	Streamlining mobile app deployment with Jenkins and Fastlane in the case of Catrobat's pocket code
S186	X. Bai; D. Pei; M. Li; S. Li	2018	The DevOps Lab Platform for Managing Diversified Projects in Educating Agile Software Engineering
S187	Guamán D., Pérez J., Díaz J.	2018	Towards a (semi)-automatic reference process to support the reverse engineering and reconstruction of software architectures
S188	K. Martin; U. Ömer; M. Florian	2018	Towards a Continuous Feedback Loop for Service-Oriented Environments
S189	Steffens A., Lichter H., Moscher M.	2018	Towards data-driven continuous compliance testing
S190	F. Klinaku; V. Ferme	2018	Towards Generating Elastic Microservices: A Declarative Specification for Consistent Elasticity Configurations
S191	M. Peuster; H. Karl	2018	Understand Your Chains and Keep Your Deadlines: Introducing Timeconstrained Profiling for NFV
S192	Kim C., Kim S., Kim J.	2018	Understanding automated continuous integration for containerized smart energy IoT-cloud service
S193	B. Snyder; B. Curtis	2018	Using Analytics to Guide Improvement during an Agile‚ÄìDevOps Transformation
S194	Schermann G., Cito J., Leitner P., Zdun U., Gall H.C.	2018	We're doing it live: A multi-method empirical study on continuous experimentation
S195	Pinto G., Castor F., Bonifacio R., Rebouças M.	2018	Work practices and challenges in continuous integration: A survey with Travis CI users

S196	Fabian Fagerholm and Alejandro {Sanchez Guinea} and Hanna Mäenpää and Jürgen Münch	2017	The RIGHT model for Continuous Experimentation
S197	Ferme, Vincenzo and Pautasso, Cesare	2017	Towards Holistic Continuous Software Performance Assessment
S198	B. P. Eddy; N. Wilde; N. A. Cooper; B. Mishra; V. S. Gamboa; K. M. Shah; A. M. Deleon; N. A. Shields	2017	A Pilot Study on Introducing Continuous Integration and Delivery into Undergraduate Software Engineering Courses
S199	C. Vassallo; G. Schermann; F. Zampetti; D. Romano; P. Leitner; A. Zaidman; M. Di Penta; S. Panichella	2017	A Tale of CI Build Failures: An Open Source and a Financial Organization Perspective
S200	A. J. Younge; K. Pedretti; R. E. Grant; R. Brightwell	2017	A Tale of Two Systems: Using Containers to Deploy HPC Applications on Supercomputers and Clouds
S201	S. Wongkampoo; S. Kiattisin	2017	Atom-Task Precondition Technique to Optimize Large Scale GUI Testing Time based on Parallel Scheduling Algorithm
S202	Wu C.-F.E., Burugula R.S., Yu H., Dubey N., Jann J., Nguyen M.	2017	Automation of cloud node installation for testing and scalable provisioning
S203	M. Shahin; M. A. Babar; M. Zahedi; L. Zhu	2017	Beyond Continuous Delivery: An Empirical Investigation of Continuous Deployment Challenges
S204	T. T. Brooks	2017	Big Data Complex Event Processing for Internet of Things Provenance: Benefits for Audit, Forensics, and Safety
S205	Ståhl D., Bosch J.	2017	Cinders: The continuous integration and delivery architecture framework
S206	Wettinger J., Breitenbücher U., Falkenthal M., Leymann F.	2017	Collaborative gathering and continuous delivery of DevOps solutions through repositories
S207	C. H. Kao	2017	Continuous evaluation for application development on cloud computing environments
S208	D. Stahl; T. Martensson; J. Bosch	2017	Continuous practices and devops: beyond the buzz, what does it all mean?
S209	Fitzgerald B., Stol K.-J.	2017	Continuous software engineering: A roadmap and agenda
S210	Metzger S., Durden D., Sturtevant C., Luo H., Pingintha-Durden N., Sachs T., Serafimovich A., Hartmann J., Li J., Xu K., Desai A.R.	2017	Eddy4R 0.2.0: A DevOps model for community-extensible processing and analysis of eddy-covariance data based on R, Git, Docker, and HDF5
S211	J. A. Kupsch; B. P. Miller; V. Basupalli; J. Burger	2017	From continuous integration to continuous assurance
S212	P. Perera; R. Silva; I. Perera	2017	Improve software quality through practicing DevOps
S213	S. Vost; S. Wagner	2017	Keeping Continuous Deliveries Safe
S214	Zimmermann, O	2017	Microservices tenets: Agile approach to service development and deployment
S215	Chung S.	2017	Object-oriented programming with DevOps
S216	Heinrich R., Van Hoorn A., Knoche H., Li F., Lwakatare L.E., Pahl C., Schulte S., Wettinger J.	2017	Performance engineering for microservices: Research challenges & directions
S217	Haili W., Renbin G., Congbin W., Lei G.	2017	Research and application of development model of information service for IOT of oil and gas production based on cloud architecture
S218	Z. Farahmandpour; S. Versteeg; J. Han; A. Kameswaran	2017	Service Virtualisation of Internet-of-Things Devices: Techniques and Challenges
S219	Bucena I., Kirikova M.	2017	Simplifying the devops adoption process
S220	A. van Deursen	2017	Software engineering without borders
S221	D. Spinellis	2017	State-of-the-Art Software Testing
S222	Martensson T., Stahl D., Bosch J.	2017	The EMFIS model - Enable more frequent integration of software
S223	Y. Zhao; A. Serebrenik; Y. Zhou; V. Filkov; B. Vasilescu	2017	The impact of continuous integration on other software development practices: A large-scale empirical study

S224	C. Parnin; E. Helms; C. Atlee; H. Boughton; M. Ghattas; A. Glover; J. Holman; J. Micco; B. Murphy; T. Savor; M. Stumm; S. Whitaker; L. Williams	2017	The Top 10 Adages in Continuous Deployment
S225	S. Palihawadana; C. H. Wijeweera; M. G. T. N. Sanjitha; V. K. Liyanage; I. Perera; D. A. Meedeniya	2017	Tool support for traceability management of software artefacts with DevOps practices
S226	E. Laukkanen; M. Paasivaara; J. Itkonen; C. Lassenius; T. Arvonen	2017	Towards Continuous Delivery by Reducing the Feature Freeze Period: A Case Study
S227	D. Ameller; C. Farr√©; X. Franch; D. Valerio; A. Cassarino	2017	Towards continuous software release planning
S228	C. Duffau; B. Grabiec; M. BlayFornarino	2017	Towards Embedded System Agile Development Challenging Verification, Validation and Accreditation: Application in a Healthcare Company
S229	Nidagundi P., Novickis L.	2017	Towards utilization of lean canvas in the devops software
S230	Hilton M., Nelson N., Tunnell T., Marinov D., Dig D.	2017	Trade-offs in continuous integration: Assurance, security, and flexibility
S231	Morris D., Voutsinas S., Hambly N.C., Mann R.G.	2017	Use of Docker for deployment and testing of astronomy software
S232	M. Zhao; F. Le Gall; P. Cousin; R. Vilalta; R. Mu√±oz; S. Castro; M. Peuster; S. Schneider; M. Siapera; E. Kapassa; D. Kyriazis; P. Hasselmeyer; G. Xilouris; C. Tranoris; S. Denazis; J. Martrat	2017	Verification and validation framework for 5G network services and apps
S233	Ur Rahman, Akond Ashfaque and Williams, Laurie	2016	Security Practices in DevOps
S234	F. Calefato; F. Lanubile	2016	A Hub-and-Spoke Model for Tool Integration in Distributed Development
S235	Di Nitto E., Jamshidi P., Guerriero M., Spais I., Tamburri D.A.	2016	A software architecture framework for quality-aware devops
S236	Hanappi O., Hummer W., Dustdar S.	2016	Asserting reliable convergence for configuration management scripts
S237	J. Bae; C. Kim; J. Kim	2016	Automated deployment of SmartX IoT-cloud services based on continuous integration
S238	Makki M., Van D., Joosen L.W.	2016	Automated workflow regression testing for multi-tenant SaaS: Integrated support in self-service configuration dashboard
S239	Schermann G., Schöni D., Leitner P., Gall H.C.	2016	Bifrost: Supporting continuous deployment with automated enactment of multi-phase live testing strategies
S240	Risdianto A.C., Shin J., Kim J.	2016	Building and operating distributed SDN-cloud testbed with hyper-convergent smartx boxes
S241	D. Liu; H. Zhu; C. Xu; I. Bayley; D. Lightfoot; M. Green; P. Marshall	2016	CIDE: An Integrated Development Environment for Microservices
S242	C. Vassallo; F. Zampetti; D. Romano; M. Beller; A. Panichella; M. Di Penta; A. Zaidman	2016	Continuous Delivery Practices in a Large Financial Organization
S243	T. Savor; M. Douglas; M. Gentili; L. Williams; K. Beck; M. Stumm	2016	Continuous Deployment at Facebook and OANDA
S244	Rossi C., Shibley E., Su S., Beck K., Savor T., Stumm M.	2016	Continuous deployment of mobile software at facebook (showcase)
S245	C. Pang; A. Hindle	2016	Continuous Maintenance
S246	M. Staples; L. Zhu; J. Grundy	2016	Continuous Validation for Data Analytics Systems
S247	Hadar E., Hadar I.	2016	CURA: Complex-system Unified reference architecture position paper: A practitioner view
S248	Riungu-Kalliosaari L., Mäkinen S., Lwakatare L.E., Tiihonen J., Männistö T.	2016	DevOps adoption benefits and challenges in practice: A case study

S249	Colavita F.	2016	Devops movement of enterprise agile breakdown silos, create collaboration, increase quality, and application speed
S250	M. Callanan; A. Spillane	2016	DevOps: Making It Easy to Do the Right Thing
S251	Sheridan C., Whigham D., Artac M.	2016	DICE fault injection tool
S252	Amith Raj MP; A. Kumar; S. J. Pai; A. Gopal	2016	Enhancing security of Docker using Linux hardening techniques
S253	M. T. Rahman; L. Querel; P. C. Rigby; B. Adams	2016	Feature Toggles: Practitioner Practices and a Case Study
S254	Mäkinen S., Leppänen M., Kilamo T., Mattila A.-L., Laukkanen E., Pagels M., Männistö T.	2016	Improving the delivery cycle: A multiple-case study of the toolchains in Finnish software intensive enterprises
S255	Jones S., Noppen J., Lettice F.	2016	Management challenges for devops adoption within UK SMEs
S256	Artač M., Borovšak T., Di Nitto E., Guerriero M., Tamburri D.A.	2016	Model-Driven continuous deployment for quality devops
S257	B. Adams; S. McIntosh	2016	Modern Release Engineering in a Nutshell -- Why Researchers Should Care
S258	Kroß J., Willnecker F., Zwickl T., Krcmar H.	2016	PET: Continuous performance evaluation tool
S259	Ohtsuki M., Ohta K., Kakeshita T.	2016	Software engineer education support system ALECSS utilizing devOps tools
S260	A. A. U. Rahman; L. Williams	2016	Software Security in DevOps: Synthesizing Practitioners‚Äô Perceptions and Practices
S261	Cito J., Mazlami G., Leitner P.	2016	TemPerf: Temporal correlation between performance metrics and source code
S262	Shahin M., Babar M.A., Zhu L.	2016	The Intersection of Continuous Deployment and Architecting Process: Practitioners' Perspectives
S263	R. Punjabi; R. Bajaj	2016	User stories to user reality: A DevOps approach for the cloud
S264	Gottesheim, Wolfgang	2015	Challenges, Benefits and Best Practices of Performance Focused DevOps
S265	Shtern, Mark and Simmons, Bradley and Smit, Michael and Lu, Hongbin and Litoiu, Marin	2015	Performance Management and Monitoring
S266	Stillwell M., Coutinho J.G.F.	2015	A DevOps approach to integration of software components in an EU research project
S267	E. Salant; P. Leitner; K. Wallbom; J. Ahtes	2015	A framework for a cost-efficient cloud ecosystem
S268	D. Bruneo; F. Longo; G. Merlino; N. Peditto; C. Romeo; F. Verboso; A. Puliafito	2015	A Modular Approach to Collaborative Development in an OpenStack Testbed
S269	Rajagopalan S., Jamjoom H.	2015	App–Bisect: Autonomous healing for microservice-based apps
S270	H. Chen; R. Kazman; S. Haziyev; V. Kropov; D. Chtchourov	2015	Architectural Support for DevOps in a Neo-Metropolis BDaaS Platform
S271	Scheuner J., Cito J., Leitner P., Gall H.	2015	Cloud workBench: Benchmarking IaaS providers based on infrastructure-ascode
S272	S. Gebert; C. Schwartz; T. Zinner; P. Tran-Gia	2015	Continuously delivering your network
S273	Lehtonen T., Suonsyrjä S., Kilamo T., Mikkonen T.	2015	Defining metrics for continuous delivery and deployment pipeline
S274	D. Bruneo; F. Longo; G. Merlino; N. Peditto; C. Romeo; F. Verboso; A. Puliafito	2015	Enabling Collaborative Development in an OpenStack Testbed: The CloudWave Use Case
S275	Wettinger J., Andrikopoulos V., Leymann F.	2015	Enabling devops collaboration and continuous delivery using diverse application environments
S276	M. Soni	2015	End to End Automation on Cloud with Build Pipeline: The Case for DevOps in Insurance Industry, Continuous Integration, Continuous Testing, and Continuous Delivery
S277	Segall I., Tzoref-Brill R.	2015	Feedback-driven combinatorial test design and execution
S278	Vasilescu B., Yu Y., Wang H., Devanbu P., Filkov V.	2015	Quality and productivity outcomes relating to continuous integration in GitHub

S279	M. de Bayser; L. G. Azevedo; R. Cerqueira	2015	ResearchOps: The case for DevOps in scientific applications
S280	E. Laukkanen; M. Paasivaara; T. Arvonen	2015	Stakeholder Perceptions of the Adoption of Continuous Integration-A Case Study
S281	A. A. U. Rahman; E. Helms; L. Williams; C. Parnin	2015	Synthesizing Continuous Deployment Practices Used in Software Development
S282	N. Rathod; A. Surve	2015	Test orchestration a framework for Continuous Integration and Continuous deployment
S283	A. Wahaballa; O. Wahballa; M. Abdellatief; H. Xiong; Z. Qin	2015	Toward unified DevOps model
S284	M. Virmani	2015	Understanding DevOps & bridging the gap from continuous integration to continuous delivery
S285	Chen J., Xu X., Osterweil L.J., Zhu L., Brun Y., Bass L., Xiao J., Li M., Wang Q.	2015	Using simulation to evaluate error detection strategies: A case study of cloudbased deployment processes
S286	J. Engblom	2015	Virtual to the (near) end: Using virtual platforms for continuous integration
S287	B. S. Farroha; D. L. Farroha	2014	A Framework for Managing Mission Needs, Compliance, and Trust in the DevOps Environment
S288	S. Harrer; C. R√∂ck; G. Wirtz	2014	Automated and Isolated Tests for Complex Middleware Products: The Case of BPEL Engines
S289	Fitzgerald B., Stol K.	2014	Continuous software engineering and beyond: Trends and challenges
S290	C. A. Cois; J. Yankel; A. Connell	2014	Modern DevOps: Optimizing software development through effective system interactions
S291	S. A. Wright; D. Druta	2014	Open source and standards: The role of open source in the dialogue between research and standardization
S292	S. W. Hussaini	2014	Strengthening harmonization of Development (Dev) and Operations (Ops) silos in IT environment through systems approach
S293	S. Bellomo; N. Ernst; R. Nord; R. Kazman	2014	Toward Design Decisions to Enable Deployability: Empirical Study of Three Projects Reaching for the Continuous Delivery Holy Grail
S294	Erculiani F., Abeni L., Palopoli L.	2014	UBuild: Automated testing and performance evaluation of embedded linux systems
S295	S. Neely; S. Stolt	2013	Continuous Delivery? Easy! Just Change Everything (Well, Maybe It Is Not That Easy)
S296	Schaefer A., Reichenbach M., Fey D.	2013	Continuous integration and automation for DevOps
S297	D. G. Feitelson; E. Frachtenberg; K. L. Beck	2013	Development and Deployment at Facebook
S298	S. Meyer; P. Healy; T. Lynn; J. Morrison	2013	Quality Assurance for Open Source Software Configuration Management

Information about authors / Информация об авторах

Brian PANDO, Computer and Systems Engineer. Research interests: Software Engineering, Web Development, Software Development.

Брайан ПАНДО, компьютерный и системный инженер. Научные интересы: программная инженерия, веб-разработка, разработка программного обеспечения.

Abraham DÁVILA is a Principal Professor of the Computer Engineering program and is a Doctoral Candidate in Software Engineering, in the field of process improvement. Field of scientific interests: Software engineering, Software quality process, Software quality product, Education in software engineering, Innovations based on software.

Авраам ДАВИЛА – профессор программы компьютерной инженерии и докторант в области программной инженерии. Область научных интересов: программная инженерия, процесс качества программного обеспечения, образование в области программной инженерии, инновации на основе программного обеспечения.

DOI: 10.15514/ISPRAS-2023-35(1)-12

A Systematic Mapping Study of ISO/IEC 29110 and Software Engineering Education

[1] *L. Vives, ORCID: 0000-0003-0280-2990 <pcsilviv@upc.edu.pe>*
[2] *K. Melendez, ORCID: 0000-0002-9518-3879 <kmelendez@pucp.edu.pe>*
[2] *A. Dávila, ORCID: 0000-0003-2455-9768 <abraham.davila@pucp.edu.pe>*

[1] *Peruvian University of Applied Sciences*
Lima, Peru, 15023
[2] *Pontificia Universidad Católica del Perú,*
Lima, Perú, 15088

Abstract. This article presents a study of the publications made on the ISO/IEC 29110 standard in the university context, especially from the perspective of software engineering education. ISO 29110 is a life cycle profiles for very small entities on systems and software engineering standard, published in many parts. ISO 29110, since its publication in 2011 and its continuous evolution to these days, is the subject of study in different contexts, with education being a relevant axis. Considering, that software engineering education has implications in the software industry in emerging countries, it is necessary to identify and consolidate the work done in this context. In this study, the main research question was what researches have been done at ISO 29110 in the training of software engineers? To answer this question, a systematic mapping study (SMS) was performed. In the SMS, 241 articles were obtained with search string and 17 of them became as primary study after a process selection. Based on these studies, it was possible to determine that the software engineering Basic profile of ISO 29110 and its processes (Project Management and Software Implementation) have been the most studied. Besides, it was identified that project-oriented learning and gamification techniques have been the most used ISO 29110 learning strategies in the training of future software industry professionals.

Keywords: ISO/IEC 29110; systems and software engineering; life cycle profiles; software engineering education

For citation: Vives L., Melendez K., Dávila A. A Systematic Mapping Study of ISO/IEC 29110 and Software Engineering Education. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 189-204. DOI: 10.15514/ISPRAS-2023-35(1)-12

Acknowledgments. Authors recognize reviews from members of Grupo de Investigación y Desarrollo en Ingeniería de Software- Pontificia Universidad Católica del Perú (GIDIS-PUCP).

Систематический обзор литературы по стандарту ISO/IEC 29110 и образованию в области программной инженерии

[1] *Л. Вивес, ORCID: 0000-0003-0280-2990 <pcsilviv@upc.edu.pe>*
[2] *К. Мелендес, ORCID: 0000-0002-9518-3879 <kmelendez@pucp.edu.pe>*
[2] *А. Давила, ORCID: 0000-0003-2455-9768 <abraham.davila@pucp.edu.pe>*
[1] *Перуанский университет прикладных наук*
Перу, 15023, Лима
[2] *Папский католический университет Перу,*
Перу, 15088, Лима

Аннотация. В этой статье представлено исследование публикаций, посвященных стандарту ISO/IEC 29110 в университетском контексте, особенно с точки зрения образования в области разработки программного обеспечения. ISO 29110 состоит из профилей жизненного цикла разработки систем и программного обеспечения, ориентированных на использование в очень мелких предприятиях, и опубликован во многих частях. Стандарт ISO 29110, с момента его публикации в 2011 году и за время его непрерывного развития по сей день, является предметом изучения в различных контекстах, причем важным элементом является образование. Учитывая, что образование в области разработки программного обеспечения имеет значение для индустрии программного обеспечения в развивающихся странах, необходимо определить и консолидировать работу, проделанную в этом контексте. В этом исследовании основным исследовательским вопросом было то, какие исследования были проведены в соответствии с ISO 29110 при подготовке инженеров-программистов? Чтобы ответить на этот вопрос, было проведено систематическое исследование литературных источников. В ходе работы с помощью поисковой строки была получена 241 статья, и после отбора 17 из них стали основными для дальнейшего исследования. Основываясь на полученных результатах, можно судить, что наиболее изученными являются базовый профиль разработки программного обеспечения и его процессы управления проектами и внедрения программного обеспечения стандарта ISO 29110. Кроме того, было выявлено, что при подготовке будущих специалистов индустрии программного обеспечения наиболее часто используемыми стратегиями обучения по стандарту ISO 29110 были проектно-ориентированное обучение и методы геймификации.

Ключевые слова: ISO/IEC 29110; системная и программная инженерия; профили жизненного цикла; обучение разработчиков программного обеспечения

Для цитирования: Вивес Л., Мелендес К., Давила А. Систематический обзор литературы по стандарту ISO/IEC 29110 и образованию в области программной инженерии. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 189-204. DOI: 10.15514/ISPRAS–2023–35(1)–12

Благодарности. Авторы признательны за отзывы членам Группы исследований и разработок в области программной инженерии Папского католического университета Перу.

1. Introduction

The ISO/IEC 29110 is a set of standards developed for very small entities (VSEs) in the IT field with up to 25 people [1]. In 2011, the publication of the software engineering Basic profile of ISO 29110 begins [2] and 2014 for systems engineering [3]. In [3], it is noted that other documents, e.g. Agile, are under development for VSEs. Also, according to [4], the research related to ISO 29110, has been growing all these years.

In 2012, in a United Nations report [5], is pointed out that the software industry in emerging countries is considered strategic to ensure adequate growth of their economies. Furthermore, in [5] two important issues are pointed out as a need for the software industry in emerging countries: (i) the need to apply appropriate (quality) process models, and (ii) the education of software industry professionals in these emerging countries. These two needs find an initial response in the 29110 standards [1] and a potential answer in the findings of [4], where education is identified as a focus of research interest.

Studies on software process models, in the training of professionals, have been carried out in other cases, such as: MoProSoft [6], CMMI [7] and ISO/IEC 12207 [8], [9]. In the Systematic Mapping Study (SMS) [4] on 179 selected studies, up to 2018, it was identified that 44 of them are classified in the context of education. However, as the authors point out [4], they consider various facets of education, including university and industry contexts.

Taking into account the work of [4], and considering that other standards are developed for 29110 such as Agile and DevOps [10], and the need expressed in [5], it is expected that research on ISO 29110 will continue in the university context. This research interest is justified by the benefit of training software engineering students [9], who can then apply it in their professional life [11], [12], [5], [13] in a company or in a software-based startup [14], [15].

On the other hand, software engineering worldwide, from its origins in 1968 [16], [17], through its consolidation as a discipline by means of a curriculum guide in 2004 [18], [19], has changed significantly in all those years [20]. Nowadays, exists new challenges for software engineering, both at the technical level (emerging technologies) and at the level of business models within the software industry itself [5]. Also, the software industry is mostly made up, worldwide, of micro, small and medium-sized companies [5], [21]; and in emerging countries, they have other characteristics and needs.

The objective of this study is to identify, detail what is researched about ISO 29110 in software engineering professional training. For this purpose, we carried out a Systematic Mapping Study (SMS). The article is organized as follows: in Section II, some concepts and related works are described; in Section III, the SMS is developed; in Section IV, the results are analyzed; and, in Section V, the conclusions are presented.

2. Background and Related Work

This section briefly introduces the concepts of ISO 29110, the discipline of software engineering and related work.

2.1 ISO/IEC 29110

The International Organization for Standardization - ISO, concerned about the global software industry, initiates the development and publication of a family of standards that has been named ISO/IEC 29110 and known as VSE (very small entities) project [3]. ISO 29110 has developed, among others, a process model for organizations that develop software [3]. This standard is based on MoProSoft (developed for the Mexican software industry [3], [22]) and some contributions from participants of the Competisoft Project. According to [23], ISO 29110 represents an aid to improve development processes in the software life cycle.

According to Part 1 of ISO 29110 [1], the group of standards is organized into an overview document, a group of profiles documents with their assessments and certifications, and a group of implementation guidelines. A profile extracts and tailors the elements of a standard to meet specific needs [2]. Profiles published of software engineering [10] are 4 profiles: Part 5-1-1:2012 Entry profile, Part 5-1-2:2011 Basic profile, Part 5-1-3:2017 Intermediate profile and Part 5-1-4:2018 Advance profile.

2.2 Software Engineering Discipline

Software Engineering discipline evolves in the context of other disciplines related to computing. Therefore, the Computing Curricula of 2001 or CC2001 [19], constitutes a first milestone since it starts a process of identification of 5 domains of its own and initial overlaps between these disciplines are identified. In 2016, the Guide was updated and published under the name CC2020 or Computing Curricula 2020, introducing the competency-based approach [27] and 7 new sub-disciplines [28]. In 2014, the SWEBOK v3 (Software Engineering Body of Knowledge version 3)

guide is published [30]. In SWEBOK V.3, ISO/IEC 29110 is included [30]. Finally, two important aspects of software engineering education are: (i) the level of introduction of software engineering programs in universities worldwide; which is still incipient [32]; and, (ii) the amount of software engineering contents (techniques and practices) incorporated in university programs that train software industry professionals is not enough [13].

2.3 Related Work

As related works we identified: (i) an SMS on ISO 29110 conducted, in [4] classifying the results according to emerging axes of the study, noting that the topic of education is one of the most researched; (ii) a Report for the 10th anniversary of ISO 29110 in [11], which presents a compilation of ISO 29110 implementation experiences in different contexts (industry and academic) and from various countries; and (iii) an SMS on Software Engineering Education (SEE) in [33], that points out, among other things, a change towards new trends in software engineering: global software development and lean software startup; in academic and industry contexts; (iv) a systematic mapping study was carried out on the application of serious games in the teaching of the software development life cycle in [34], concluding that serious games are a motivational tool to increase the knowledge and learning of students.

3. Research Method

For this research, a SMS was carried out taking as the main reference point Petersen's proposal [37].

3.1 Research Questions

The achievement of the objective is translated into the following research questions (RQ):

RQ1. How has the number of ISO 29110 publications in university education evolved over time?

RQ2. What is the studies distribution by type of article in relation to ISO 29110 in education?

RQ3. What ISO 29110 processes have been addressed in university education?

RQ4. What kind of pedagogical techniques or activities have been used for learning ISO 29110 in university education?

RQ5. What academic purpose is served by studies of ISO 29110 in the university context?

3.2 Research Protocol

Based on Petersen's recommendation [37], the Population and Intervention scheme (P AND I) was used to define the search string. The population is “ISO/IEC 29110”, and intervention is “education”. From the main terms and alternate terms, the search string was established as: *("ISO/IEC 29110" OR "ISO 29110") AND ("education" OR "program" OR "university" OR "training" OR "undergraduate" OR "postgraduate" OR "student" OR "academic" OR "course" OR "doctor" OR "master")*. This search string has applied to the follow digital database: Web of Science, IEEE Xplore, ACM DL, ProQuest, Scopus, and Springer. Also, as ISO/IEC 29110 is based on MoProSoft, a Spanish search string *("ISO/IEC 29110" OR "ISO 29110") AND ("educación" OR "programa" OR "universidad" OR "entrenamiento" OR "pregrado" OR "postgrado" OR "estudiante" OR "academico" OR "curso" OR "doctorado" OR "maestria")* was applied to Scielo digital database. It is important to note that papers in Scielo have a title and abstract write both in Spanish and Portuguese.

The inclusion (IC#) and exclusion (EC#) criteria established were:

IC1. Refers to ISO 29110 at the university context (or similar).

IC2. Written in English, Spanish or Portuguese.

EC1. Duplicates.

EC2. Refers to ISO 29110 in a non-university environment, such as companies or government agencies.

EC3. Content does not present information about the research questions.

EC4. Not available as full text.

3.3 Selection and data extraction

The selection process was carried out in five stages (see Fig. 1), starting with the execution of the search string in March 2022, where 249 articles were found. It was applied the inclusion and exclusion criteria in stages, reading titles, abstracts and contents. In addition, as indicated by [37], it was decided not to perform quality assessment. After the selection process, 19 articles were obtained as primary studies (see Appendix A)[1].

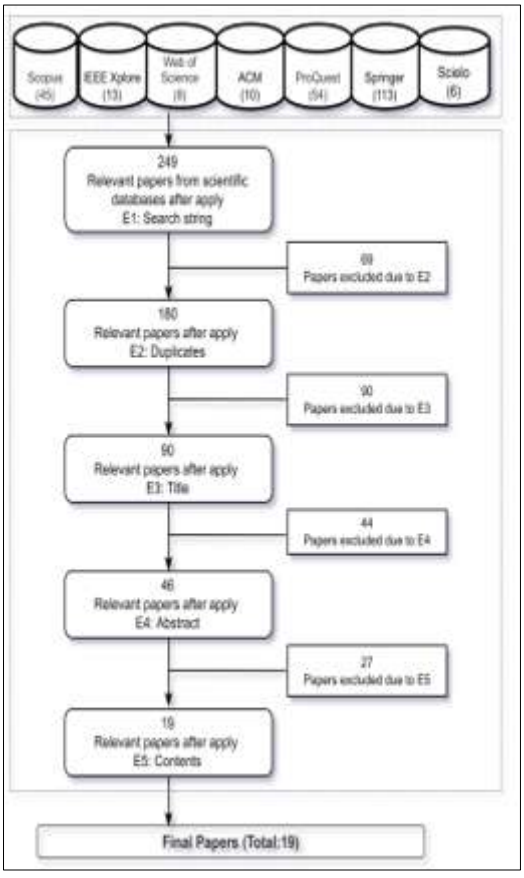

Fig. 1. SMS process selection

To perform the extraction, a spreadsheet was defined with the necessary structure to collect the data (see Table 1). In this table, data from de papers were recorded, and categorized to answer the RQs, see Appendix B. In addition, a pilot was conducted to verify the criteria and determine whether the questions could make sense in a research sample. For this pilot, prior to formal selection, a group of 10 articles with high potential that could become primary studies was extracted. The group was reduced to 7 articles after reading the titles and applying the established criteria. Data were extracted

[1] The whole data from selection process are available at https://drive.google.com/drive/folders/1vURTrYIipJuBQiC2PJIwur4HoatyzzJD?usp=sharing.

from this final group on the extraction sheet and the questions and format could be verified and adjusted.

Table 1. Data extraction item

Data item	Details	RQ
Bibliographic reference	Title, authors(s), year of publication.	RQ1
Paper type	Paper article, paper conference.	RQ1
Name of the journal or conference	Name of the journal or event where the article is published.	RQ2
Process of the ISO/IEC 29110	ISO/IEC 29110 processes that have been used in the academic environment.	RQ3
Pedagogical techniques or activities.	Pedagogical techniques or activities used for learning ISO/IEC 29110 in the academic field.	RQ4
Academic purpose	Academic proposals based on ISO 29110 that contribute to the field of education	RQ5

4. Results

In this section, the results and findings, as well as the validity threats are presented.

4.1 RQ1. How has the number of ISO 29110 publications in university training evolved over time?

As shown in Fig. 2. The evolution, in the number of publications, of ISO 29110 in the university context, has remained almost constant, as it has been published in 2016 (3), 2017 (4), 2018 (3), 2019 (2), 2020 (2). And 2021 (1) However, it is noted that it starts in 2016, which is 5 years after the publication of Part 5-1-2 (software engineering Basic profile) [24]. Furthermore, it should be added that at least the Entry and Basic profiles are freely available from ISO home page.

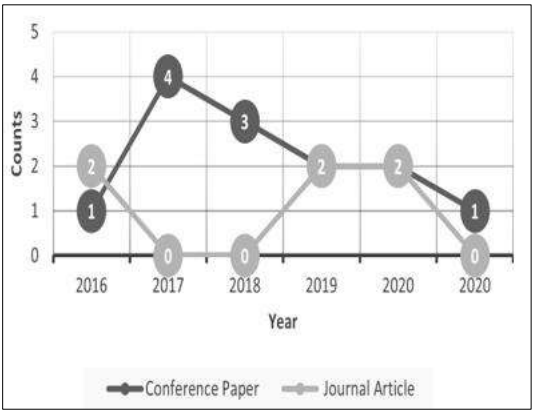

Fig. 2. Studies by type of work and years

Besides, it was noticed that researchers from Latin American have 14 of the 19 primary studies. A possible justification for this is that ISO 29110 is based on MoProSoft, a model developed for the Mexican software industry, and that the ISO working group includes several Latin American researches.

4.2 RQ2. What is the distribution of studies by article type in relation to ISO/IEC 29110 in education?

Fig. 2 shows that the types of articles according to where they were published are conferences (13) and journals (6). On the side of the articles published in conferences (S01, S02, S04, S08, S10, S11, S12, S13, S14, S16, S17, S18, S19) the International Conference on Software Process Improvement

(CIMPS) event stands out, where 3 articles have been published. As for the articles published in journals (S03, S05, S06, S07, S09, S15), the journals Computer Standards and Interfaces and RISTI – Revista Ibérica de Sistemas e Tecnologias de Informação stand out, each with 2 articles.

4.3 RQ3. Which ISO 29110 processes have been addressed in university education?

From the primary studies (see Fig. 3), we have:

Profile level. The most referenced profile is the Basic profile (89%) – (S04, S06), in smaller percentage the Entry profile (11%) – (S01, S02, S03, S05, S07, S08, S09, S10, S11, S12, S13, S14, S15, S16, S17, S18, S19). And no studies are reported for the Intermediate or Advanced profiles. The percentage, in the Basic profile, may be, among others, that it is the first published and certifiable profile, and therefore, of greater interest for companies and researchers. Other possible reasons, attributable to the Intermediate and Advanced profiles, are presented at the end of this question.

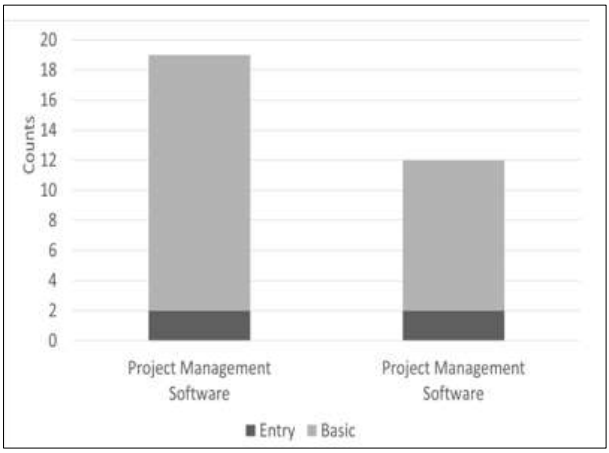

Fig. 3. Activities of the Entry and Basic profiles of ISO 29110

Process level. The most referenced processes are Project Management (19) and Software Implementation (12), both present in the Entry and Basic profiles. No processes have been reported for the Intermediate and Advanced profiles.

Level of processes by profiles. The studies for the Entry profile do not distinguish between the processes so it is understood that they implement both (S04, S06). The studies on Basic profile that only cover the project management process are 7 (S03, S11, S12, S14, S16, S17, S18), and both processes are 10 (S01, S02, S05, S07, S08, S09, S10, S13, S15, S19).

In addition, consulting with experts, there are at least 3 possible causes were identified as to why the Intermediate and Advanced profiles have not been adopted in the university context: (i) the publication of the Intermediate profile was 2017 and of the Advanced profile was 2018 [10] and they have not had the possibility to implement it, a situation accentuated by the pandemic since the end of 2019; (ii) the Intermediate and Advanced profile, focus on an organization that manages multiple projects, something that is unusual in university courses; and (iii) the lack of adequate dissemination of the other profiles in the university environment, accentuated by the COVID-19 pandemic that has forced several changes in university environments.

4.4 RQ4. What kind of pedagogical techniques or activities have been used for learning ISO 29110 in university training?

The application of ISO 29110 in the university environment has been carried out using mainly POL (project-oriented learning) 10 (53%) papers, by the gamification technique 7 (37%) papers and do

not specify the technique in other cases 2 (10%) papers (see Fig.4). The first case, project-oriented learning, involved software development projects within a course where students apply (and learn) ISO 29110 (S02, S03, S04, S05, S06, S09, S13, S14, S15, S18); although it is a controlled context, it offers real opportunities to learn the standard. The second case, the gamification technique, is a practical and enjoyable way to interact with the knowledge associated with the tasks provided in each process considered (S01, S08, S11, S12, S16, S17, S19). Finally, the studies that do not present pedagogical techniques (S07, S10), conducted a comparative analysis of the level of coverage of ISO 29110 with respect to university academic programs.

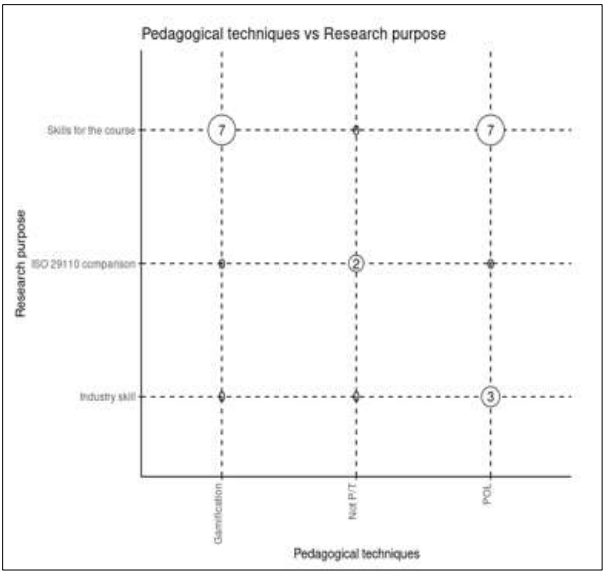

Fig. 4. Pedagogical techniques used for learning ISO 29110 in university environment

4.5 RQ5. What academic purpose is presented in the ISO 29110 studies in the university context?

The main purpose (see Fig. 4) is the development of cognitive skills within an academic course 14 (74%) papers, whose objective is to improve the mastery of ISO 29110 concepts and where the pedagogical techniques of project-oriented learning (S03, S05, S06, S13, S14, S15, S19) and gamification predominate (S01, S08, S11, S12, S16, S17, S018).

On the other hand, 3 (16%) papers of the studies focus on generating technical skills in the management of ISO 29110 applied to cases in the software development industry and use the project-oriented learning technique (S02, S04, S09).

Likewise, 2 (11%) papers of the researches perform an analysis of curricular content versus ISO 29110 activities and do not have pedagogical techniques for this purpose (S07, S10). In these papers, the contents of 4 Mexican curricula (including its courses) of Software Engineering, Computer Science, Computer Engineering and Computer Engineering programs are analyzed and the coverage of ISO 29110 processes and activities is evaluated through a comparative analysis. The analysis of the curricular frameworks with respect to the software engineering Basic profile is performed based on descriptions of both. There are no studies that evaluate the level of adherence to the Basic profile of the projects and results of software development projects carried out by students in their university courses. As can be seen, there is a predominance of ISO 29110 papers in university education that look to generate cognitive skills in students according to their level of education: (i) studies that focus only on undergraduate (S01, S02, S03, S05, S06, S08, S13, S14, S16, S17, S18, S19) cover the last 5 semesters and to Software Quality, Software Engineering I, II or III courses; (ii) in studies

that cover undergraduate and masters (S11, S12 and S15), reinforcements of project management knowledge and software process improvement are described. In addition, two papers (S04, S09) report the participation of undergraduate students in ISO 29110 implementation experiences, which provided them with an enriching experience with real cases from the software development industry. Two studies related to the analysis of course content and its relationship with ISO 29110 activities (S07, S10) have allowed establishing some correspondence between what a student has as potential with respect to what the standard, at the software engineering Basic profile level is expected. Finally, from the synthesis of the conclusions of the papers (see Table 2), it can be noted that: (i) there is an effort to raise awareness among students about the usefulness of 29110 to the industry; (ii) learn the concepts and application of the standard, and (iii) apply techniques such as POL and gamification to achieve the objectives mentioned above.

Table 2. Synthesis of the conclusions from primary studies

id	Contribution
S01	It creates a serious game-based framework for teaching the ISO 29110 standard, however, it has not been implemented.
S02	ISO 29110 was implemented in a university software development center, with students and teachers, based on its own methodology. It was identified that the activities of: evaluation and control of the project management process, as the most difficult for students.
S03	The students who followed the ISO 29110 guide of the implementation process achieved a better quality of the software product.
S04	They manage to verify that it is possible to train students in the development of real software projects using the ISO 29110 standard.
S05	They develop a methodology to achieve ISO 29110 certification and the students evidenced a better understanding of the standard by participating in real projects.
S06	Integrates agile methods, quality standards for teaching ISO 29110, achievn ing an acceptable level of understanding of the concepts of the standard.
S08	A better understanding of ISO 29110 concepts is achieved, based on a monopoly-type board game with roles.
S09	It is determined that a set of deficiencies and lack of skills of a small organization (students) are overcome with the implementation of ISO 29110.
S11	Students achieve course proficiency related to ISO 29110 project management concepts and activities using a serious simulation-based game.
S12	The students acquire the knowledge using a serious game based on the simulation of the ISO 29110 Project Management process for the training of university students.
S13	The results indicate that the solution based on ISO 29110 can be adapted to software engineering student projects.
S14	A monitoring tool for ISO 29110 processes was developed for a better understanding and evaluation of student achievement.
S15	The use of ISO 29110 instead of CMMI facilitated the understanding and implementation of a suitable software engineering framework in the software process improvement course.
S16	A serious game was developed as a way to support the understanding of the ISO 29110 project management process .
S17	Was achieved the understanding and management of the ISO 29110 project management process by students, based in a serious card game.
S018	The game helps teaching and reinforces knowledge of structures and elements of ISO 29110 based on a serious game.
S019	Student participation on process improvement projects with ISO 29110, contributes to the increase of student results in engineering training according to what is established by ABET in the USA or ICACIT in Peru.

4.6 Validity threats

This section presents the validity threats related to study selection, data validity and research validity according to [38] and [39].

4.6.1 Validity of study selection

The procedure consisted of generating and validating the search string through the most popular search engines, such as: Web of Science, IEEE Xplore, ACM DL, ProQuest, Scopus and Springer. In addition, it considered the period between 2011 (the year ISO 29110 was published) and March 2021. The search string was created according to the authors' knowledge about ISO 29110 and university education in software engineering. In the study, 17 articles were selected which were contrasted with the 44 obtained for in the context of education by [4]. From the 17 studies selected, it was determined that 9 studies were considered in the mapping of [4] and the remaining 8, since they are more recent, are not included in Larrucea's mapping [4]. The other 35 Larrucea's studies that were not selected are due to the fact that the author used the concept of education in a more general way and not as something specific to the university environment. Some examples are: frameworks or tools developed based on ISO and applied to educational processes [40], [41], [42]; teaching development teams [43], use of gamification in development teams [44], among others. After obtaining the secondary studies, the articles were distributed to two of the researchers who evaluated the inclusion and exclusion process of the articles; each article received 2 reviews. Finally, a third author randomly reviewed the selection of articles.

4.6.2 Validity of the data

For data validation, the authors held working meetings to discuss the inclusion or exclusion of articles. Statistical tools were not used, since it was not necessary to test hypotheses. The authors have complied with reviewing and applying the stages proposed for the systematic mapping study proposed by [37]. The authors acknowledge that the places of publication of the articles are relevant and of interest to the software engineering community.

4.6.3 Validity of the research

This process considered the experience of the researchers who are familiar with the concepts, methods and terms used in the research, since they have published articles and are reviewers of topics addressed in this research. Likewise, bias and subjectivity are minimized, given that data extraction and data validity were based on the opinion of the first author in contrast to the opinion of the other two authors. Finally, because the Petersen methodology [37] has been followed as reliably as possible, it is certain that the study covers all ISO 29110 investigations in the university setting. However, the number of primary studies does not allow the results to be generalized; but if it has made it possible to identify relevant aspects, they can be used as a basis for future research.

5. Conclusions

In this research, an SMS was defined and performed based on the methodological procedure proposed by Petersen [37], where 17 primary studies were obtained to answer the research questions. It was determined that the software engineering Basic profile and its two processes: Project Management and Software Implementation of ISO 29110, are the most used in the training of university students with respect to the other profiles. The authors of these studies point out that the industries in their countries require people with more competence in ISO/IEC 29110 to provide greater benefits for small software development companies. In this sense, it is important that the university curriculum covers the topics and that the training achieves these competencies in the graduates.

From our study, it is highlighted that two didactic techniques are the most used for students to learn ISO 29110. The first is the didactic technique of project-oriented learning, which provides a space for the experience of skills such as teamwork, holistic vision, critical thinking, and analytical skills in real or realistic situations. The second is the didactic technique of gamification applied with the objective of extrinsically motivating the student to achieve the course objectives.

Moreover, it can be noted that there is a great effort in the field of research in Latin America, which may correspond to the fact that the base of ISO 29110 is MoProSoft. However, from the works of [4] and [45], it can be noted that there are several works that are still being deployed in the industry and more slowly in the academy. Also, here have been studies on the relationship between ISO/IEC 29110 and the curriculum of software engineering programs or related. Such as the verification of the practical exercise or application of ISO 29110 in the courses or projects at the end of the career.

References

[1] ISO/IEC TR 29110-1:2016 Systems and Software Engineering – Lifecycle Profiles for Very Small Entities (VSEs) – Part 1: Overview. Geneva, 2016.

[2] Laporte C.Y., O'Connor R.V., García Paucar L. The Implementation of ISO/IEC 29110 Software Engineering Standards and Guides in Very Small Entities. Communications in Computer and Information Science, vol. 599, Springer, 2016, pp. 162-179.

[3] O'Connor R.V., Laporte C.Y. The evolution of the ISO/IEC 29110 set of standards and guides. International Journal of Information Technologies and Systems Approach, vol. 10, issue 1, 2017, article no. 1, 21 p.

[4] Larrucea X., Fernandez-Gauna B. A mapping study about the standard ISO/IEC29110. Computer Standards & Interfaces, vol. 65, 2019, pp. 159-166.

[5] Information Economy Report 2012. The Software Industry and Developing Contries. United Nations Publication UNCTAD/IER/2012, 2012, 126 p.

[6] Muñoz M., Peña A. et al. Coverage of the university curricula for the Software Engineering industry in Mexico. IEEE Latin America Transactions, vol. 14, issue 5, 2016, pp. 2382-2388.

[7] Dagnino A. Increasing the effectiveness of teaching software engineering: A University and industry partnership. In Proc. of the IEEE 27th Conference on Software Engineering Education and Training, 2014, pp. 49-54.

[8] Aydan U., Yilmaz M. et al. Teaching ISO/IEC 12207 Software Lifecycle Processes: A Serious Game Approach. Computer Standards & Interfaces, vol. 54, part 3, 2017, pp. 129-138.

[9] Jovanovic V., Andres L. Use of Software Engineering Standards in Teaching. In Proc. of the 4th IEEE International Software Engineering Standards Symposium and Forum, 1999, pp. 122-130.

[10] ISO Portal Catalogue of standards of 29110. Available at: https://www.iso.org/search.html?q=29110&hPP=10&idx=all_en&p=0&hFR%5Bcategory%5D%5B0%5D=standard, accessed Jul. 14, 2021.

[11] Laporte C.Y., O'Connor R.V. Software process improvement in graduate software engineering programs. In Proc. of the 1st International Workshop on Software Process Education, Training and Professionalism co-located with 15th International Conference on Software Process Improvement and Capability dEtermination (SPICE 2015), 2016, pp. 18-24.

[12] Muñoz M., Peña A. et al. Analysis of Coverage of Moprosoft Practices in Curricula Programs Related to Computer Science and Informatics. Advances in Intelligent Systems and Computing, vol. 405, Springer, 2016, pp. 15-24.

[13] Garousi V., Giray G. et al. Aligning software engineering education with industrial needs: A meta-analysis. Journal of Systems and Software, vol. 156, 2019, pp. 65–83.

[14] García Paucar L., Laporte C. et al. Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru. Software Quality Professional, vol. 17, issue 2, 2015, pp. 16-29.

[15] Laporte C.Y., Munoz M. et al. Applying Software Engineering Standards in Very Small Entities: From Startups to Grownups. IEEE Software, vol. 35, issue 1, 2018, pp. 99-103.

[16] Naur P., Randell B. Software Engineering. Report on a conference sponsored by the NATO Science Committee,1968, 138 p. Available at: https://onlinebooks.library.upenn.edu/webbin/book/lookupid?key=olbp48950, accessed Jul. 14, 2021.

[17] Mahoney M.S. Finding a History for Software Engineering. IEEE Annals of the History of Computing, vol. 26, issue 1, 2004, pp. 8-19.

[18] The Joint Task Force on Computing Curricula. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. ACM, 2004, 135 p.

[19] Shackelford R., McGettrick A. et al. Computing Curricula 2005: The overview report. In Proc. of the 37th SIGCSE Technical Symposium on Computer Science Education, 2006, pp. 456-457.

[20] Boehm B. A View of 20th and 21st Century Software Engineering. In Proc. of the 28th International

Conference on Software Engineering, 2006, pp. 12-29.
[21] Takahashi M. Problems of Small and Mid-Sized Enterprises in Japan'S Software Industry. Eurasian Journal of Economics and Finance, vol. 8, issue 4, 2020, pp. 274-278.
[22] Oktaba H., García F. et al. Software Process Improvement: The Competisoft Project. Computer, vol. 40, issue 10, 2007, pp. 21-28.
[23] Suteeca K. A Software Process Gap Analysis Methodology for Very Small Entity. In Proc. of the Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT and NCON), 2020, pp. 190-193.
[24] ISO/IEC TR 29110-5-1-2:2011 Software Engineering – Lifecycle Profiles for Very Small Entities (VSEs) – Part 5-1-2: Management and Engineering Guide: Generic Profile Group: Basic Profile. Geneva, 2011.
[25] ISO/IEC TR 29110-5-1-3:2017 Systems and Software Engineering – Lifecycle Profiles for Very Small Entities (VSEs) – Part 5-1-3: Software Engineering – Management and Engineering Guide: Generic Profile Group – Intermediate Profile. Geneva, 2017.
[26] ISO/IEC TR 29110-5-1-4:2018 System and Software Engineering – Lifecycle Profiles for very Small Entities (VSES) – PART 5-1-4: Software Eengineering: Management and Engineering Guidelines : Generic Profile Group: Advanced Profile. Geneva, 2018.
[27] Impagliazzo J., Clear A., Alrumaih H. Developing an overview of computing/engineering curricula via the CC2020 project. In Proc. of the IEEE World Engineering Education Conference (EDUNINE), 2018, pp. 1-4.
[28] Takada S., Cuadros-Vargas E. et al. Toward the visual understanding of computing curricula. Education and Information Technologies, vol. 25, no. 5, pp. 4231–4270, 2020.
[29] Joint Task Force on Computing Curricula. Software Engineering 2014. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Association of Computing Machinery, IEEE Computer Society, 2015, 134 p.
[30] Guide to the Software Engineering Body of Knowledge (SWEBOK), Version 3.0. IEEE Computer Society, 2014, 339 p.
[31] Software Engineering – Guide to the Software Engineering Body of Knowledge (SWEBOK). Technical Report ISO/IEC TR 19759:2005, 2005.
[32] Ouhbi S., Pombo N. Software engineering education: Challenges and perspectives. In Proc. of the IEEE Global Engineering Education Conference (EDUCON), 2020, pp. 202-209.
[33] Cico O., Jaccheri L, et al. Exploring the intersection between software industry and Software Engineering education – A systematic mapping of Software Engineering Trends. Journal of Systems and Software, vol. 172, 2021, article no. 110736, 28 p.
[34] Jimenez-Hernendez E.M., Oktaba H. et al. Serious Games When Used to Learn Software Processes: An Analysis from a Pedagogical Perspective. In Proc. of the 5th International Conference in Software Engineering Research and Innovation (CONISOFT), 2017, pp. 194-203.
[35] Britto R., Usman M. Bloom's Taxonomy in Software Engineering Education: A Systematic Mapping Study. IEEE Frontiers in Education Conference (FIE), 2015, pp. 1-8.
[36] Huang X., Zhang H. et al. A Research Landscape of Software Engineering Education. In Proc. of the 28th Asia-Pacific Software Engineering Conference (APSEC), 2021, pp. 181-191.
[37] Petersen K., Vakkalanka S., Kuzniarz L. et al. Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology, vol. 64, 2015, pp. 1-18.
[38] Ampatzoglou A., Bibi S. et al. Identifying, Categorizing and Mitigating Threats to Validity in Software Engineering Secondary Studies Apostolos. Information and Software Technology, vol. 106, 2019, pp. 201-230.
[39] Zhang H., Babar M.A., Tell P. Identifying relevant studies in software engineering. Information and Software Technology, vol. 53, issue 6, 2011, pp. 625-637.
[40] Al-Hawari F., Alufeishat A. et al. The Software Engineering of a Three-Tier Web-Based Student Information System (MyGJU). Computer Applications in Engineering Education, vol. 25, issue 2, 2017, pp. 242-263.
[41] Jimenez-Hernandez E.M., Oktaba H. et al. Methodology to construct educational video games in software engineering. In Proc. of 4th International Conference in Software Engineering Research and Innovation (CONISOFT), 2016, pp. 110–114.
[42] Espinosa-Curiel I.E., Rodríguez-Jacobo J., Fernández-Zepeda J. Graphical technique to support the teaching/learning process of software process reference models. Communications in Computer and Information Science, vol. 99, Springer, 2010, pp. 13–24.

[43] Basri S., O'Connor R.V. A study of software development team dynamics in SPI. Communications in Computer and Information Science, vol. 172, Springer, 2011, pp. 143-154.
[44] Muñoz M., Hernández L. et al. State of the Use of Gamification Elements in Software Development Teams. Communications in Computer and Information Science, vol. 748, Springer, 2017, pp. 249-258.
[45] Laporte C.Y. ISO/IEC JTC 1 SC 7 Working Group 24. 10th Anniversary Overview of Accomplishments. 2016, 132 p.

Appendix A. List of Primary Studies

S01 E. Bonilla Rivas and M. Munoz. Creación De Estrategias Para La Capacitación En El Estándar De Desarrollo De Software ISO/IEC 29110: Propuesta De Un Marco De Trabajo. In Proc. of the 9th International Conference on Software Process improvement (CIMPS), 2020, pp. 155-155, doi: 10.1109/cimps52057.2020.9390152.
S02 J.J. Minero, J. García, and E. Lara. Evaluation of the Implementation of the ISO/IEC 29110 Standard at the Software Development Center from the Institute Technological Superior of Nochistlán. In Proc. of the 9th International Conference on Software Process improvement (CIMPS), 2020, pp. 12-18, doi: 10.1109/CIMPS52057.2020.9390105.
S03 L. Castillo-Salinas, S. Sanchez-Gordon, J. Villarroel-Ramos, and M. Sánchez-Gordón. Evaluation of the implementation of a subset of ISO/IEC 29110 Software Implementation process in four teams of undergraduate students of Ecuador. An empirical software engineering experiment. Computer Standards & Interfaces, vol. 70, 2020, article no. 103430, doi: 10.1016/j.csi.2020.103430.
S04 M. De León-Sigg, J.L. Villa-Cisneros, and B.E. Solísrecéndez. Uso del Estándar ISO / IEC 29110 para Entrenar Estudiantes en Procesos de Ingeniería de Software. RISTI – Revista Ibérica de Sistemas e Tecnologias de Informação, no. 40, 2020, pp. 60-72, doi: 10.17013/risti.40.60-72.
S05 J.J. Minero, J. Garcia, and E.L. Instituto. A Methodology in the implementation of International Standards in Software Development Centers in Universities. In Proc. of the 8th International Conference on Software Process improvement (CIMPS), 2019, pp. 1-6, doi: 10.1109/CIMPS49236.2019.9082424.
S06 S.V. Hurtado-Gil. ÁgilUC : Proceso de desarrollo de software para equipos pequeños y una estrategia para su enseñanza. Revista Educación en Ingeniería, vol. 15, no. 29, 2020, pp. 21-27.
S07 M. Muñoz, J. Mejia, A. Peña, G. Lara, and C.Y. Laporte. Transitioning international software engineering standards to academia: Analyzing the results of the adoption of ISO/IEC 29110 in four Mexican universities. Computer Standards & Interfaces, vol. 66, 2019, article no. 103340, doi: 10.1016/j.csi.2019.03.008.
S08 V. Moura, D.I.A. Unirio, P. Unirio, and G. Santos. ProcSoft: A Board Game to Teach Software Processes Based on ISO/IEC 29110 Standard. In Proc. of the 17th Brazilian Symposium on Software Quality, 2018, pp. 363-372, doi: 10.1145/3275245.3276319.
S09 M. Muñoz, J. Mejia, and C. Y. Laporte. Implementation of ISO/IEC 29110 in software development centers from Mexican universities: An experience of the Zacatecas State. RISTI – Revista Ibérica de Sistemas e Tecnologias de Informação, no. 29, 2018, pp. 43-54, doi: 10.17013/risti.29.43-54.
S10 M. Muñoz, A.P.P. Negrón, J. Mejia, and G.L. Lopez. ISO/IEC 29110 and curricula programs related to computer science and informatics in Mexico: Analysis of practices coverage. Advances in Intelligent Systems and Computing, vol. 688, Springer, 2018, pp. 3-12, doi: 10.1007/978-3-319-69341-5_1.
S11 A. Calderón, M. Ruiz, and E. Orta. Integrating serious games as learning resources in a software project management course: The case of ProDec. In Proc. of the IEEE/ACM 1st International Workshop on Software Engineering Curricula for Millennials (SECM), 2017, pp. 21-27, doi: 10.1109/SECM.2017.3.
S12 A. Calderón, M. Ruiz, and R.V.O'Connor. Coverage of ISO/IEC 29110 project management process of basic profile by a serious game. Communications in Computer and Information Science, vol. 748, Springer, 2017, pp. 111-122, doi: 10.1007/978-3-319-64218-5_9.
S13 N. Chotisarn and D. Sanpote. A demonstration case study of software engineering senior project coordinating the international standard. In Proc. of the International Conference on Digital Arts, Media and Technology (ICDAMT), 2017, pp. 314-319, doi: 10.1109/ICDAMT.2017.7904983.
S14 M. Bougaa, S. Bornhofen, R.V. O'Connor, and A. Rivereю A standard based adaptive path to teach systems engineering: 15288 and 29110 standards use cases. In Proc. of the 11th Annual IEEE International Systems Conference (SysCon), 2017, pp. 1-8, doi: 10.1109/SYSCON.2017.7934712.
S15 C.Y. Laporte and R.V. O'Connor. Software process improvement in graduate software engineering programs. Software Quality Professional, vol. 18, no. 3, 2016, pp. 4-17.

S16 M.-L. Sanchez, R.V. O' Connor, R. Palacios, and E. Herranz. Bridging the Gap Between SPI and SMEs in Educational Settings: A Learning Tool Supporting ISO/IEC 29110. Communications in Computer and Information Science, vol. 633, Springer, 2016, pp. 123–135, doi: 10.1007/978-3-319-44817-6.

S17 M. L. Sánchez-Gordón, R.V.O'Connor, R. Colomo-Palacios, and S. Sanchez-Gordon. A learning tool for the ISO/IEC 29110 standard: Understanding the project management of basic profile. Communications in Computer and Information Science, vol. 609, Springer, 2016, pp. 270–283, doi: 10.1007/978-3-319-38980-6_20.

S18 A. Davila, K. Melendez, and M.S.P. Pessoa. University-Enterprise as Educational Space for Students. An Experience in ProCal-ProSer Project. In Proc. of the World Engineering Education Conference (EDUNINE), 2019, pp. 1-6, doi: 10.1109/EDUNINE.2019.8875776.

S19 E. Bonilla-Rivas, M. Munoz, and A. P. P. Negron. Strategy for training in the ISO/IEC 29110 standard based on a serious game. In Proc. of the 10th International Conference on Software Process Improvement (CIMPS), 2021, pp. 74-83, doi: 10.1109/CIMPS54606.2021.9652748.

Appendix B. Primary studies categorized

Id	Year	Paper type	Coverage profile (partial or total)	ISO/IEC 29110 process	Codification: Pedagogical Techniques	Study level	Country of application of the study
S01	2020	Conference Paper	Basic	Project Management Software Implementation	Gamification	undergraduate	Mexico
S02	2020	Conference Paper	Basic	Project Management Software Implementation	POL	undergraduate	Mexico
S03	2020	Journal Article	Basic	Project Management	POL	undergraduate	Ecuador
S04	2020	Journal Article	Entry profile	Project Management Software Implementation	POL	undergraduate	Mexico
S05	2019	Conference Paper	Basic	Project Management Software Implementation	POL	undergraduate	Mexico
S06	2019	Journal Article	Entry profile	Project Management Software Implementation	POL	undergraduate	Colombia
S07	2019	Journal Article	Basic	Project Management Software Implementation	Not apply	undergraduate	Mexico
S08	2018	Conference Paper	Basic	Project Management Software Implementation	Gamification	Not apply	Brazil
S09	2018	Journal Article	Basic	Project Management Software Implementation	POL	undergraduate	Mexico
S10	2018	Conference Paper	Basic	Project Management Software Implementation	Not apply	undergraduate	Mexico
S11	2017	Conference Paper	Basic	Project Management	Gamification	undergraduate and graduate	Spain

S12	2017	Conference Paper	Basic	Project Management	Gamification	undergraduate and graduate	Spain
S13	2017	Conference Paper	Basic	Project Management Software Implementation	POL	undergraduate	Thailand
S14	2017	Conference Paper	Basic	Project Management	POL	undergraduate	France
S15	2016	Journal Article	Basic	Project Management Software Implementation	POL	undergraduate and graduate	Canada
S16	2016	Conference Paper	Basic	Project Management	Gamification	undergraduate	Ecuador
S17	2016	Journal Article	Basic	Project Management	Gamification	undergraduate	Mexico
S18	2021	Conference Paper	Basic	Project Management	Gamification	undergraduate	Colombia
S19	2019	Conference Paper	Basic	Project Management	POL	undergraduate	Peru

Information about authors / Информация об авторах

Luis VIVES, Researcher at Department of Computer Science. Field of scientific interests: statistics, probability theory, computer graphics, data mining, computing in mathematics, natural science, engineering and medicine.

Луис ВИВЕС, научный сотрудник отдела компьютерных наук. Область научных интересов: статистика, теория вероятностей, компьютерная графика, интеллектуальный анализ данных, вычисления в математике, естествознании, технике и медицине.

Karin MELENDEZ, Magister, Researcher. Field of scientific interests: software, quality management, software development, process improvement, quality assurance, process management, performance measurement.

Карин МЕЛЕНДЕС, магистр, исследователь. Область научных интересов: программное обеспечение, управление качеством, разработка программного обеспечения, совершенствование процессов, обеспечение качества, управление процессами, измерение эффективности.

Abraham DÁVILA is a Principal Professor of the Computer Engineering program and is a Doctoral Candidate in Software Engineering, in the field of process improvement. Field of scientific interests: software engineering, software quality process, software quality product, education in software engineering, Innovations based on software.

Авраам ДАВИЛА – профессор программы компьютерной инженерии и докторант в области программной инженерии. Область научных интересов: программная инженерия, процесс качества программного обеспечения, образование в области программной инженерии, инновации на основе программного обеспечения.

DOI: 10.15514/ISPRAS-2023-35(1)-13

Разработка и реализация средства тестирования на устойчивость хранимых данных для приложений, основанных на файловых системах

[1] *Д.К. Родионов, ORCID: 0000-0002-4112-3969 <rodionov.d.k@yandex.ru>*
[1,2,3,4] *С.Д. Кузнецов, ORCID: 0000-0002-8257-028X <kuzloc@ispras.ru>*

[1] *Институт системного программирования им. В.П. Иванникова РАН, 109004, Россия, г. Москва, ул. А. Солженицына, д. 25*
[2] *Московский государственный университет им. М.В. Ломоносова, 119991, Россия, Москва, Ленинские горы, д. 1*
[3] *Московский физико-технический институт, 141700, Россия, Московская область, г. Долгопрудный, Институтский пер., 9*
[4] *НИУ «Высшая школа экономики», 101978, Россия, Москва, ул. Мясницкая, д. 20*

Аннотация. Приложения, работающие с данными, обязаны обеспечивать их надежное хранение. Интерфейсы, доступные для работы с файловой системой, недостаточно специфицированы и требуют высокой квалификации для корректного использования, не приводящего к потере данных пользователей. В рамках данной работы был разработан инструмент, предоставляющий разработчикам возможность тестировать свои приложения и выявлять наиболее распространенные ошибки. Инструмент основан на сборе событий взаимодействия приложения с файловой системой и последующем запуске проверок, способных указать на допущенные ошибки. Инструмент реализует модульную архитектуру, позволяющую расширять доступный набор проверок. Разработанный инструмент был интегрирован в процесс тестирования реализации долговечного журнала, подобного журналу упреждающей записи – компоненту, реализованному во многих системах управления базами данных. Инструмент позволил обнаружить и исправить несколько ошибок, приводящие к возможной потере данных.

Ключевые слова: тестирование; долговечность; файловые системы; io_uring; Rust

Для цитирования: Родионов Д.К., Кузнецов С.Д. Разработка и реализация средства тестирования на устойчивость хранимых данных для приложений, основанных на файловых системах. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 205-222. DOI: 10.15514/ISPRAS–2023–35(1)–13

Design and Implementation of a Tool for Testing Stored Data Durability for Applications Based on File Systems

[1] *D.K. Rodionov, ORCID: 0000-0002-4112-3969 <rodionov.d.k@yandex.ru>*
[1,2,3,4] *S.D. Kuznetsov, ORCID: 0000-0002-8257-028X <kuzloc@ispras.ru>*

[1] *Ivannikov Institute for System Programming of the Russian Academy of Sciences, 25, Alexander Solzhenitsyn st., Moscow, 109004, Russia*
[2] *Lomonosov Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russia*
[3] *Moscow Institute of Physics and Technology (State University), 9 Institutskiy per., Dolgoprudny, Moscow Region, 141700, Russia*
[4] *National Research University Higher School of Economics 20, Myasnitskaya Ulitsa, Moscow, 101978, Russia*

Abstract. Applications that work with data are required to ensure their reliable storage. The interfaces available for working with file systems are not sufficiently specified and require high qualifications for correct use that does not lead to loss of user data. As part of this work, a tool was developed that provides developers with the opportunity to test their applications and identify the most common errors. The tool is based on collecting events from the interaction of the application with the file system and then running checks that can indicate errors. The tool implements a modular architecture that allows you to expand the available set of checks. The developed tool was integrated into the process of testing the implementation of a durable log, similar to the write ahead log, a component implemented in many database management systems. The tool allowed to detect and correct several errors leading to possible data loss.

Keywords: testing; durability; file systems; io_uring; Rust

For citation: Rodionov D.K., Kuznetsov S.D. Design and implementation of a tool for testing stored data durability for applications based on file systems. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 205-222 (in Russian). DOI: 10.15514/ISPRAS-2023-35(1)-13

1. Введение

Долговечность (durability) данных – одно из основных требований к системам хранения данных наряду с согласованностью (consistency) данных. Недавнее исследование [1] показывает, что приложения (в том числе и системы хранения данных) зачастую некорректно взаимодействуют с файловой системой, допуская повреждение данных и нарушая предоставляемые пользователю гарантии. Корректное взаимодействие с файловыми системами не является тривиальной задачей. Обеспечение сохранности данных при отказе системы зависит от многих факторов. Файловые системы, работающие на разных платформах, предоставляют разные гарантии в случае отказа с потерей питания в зависимости от значений конфигурационных параметров [2]. Различия могут скрываться в на первый взгляд похожих интерфейсах.

Примером может служить отличие в работе системного вызова *fsync* в Linux и в Mac OS X. В случае последней вызова *fsync* недостаточно для обеспечения долговечности, требуется установка дополнительного параметра через вызов *fcntl* (`F_FULLFSYNC`) [3]. Более того, некоторые жесткие диски игнорируют команду переноса данных из кеша в основной памяти (*flush*) на постоянный носитель, чтобы показывать себя лучше в тестах производительности [4].

Еще один пример – отличие семантики системного вызова *fdatasync* в Linux [5] и FreeBSD [6]. *fdatasync* в Linux гарантирует обновление размера файла в метаданных, а во FreeBSD такой гарантии нет. Таким образом в случае потери питания операция записи, которая увеличивает размер файла, может быть потеряна. Данные могут быть успешно записаны, но из-за потери обновления размера файла могут стать недоступными для операций чтения.

1.2 Мотивация работы

Семантика гарантий, предоставляемых файловыми системами, как правило, не описана в виде формальной модели, что приводит к спорам о том, как трактовать те или иные положения, указанные в стандартах (например, POSIX) и документации. Возникает необходимость проверять операционную систему и приложения на взаимную корректность реализации и использования интерфейсов файловой системы.

Примером такой проблемы является обнаруженная разработчиками PostgreSQL особенность реализации системного вызова *fsync* в Linux и некоторых других операционных системах, приводящая к потере данных [7]. Разработчиками было обнаружено неочевидное, интуитивно непонятное поведение системного вызова *fsync*. Проблема возникает в случае повторения системного вызова после завершения предыдущей попытки с ошибкой ввода вывода (`EIO`). В этом случае страницы, находящиеся в кеше основной памяти и помеченные как «грязные», помечаются как чистые, несмотря на возникшую ошибку, и ошибка не сохраняется для последующих вызовов. Таким образом повторная попытка всегда оказывается успешной, поскольку система считает, что все изменения уже синхронизированы с диском. Такое поведение привело к потере данных. Проблема усугублялась тем фактом, что в некоторых случаях произошедшая ошибка могла быть недостижима для пользовательской программы. Исправление, гарантирующее возможность пользовательской программы узнать об ошибке, было внедрено в версии 4.13 [8]. Тестирование программ, критически зависящих от корректного взаимодействия с файловой системой, также затруднено в силу недетерминизма операционных систем. Традиционные методы тестирования могут показать наличие проблем, но не могут доказать их отсутствие. Выявлять проблемы помогает рандомизированное тестирование. Примером использования рандомизированного подхода для проверки долговечности может служить запуск тестовой программы со случайными отключениями питания во время работы. Такой метод помог выявить проблему потери данных в PostgreSQL [9]. В процессе тестирования выполнялось физическое отключение питания. Найденная проблема заключалась в некорректной последовательности действий при переименовании файла, а именно, отсутствии вызова *fsync* для синхронизации родительской директории.

1.2 Суть предлагаемого подхода

В данной работе предлагается инструмент динамического анализа программ реализующий набор методов для обнаружения ошибок в логике взаимодействия программы с файловой системой. Разработанные компоненты были применены для поиска ошибок в реализации долговечного журнала и позволили обнаружить несколько ошибок, приводящих к потере данных. Инструмент реализует пессимистичную модель долговечности и таким образом позволяет выявлять ошибки несоответствия кода приложения заложенной модели. В представленной реализации модель интегрируется в приложение, реализуя метод «белого ящика». Этот подход был выбран в виду его простоты и желания развивать инструмент как часть платформы симуляционного тестирования. Модульная архитектура инструмента позволяет реализовать другие варианты интеграции с пользовательской программой. Интеграция непосредственно в приложение позволяет избавиться от взаимодействия с реальной файловой системой при выполнении тестовых сценариев, что позволяет сократить время необходимое для их запуска. Дополнительно, преимуществом такого подхода является возможность в некоторых случаях отвязать инфраструктуру тестирования от платформы, на которой разрабатывается приложение, позволяя, таким образом, разрабатывать и тестировать приложение, например, в среде операционной системы Windows, но используя модель долговечности, соответствующую Linux.

Однако имеются и недостатки. Основным недостатком является необходимость модификации кода приложения. Дополнительно, при использовании сторонних библиотек не

всегда есть возможность подменить реализацию функций, выполняющих непосредственное взаимодействие с файловой системой. Поэтому переиспользование готовых библиотек может быть ограничено. Так же, так как анализ является динамический необходимо принимать во внимание метрику покрытия кода приложения тестовыми сценариями.

Разработанный инструмент фокусируется на удобстве использования разработчиком, слабо осведомленном о деталях реализации файловых систем и особенностей во взаимодействии с ними. Выдаваемая информация об обнаруженной ошибке включает в себя конкретное место в исходном коде, где была инициирована операция ввода-вывода, и набор рекомендаций по решению проблемы.

Инструмент требует от разработчика приложения использования подмененных методов для взаимодействия с файловой системой. Поскольку сигнатура этих методов полностью соответствуют сигнатуре, предлагаемой стандартной библиотекой, переписывание кода для этого не требуется. При помощи специализированных методов разработчик запрашивает проверку у модели и указывает файл или его часть, которая по логике программы должна быть долговечно записана. Если модель обнаруживает несоответствие, пользователю предоставляется отчет об ошибке.

```
fn write(f: &File, data: &[u8]) -> io::Result<()> {
      f.write_at(0, data)?;
      f.ensure_durable(0..data.len())
}
```

Листинг 1. Пример фрагмента кода, вызывающего проверку соответствия требованиям долговечности

Listing 1. Example fragment of code that calls the satisfaction check of durability requirements

Например, выполнение кода, представленного на листинге 1, приведет к ошибке: Файл не синхронизирован, незавершенная операция записи в диапазоне `<0, data.len()>`.

Дальнейший материал статьи организован следующим образом. В разд. 2 описываются архитектура описываемого инструмента и возможные способы ее реализации. Разд. 3 посвящен обсуждению используемой модели долговечности. В четвертом разделе описываются первые результаты применения разработанного инструмента. Пятый раздел посвящен краткому анализу близких по тематике работ. Наконец, шестой раздел статьи содержит заключение.

2. Архитектура инструмента

В разработанном инструменте используется модульная архитектура. Для работы необходимо наличие двух основных компонентов: модуля сбора событий взаимодействия с файловой системой и реализация модели долговечности, анализирующая собранные события. Модульная архитектура позволяет использовать разные реализации компонентов, т.е. использовать разные модули сбора событий с разными моделями долговечности.

2.1. Сбор событий взаимодействия

Источником данных для анализа являются события взаимодействия приложения с файловой системой. Примерами событий являются `Write` – событие, представляющее операцию записи или `Fsync` – запрос на синхронизацию изменений с диском.

Сбор событий может быть реализован несколькими способами, каждый из которых имеет свои преимущества и недостатки.

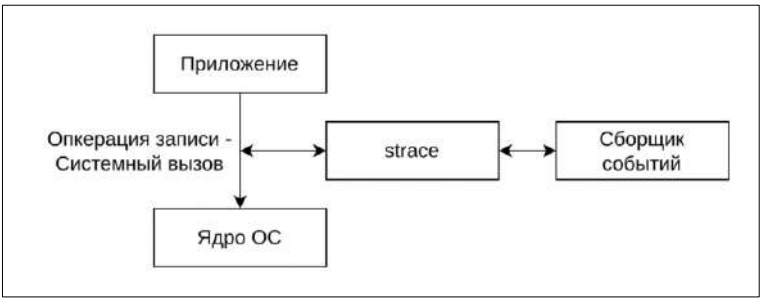

Рис. 1. Сбор событий ввода-вывода с использованием strace
Fig. 1. Collecting I/O events using strace

2.1.1 Перехват системных вызовов

Самым простым в реализации является вариант с перехватом системных вызовов при помощи утилиты *strace* [10] (рис. 1). *strace* - утилита, основанная на системном вызове ptrace [11]. Преимуществом данного метода является отсутствие необходимости модифицировать программу. В то же время в ALICE потребовалась модификация самой утилиты *strace* для получения дополнительного контекста.

Одним из недостатков данного подхода является зависимость от платформы: ptrace доступен в Linux, FreeBSD [12] / OpenBSD [13], RedoxOS[14]. Альтернативой *strace* на некоторых платформах может служить утилита *dtrace* [15]. Другим недостатком является зависимость от конкретного способа осуществления взаимодействия с файловой системой – использования системных вызовов. В случае использования отображения файлов в виртуальную память посредством *mmap* операции ввода-вывода не будут перехвачены утилитой *strace*. Эту проблему можно решить, в версии ALICE использованной в статье эта проблема решена, но поддержка отсутствует в опубликованном коде фреймворка. Вероятнее всего для этого используется механизм userfaultfd [16].

Кроме того, *strace* не позволяет обрабатывать операции ввода-вывода, осуществляемые с помощью *io_uring* [17] – механизма ядра Linux, реализующего неблокирующие операции ввода-вывода.

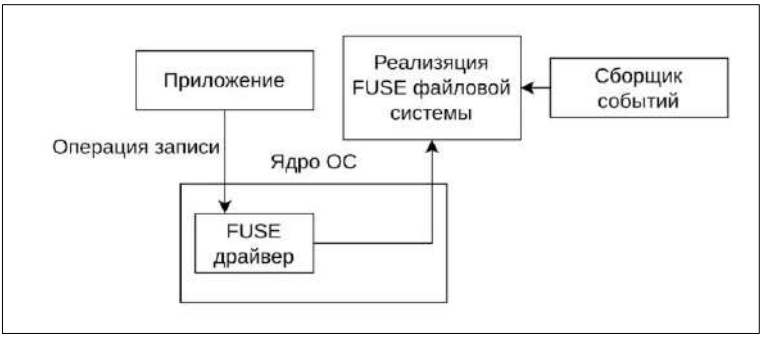

Рис. 2. Сбор событий ввода-вывода с использованием FUSE
Fig. 2. Collecting I/O events using FUSE

2.1.2 Использование механизма FUSE

Другим вариантом является использование механизма *FUSE* (Filesystem in Userspace), позволяющего непривилегированным пользователям создавать собственные файловые системы, не затрагивая коды ядра ОС. В этом случае в нашем инструменте можно было бы реализовать свою файловую систему, которая могла бы регистрировать все проходящие

через нее операции и внедрять ошибки для расширения тестирования (рис. 2). Этот вариант также не требует модификаций исходного кода приложения и позволяет захватывать события, поступающие из *io_uring*.

Недостатком использования *FUSE* подхода является невысокая производительность. По замерам производительности в зависимости от рабочей нагрузки использование *FUSE* может приводить к снижению производительности до 5 раз по сравнению с использованием файловых систем, реализованных в пространстве ядра [18].

К минусам этого подхода можно также отнести отсутствие мультиплатформенной поддержки. Чаще всего библиотеками реализуется поддержка *FUSE* в ядре Linux [19]. Использование других платформ может требовать доработок. Для MacOSX аналогичный набор возможностей поддерживается проектом macFuse [20]. Для Windows поддерживается проект Windows File System Proxy [21]. Поддержка всех трех платформ требует дополнительных трудозатрат. В экосистемах языков программирования может не быть библиотеки, взявшей на себя работу по унификации всех трех проектов в одном интерфейсе. Таким образом в случае необходимости кросс платформенной поддержки *FUSE*-подобной технологии разработчику потребуется реализовать общий интерфейс, учитывающий особенности реализации *FUSE* на всех трех платформах. В некоторых ситуациях удобство использования *FUSE* перевешивает этот недостаток. Например, инструмент unreliablefs [22] реализует поддержку *FUSE*.

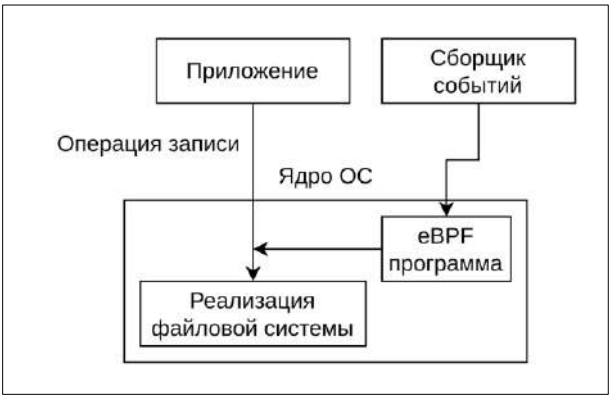

Рис. 3. Сбор событий ввода-вывода с использованием eBPF
Fig. 3. Collecting I/O events using eBPF

2.1.3 Использование возможностей механизмов ядра Linux

Еще одним способом является вариант использовать возможности механизмов ядра Linux – *eBPF* [23] и *tracepoints* [24]. Поддержка eBPF также начала появляться в Windows [25], но на данный момент применение ограничено областью сетевого взаимодействия. В Linux *eBPF* в сочетании с *tracepoints* позволяют извлекать информацию о взаимодействии с файловой системой непосредственно из контекста самого ядра (рис. 3). Гибкость *eBPF* обеспечивается возможностью присоединять пользовательский обработчик ко многим функциям ядра, например, к функции *ext4_file_write_iter*, отвечающей за запись данных на диск в файловой системе *ext4*. *Tracepoints* позволяют захватывать событие, содержащее в себе атрибуты, специфичные для конкретной точки трассировки. Например, для *io_uring_complete* будут переданы ссылки на контекст *io_uring*, на запрос и ответ, и на данные пользователя (листинг 2).

```
/**
 * io_uring_complete - called when completing an SQE
 *
```

```
 * @ctx:      pointer to a ring context structure
 * @req:      pointer to a submitted request
 * @user_data:     user data associated with the request
 * @res:      result of the request
 * @cflags:     completion flags
 * @extra1:     extra 64-bit data for CQE32
 * @extra2:     extra 64-bit data for CQE32
 *
 */
```

Листинг 2. Сигнатура события io_uring_complete [26]
Listing 2. Signature of the io_uring_complete event [26]

Данные, собранные пользовательским обработчиком, передаются в пространство пользователя и на этом этапе готовы для анализа.

Точки трассировки заранее расставлены на пути основных операций, релевантных для инструмента. Например, файловая система xfs в 6 версии ядра Linux предоставляет возможность анализировать события из 600 точек трассировки. Новые точки трассировки добавляются по мере необходимости в последующие версии ядра.

Плюсом данного подхода является потенциальная независимость от способа, которым осуществляются операции ввода вывода. Непосредственный ли это системный вызов, или операция io_uring – вне зависимости от интерфейса вызов доходит до уровня абстракции файловой системы и здесь может быть зарегистрирован для последующего анализа. Такой подход достаточно эффективен с точки зрения оптимизации накладных расходов. Он дает возможность использовать эту технологию для анализа непосредственно на основе инфраструктуры, обслуживающей пользовательские запросы. Поскольку данные о событиях ввода вывода берутся непосредственно с уровня ядра операционной системы, модификация кода приложения для сбора данных не требуется.

Минусами данного подхода являются привязка к конкретной операционной системе (Linux), а также необходимость писать eBPF-подпрограммы и связывать их для доставки данных инструменту для анализа. Кроме того, для привязки eBPF-программ необходимы права суперпользователя (администратора), что в свою очередь несколько усложняет процесс тестирования.

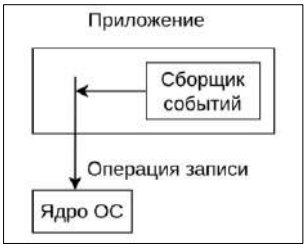

Рис. 4. Сбор событий ввода-вывода компонентом тестовой инфраструктуры приложения
Fig. 4. Collection of I/O events by the component of application test infrastructure

2.1.4 Сбор данных на уровне приложения

Наконец, возможным способом перехвата операций ввода-вывода является отказ от идеи «черного ящика» и реализация сбора данных непосредственно на уровне приложения (рис. 4). Данный подход позволяет не зависеть от платформы, на которой осуществляется тестирование. Таким образом, появляется возможность тестировать приложение, сопоставляя его с моделью, проверяющей программу на работу в среде Linux, но запускать тесты на другой платформе, например, Windows или MacOS. На данный момент в разработанном инструменте существует ряд ограничений, так как инструмент все равно

выполняет запрошенные операции на реальной файловой системе само приложение (или его модуль релевантный для описанного вида тестирования) должно иметь возможность запуска на альтернативной платформе.

Минусом данного подхода является сложность переиспользования такого решения, так как на данный момент для интеграции с приложением необходимо чтобы оно было написано на том же языке программирования, в дополнение к этому, требуется изменение кода приложения, т.е. инструмент не сможет захватывать события произвольного приложения. Основная проблема заключается в необходимости модифицировать не только код, написанный непосредственно авторами приложения, но и код используемых библиотек, если они содержат вызовы операций ввода-вывода. Также стоит отметить, что такой подход позволяет полностью отказаться от использования реальной файловой системы в тестах. Библиотека для сбора данных может полностью эмулировать операции файловой системы, что позволяет ускорить тестирование, поскольку все операции будут осуществляться в том же процессе, в котором выполняется приложение, и будут использовать виртуальную память для хранения данных. При использовании рандомизированного тестирования скорость выполнения тестов оказывает серьезное влияние на применимость метода, так как увеличение времени выполнения одного теста может привести к кратному увеличению времени запуска всех тестовых сценариев, что также увеличивает затраты на инфраструктуру тестирования.

В дополнение, этот подход является шагом к возможности применения симуляционного тестирования – методики, являющейся вариантом случайного тестирования, при которой выполнение кода приложения не зависит от случайных факторов, таких как время или расписание планировщика потоков. Такое поведение позволяет манипулировать средой выполнения и контролируемо вносить изменения в симулированную внешнюю среду случайным образом генерируя задержки, переупорядочивая сообщения, изменяя расписание выполнения задач. Подобный подход успешно применяется в FoundationDB [27]. Также этот подход может применяться на уровне всей операционной системы, а не отдельного приложения [28]. Важно отметить, что одного только абстрагирования файловой системы недостаточно для применения симуляционного тестирования. Для этого также необходимо избегать остальных источников недетерменизма в программе, таких как время, вышеупомянутое влияние планировщика потоков ОС, и т.д.

Таким образом, многообразие описанных подходов позволяет специализировать инструмент для различных сценариев использования. Для тестирования по методологии черного ящика и анализа производительности наиболее универсальным является подход, основанный на трассировке ядра и eBPF-программах. Для нужд рандомизированного и симуляционного тестирования лучше всего подходит вариант реализации сбора данных непосредственно в приложении.

Поскольку подход сбора данных на уровне приложения является достаточно простым в реализации и подходит для развития в направлении симуляционного тестирования, он был выбран первым для реализации в инструменте. Другие подходы или их комбинации также могут быть реализованы и применены в рамках разработанного инструмента, но эта работа не является частью данного исследования.

2.2 Интеграция с модулем сбора данных

Для реализации инструмента и целевого приложения был выбран язык программирования Rust [29]. Это системный язык программирования, основной отличительной особенностью которого является специализированный механизм проверки заимствований (borrowing check), позволяющий избежать типичных для системных языков проблем с безопасностью доступа к памяти. Механизм отслеживания заимствований позволяет избежать таких

ошибок, как использование после освобождения (use after free), двойное освобождение (double free), висячие ссылки (dangling pointer).

Целевое приложение использует фреймворк *glommio* [30], реализующий примитивы архитектуры *thread-per-core*, основной идеей которой является отказ от неявной коммуникации между потоками, запущенными на разных ядрах процессора. В *glommio* упор делается на хранение данных локально для каждого ядра, что позволяет снизить накладные расходы на переключения контекста и избежать блокировок для синхронизации доступа к данным. Основным примитивом для конкурентного выполнения кода в *glommio* являются сопрограммы, выполняемые всегда в рамках одного потока ОС. Эта особенность позволяет упростить внедрение симуляционного тестирования.

Для упрощения интеграции инструмент предоставляет абстракцию адаптера. Адаптер является отдельной сущностью, отвечающей за предоставление интерфейса максимально близкого тому, который представляется библиотекой, используемый в приложении для дискового ввода-вывода. Модульная архитектура позволяет реализовать адаптеры как для функций стандартной библиотеки, так и для различных фреймворков, реализующих модель неблокирующего ввода-вывода. Добавление новых адаптеров не требует изменения остальных частей инструмента. Важно отметить, что адаптеры также необходимо тестировать т. к. ошибки в них могут приводить к некорректной работе инструмента.

Каждый адаптер параметризуется обобщенным типом, реализующим типаж (trait) `Instrument` (листинг 3).

```
pub trait Instrument {
      type Error: std::error::Error;

      fn apply_event(&self, event: Event) -> Result<(), Self::Error>;
}
```

Листинг 3. Типаж Instrument
Listing 3. Instrument trait

Типаж состоит из одной функции `apply_event` принимающей событие ввода-вывода, описываемое типом `Event` (листинг 4).

```
#[derive(Debug, Clone)]
pub enum Event {
    // To associate path with the fd
    Open(PathBuf, i32),
    // dirties file
    Write(WriteEvent),
    // directly sets max_written_pos for a file
    // disgards write events past specified size
    SetLen(i32, u64),
    // clears pending changes
    Fsync(i32),
    ...
    EnsureFileDurable {
        target: EitherPathOrFd,
        up_to: Option<u64>,
    },
    ...
}
```

Листинг 4. Тип Event
Fig. 4. Event Type

Объект, реализующий типаж `Instrument`, получает события ввода вывода, сгенерированные приложением на основе осуществляемых операций ввода вывода. С применением этого механизма адаптер параметризуется реализацией модели долговечности, что позволяет запускать одни и те же тесты с разными реализациями моделей долговечности или другими анализаторами.

Рассмотрим модуль-адаптер для фреймворка *glommio*. Модуль представляет две структуры: `InstrumentedDirectory` и `InstrumentedDmaFile`. Их интерфейс повторяет интерфейс, предоставляемый соответствующими типами *glommio* (листинг 5).

```
use glommio::io::DmaFile;

pub struct InstrumentedDmaFile<I: Instrument + Clone> {
    file: DmaFile,
    instrument: I,
}
```

Листинг 5. Тип InstrumentedDmaFile
Listing 5. InstrumentedDmaFile Type

Тип `InstrumentedDmaFile` оборачивает тип `DmaFile` и отправляет события ввода вывода в переданный тип `Instrument`. Рассмотрим передачу события записи данных.

```
pub async fn write_at(&self, buf: DmaBuffer, pos: u64) -> Result<usize> {
      self.instrument
            .apply_event(Event::Write(WriteEvent {
                  fd: self.file.as_raw_fd(),
                  file_range: FileRange {
                  start: pos,
                  end: pos + buf.len() as u64,
                  },
            }))
            .unwrap();

      self.file.write_at(buf, pos).await
}
```

Листинг 6. Реализация write_at
Linting 6. Implementation of write_at

Запись данных осуществляется методом `write_at` (листинг 6). Адаптер генерирует событие записи данных и передает управление оригинальной функции `write_at`.

Таким образом, `InstrumentedDmaFile` параметризован типом, реализующим типаж `Instrument`. Это позволяет использовать разные анализаторы в разных тестах и выбирать специальную – пустую реализацию при сборке приложения. Такая архитектура также оставляет возможность реализовать альтернативные способы сбора данных, рассмотренные ранее. В этом случае реализацию типажа `Instrument` изменять не придется.

3. Модель долговечности

Модель долговечности – это часть инструмента, отвечающая за обработку потока событий ввода-вывода от приложения. Модель реализует семантику долговечности файловой системы и отвечает за реализацию запросов на проверку надежности записи тех или иных изменений. Предлагаемая модель основана на семантике файловых систем ОС Linux и поддерживает некоторые особенности поведения FreeBSD. Модель не является исчерпывающей, т.е. не позволяет доказать отсутствие проблем, но позволяет успешно

показать наличие определенного класса ошибок. Основная цель модели – автоматизировать отслеживание наиболее известных проблем. Модель не может заменить тесты или верификацию.

Модель построена на отслеживании изменений, не синхронизированных с диском, для каждого используемого программой пути на файловой системе. Таким образом, модель получает события записи от приложения и отвечает на запросы о долговечности записи изменений в указанном диапазоне файла. Стоит отметить, что на данный момент символические и жесткие ссылки не поддерживаются. Это не является ограничением архитектуры инструмента, в дальнейшем поддержка может быть добавлена.

Далее рассматриваются характеристики модели долговечности на примере записи данных в файл. Обозначается последовательность действий, которую модель считает безопасной.

```
let f = InstrumentedDmaFile::create(fname, checker)
      .await
      .expect("create failed");

let buf = f.alloc_dma_buffer(512);

f.write_at(buf, 0).await.expect("write failed");
f.ensure_durable(0..buf.len());
```

Листинг 7. Запись в файл без вызова fsync
Listing 7. Writing to a file without calling fsync

Листинг 7 демонстрирует пример записи в файл без синхронизации записанных изменений. Вызов `ensure_durable` в данном случае вызовет панику и завершит тест со сообщением об ошибке, показанным на листинге 8.

```
Error: Durability constraint violation: File has pending changes.
Max synced position: 0
Horizon: max written pos 511
Pending changes:
    [0..511] earlier than max written pos 511
Call fsync or fdatasync to synchronize them
Max durable pos 0 != up to 511
```

Листинг 8. Сообщение об ошибке
Listing 8. Error message

Таким образом модель указывает на необходимость синхронизации изменений файла с диском.

Последовательность действий с листинга 9 приведет к ошибке, указанной на листинге 10.

```
let f = InstrumentedDmaFile::create(fname, checker)
      .await
      .expect("create failed");

let buf = f.alloc_dma_buffer(512);

f.write_at(buf, 0).await.expect("write failed");
f.fdatasync().await.expect("sync failed");
f.ensure_durable(None);
```

Листинг 9. Запись в файл без синхронизации родительской директории
Listing 9. Writing to a file without synchronizing the parent directory

```
Error: Durability constraint violation:
File parent directory /test_data/adapter_without_fsync is not synchronized
to disk,
synchronize it via fsync or fdatasync to fix the the problem.
```

Листинг 10. Сообщение об ошибке
Listing 10. Error message

Таким образом модель указывает на необходимость синхронизации родительской директории после создания файла для того, чтобы наличие файла в директории было синхронизировано с диском. В противном случае при потере питания запись о наличии файла в директории может быть потеряна.

После добавления вызова *fdatasync* для родительской директории файла тест завершается успешно.

Модель также учитывает наличие различий в семантике системного вызова *fdatasync* между Linux и FreeBSD. Основное отличие *fdatasync* от *fsync* заключается в том, что *fdatasync* не синхронизирует метаданные файла, например, дату последнего изменения. Размер файла является частью метаданных, а потеря обновления размера может привести к потере данных, в случае, когда операция записи увеличила размер файла. В Linux *fdatasync* гарантирует обновление размера файла, в то время как во FreeBsd такой гарантии нет. По умолчанию используется более строгая семантика FreeBSD, но доступна настройка, позволяющая переключиться на использование семантики Linux.

4. Начальные результаты использования

Разработанный инструментарий был интегрирован в процесс тестирования реализации долговечного журнала. Журнал основан на библиотеке *glommio*, использующей *io_uring* для операций ввода-вывода.

Журнал реализован в виде библиотеки для последующего использования в качестве компонента другой системы. Библиотека предоставляет два типа для работы с журналом: читатель и писатель соответственно. Проверка свойств долговечности необходима в контексте объекта-писателя, поскольку операции чтения журнала не изменяют его, т.е. не могут привести к потере данных.

Запись журнала выполняется посегментно, каждый сегмент – это файл, размер которого устанавливается при инициализации писателя. Запись каждого сегмента выполняется блоками фиксированного размера. Каждый блок записывается на диск либо в случае заполнения, либо по наступлении временной отсечки.

С точки зрения анализа долговечности точками интереса являются запись блока в файл сегмента и переключение на новый сегмент.

При помощи инструмента удалось обнаружить три ошибки.

Ошибка 1: *Error: Durability constraint violation: File has no pending changes, but it wasnt synced after call to `create`*.

Первая ошибка заключается в отсутствии синхронизации после создания файла журнала до последующих операций записи (листинг 11).

```
1: segment_file = dir.open(segment_file_name)
2: dir.sync()
```

Листинг 11. Алгоритм создания файла сегмента, содержащий ошибку
Listing 11. The erroneous algorithm for creating a segment file

Данная ошибка на первый взгляд не кажется критичной, так как пустой файл вроде бы не несет в себе ценности. Тем не менее, в данном случае вызов *fdatasync* необходим, поскольку обеспечивает сериализацию последовательности между созданием файла и синхронизацией

родительской директории. Отсутствие синхронизации может привести к переупорядочиванию этих операций на уровне файловой системы, т.е. состояние директории, синхронизированное с диском, может не включать созданный файл [2].

Таким образом, данный вызов все же необходим для предотвращения возможной потери данных.

Сообщение об ошибке 2: *Error: Durability constraint violation: File parent directory /test_data/read_write_many_segments is not synchronized to disk, synchronize it via fsync or fdatasync to fix the problem*.

В данном случае инструментом было обнаружено отсутствие синхронизации родительской директории при создании файла для первого сегмента журнала. Ошибка серьезная, без синхронизации родительской директории при потере питания файловой системой не гарантируется сохранение директории. Следственно, при отсутствии директории файлы, созданные в ней, также будут недостижимы [2].

Сообщение об ошибке 3 *Error: Durability constraint violation: File has pending changes. Max synced position: 0 Horizon: up to 2047 Max durable pos 0 != up to 2047*. *The error is applicable to FreeBSD semantics of fdatasync system call and is not applicable to Linux*

Эта ошибка относится к проверке на соответствие семантике FreeBSD. При записи в журнал для синхронизации изменений используется системный вызов *fdatasync*. На платформе FreeBsd вызов *fdatasync* не гарантирует обновление размера файла в метаданных файловой системы. Таким образом, возможна ситуация, когда запись увеличила размер файла, и данные были записаны, но размер файла не обновился. При восстановлении после отказа будет прочитан устаревший размер файла и новые данные будут потеряны.

Ошибка серьезная, может привести к потере данных. Однако в этом случае библиотека не рассчитана под работу в системе FreeBSD и не сможет на ней работать из-за использования *io_uring*, механизма, доступного только в Linux.

5. Родственные работы

В индустрии для решения задачи поиска ошибок взаимодействия с файловой системой применяется рандомизированное тестирование в сочетании с искусственным внесением ошибок [31, 32]. Инструмент, созданный в рамках данной работы, не заменяет этот подход, но дополняет его. Наш инструмент позволяет проще находить более очевидные ошибки. Для этого достаточно реализовать модульные тесты и запустить их с включенной инструментацией. Рандомизированное тестирование может помочь улучшить покрытие и найти дополнительные ошибки.

Как утверждает анализ взаимодействия работы приложений с файловыми системами [2], зачастую логика протокола долговечности находится в разных файлах, и поэтому трудно убедиться в том, что все необходимые действия, обеспечивающие долговечность, действительно выполнены в тот момент, когда система возвращает пользователю подтверждение успешного выполнения запроса. Разработанный инструмент позволяет записывать информацию и проверять ее на соответствие модели. Еще одним преимуществом использования модели белого ящика является гибкость, так как приложение напрямую использует библиотеку инструмента для работы с файловой системой, что упрощает возможное расширение инструмента другими режимами проверки; примером может быть поддержка возможности внедрения случайных ошибок. В случае модели черного ящика для этих задач скорее всего пришлось бы использовать платформенно-зависимые инструменты, такие как *strace* (например, для записи системных вызовов и подмены кодов возврата) или реализация своей обертки для файловой системы в пространстве пользователя с применением *FUSE* как для записи активности приложения, так и для внедрения ошибок.

Исследователями предложены и другие подходы. Наиболее близким по принципу работы является инструмент *Application-Level Intelligent Crash Explorer* (ALICE) [2]. Данный

инструмент ограничен работой под ОС Linux и реализует принцип черного ящика, собирая все системные вызовы, сделанные приложением при выполнении тестовой рабочей нагрузки при помощи модифицированной версии инструмента *strace*, чтобы на втором этапе при помощи модели долговечности сгенерировать все возможные состояния файловой системы в случае отказа и запустить специально разработанные тестовые сценарии для проверки сохранения инвариантов приложения при восстановлении после отказа.

Данный инструмент имеет преимущество в отсутствии необходимости модифицировать приложение для записи необходимой информации, но поскольку используется платформенно- зависимый метод сбора этой информации, приложение необходимо разрабатывать и тестировать на целевой ОС – в данном случае Linux. В случае использования *io_uring* метод сбора данных посредством мониторинга системных вызовов также перестает работать, так как интерфейс *io_uring* стремится минимизировать количество системных вызовов и соответственно не использует их для выполнения отдельных операций. Кроме того, наш инструмент позволяет привязывать возникающие ошибки к исходному коду программы.

Другим направлением развития этого подхода с углублением в направлении применения методов формальной верификации является работа исследователей из университета Вашингтона [33]. Основной идеей является применение методов, используемых для разработки и верификации моделей памяти, проводимой для уточнения семантики многопоточных программ. Работа описывает набор инструментов, позволяющий формально верифицировать соответствие приложения модели долговечности определенной файловой системы, которая также синтезируется в рамках исследования. Отличительной особенностью разработанного в [33] инструмента является возможность синтезировать минимальный набор барьеров (например, вызовов *fsync*) для того, чтобы модель посчитала программу корректной.

Для реализации используется генерация контрпримеров инструментом проверки моделей (model-checking). Предлагаемый инструментарий может гарантировать соответствие программы модели долговечности файловой системы. Однако для применения инструмента необходимо использовать верификатор и специализированный язык моделирования, что повышает порог входа и усложняет широкое распространение инструмента среди разработчиков приложений. В свою очередь подход ALICE [2] проще и требует от разработчиков меньше специализированных знаний.

Оба подхода [2, 33] используют похожие эмпирические методы синтезирования спецификации популярных файловых систем (таких как ext4) исследуя последовательность дисковых операций, получившихся в результате той или иной рабочей нагрузки, консультируясь с разработчиками, изучая документацию. Достоверные спецификации файловых систем являются необходимым базовым блоком для разработки инструментов, проверяющих корректность приложений.

Также была предложена реализация верифицированной файловой системы [34]. Данный подход позволяет избежать ошибок в коде самой файловой системы и позволяет формально определить модель взаимодействия приложений с файловой системой. Однако, несмотря на высочайшую степень предоставляемых гарантий, данный подход имеет ряд недостатков.

Широкому распространению этого метода может препятствовать зависимость приложений от наличия конкретной файловой системы на компьютере пользователя. Это усложняет непростую задачу написания и доставки до пользователей кроссплатформенных приложений, так как файловая система зачастую является компонентом ядра ОС, а альтернативные решения (такие, как FUSE в Linux) также специфичны для каждой платформы, и их использование может быть сопряжено со снижением производительности. Даже при решении вышеупомянутых проблем реализованное таким образом приложение может требовать более высоких привилегий при установке и наладке специализированной файловой системы.

6. Заключение и будущие направления исследований

Разработанный инструмент и его модель долговечности показали свою практическую пригодность, выявив три ошибки, потенциально приводящие к потере данных. Инструмент легко интегрируется в процесс разработки, требования к изменению кода минимальны. Гибкость инструмента позволяет добавлять новые механизмы проверок, что позволит расширить список обнаруживаемых классов ошибок.

Реализованный подход сбора данных накладывает серьезные ограничения на внедрение инструмента в существующие проекты, которые не были разработаны с нуля, опираясь на абстракции, представленные инструментом. Реализация альтернативных способов сбора данных позволит расширить область применимости инструмента.

В дальнейшем разработанный инструмент может быть использован как часть платформы симуляционного тестирования. Такой подход требует больше изменений в приложении, но позволяет значительно расширить возможности тестирования, включающие в себя детерминизм (гарантированная воспроизводимость) ошибок. Также это поможет достичь ускорения тестирования за счет перехода к полной симуляции ввода вывода.

Еще одной потенциальной возможностью может стать реализация проверки, подобной инструменту ALICE. Процесс состоит из записи рабочей нагрузки и генерирования потенциальных снимков состояния файловой системы в случае отказа с последующей проверкой пользовательского инварианта. В таком случае, если система сообщила пользователю, что данные были записаны, но не может их прочитать после восстановления, будет сгенерирована ошибка.

Модель долговечности также может быть усовершенствована. На данный момент моделью представлена некая абстрактная POSIX-совместимая файловая система. Выделение моделей для конкретных файловых систем может помочь приложениям, применяющим оптимизации, заточенные под конкретную файловую систему. Сама модель может быть также расширена дополнительными проверками для обнаружения большего количества ошибок. Например, на данный момент модель неявно предполагает линеаризуемость операций, т е что одна операция заканчивается до начала следующей. Это упрощение не является истиной в конкурентных программах. Выделение событий начала и конца операции позволит обнаруживать гонки данных (data races), случаи, когда операции одновременно модифицируют один участок файла. Так же это приводит к ситуации, описанной в листенге 12.

```
T1: write(0, 512)
T1: fsync
T2: write(0, 256)
T1: ensure_durable
```

Листинг 12. Последовательность действий двух сопрограмм, приводящих к ошибке
Listing 12. The sequence of actions of two coroutines leading to confusing error

В данном случае сопрограмма T1 получит ошибку при вызове ensure_durable, но «виновником» является сопрограмма T2. Для упрощения отладки подобных ситуаций необходимо расширение возможностей инструмента.

Описанная проблема решается применением методик из области спецификации и верификации моделей памяти. Для борьбы с похожими проблемами в отношении основной памяти применяются так называемые санитайзеры (ASAN [35], TSAN [36], KASAN [37]). В работе [33] авторы применяют подходы, разработанные для поиска ошибок доступа к основной памяти для поиска ошибок доступа к файлам. Таким образом, данный аспект является еще одним местом для потенциального внедрения улучшений, повышающих эффективность инструмента.

Исходный код инструмента доступен по ссылке [38].

Список литературы / References

[1]. Rebello A., Patel Y. et al. Can applications recover from fsync failures? ACM Transactions on Storage (TOS), vol. 17, issue 2, 2021, article no. 12, 30 p.

[2]. Pillai T.S., Chidambaram V. et al. All file systems are not created equal: On the complexity of crafting crash-consistent applications. In Proc. of the 11th USENIX Symposium on Operating Systems Design and Implementation, 2014, pp. 433-448.

[3]. Mac OS X Manual Page for fsync(2). Available at: https://developer.apple.com/library/archive/documentation/System/Conceptual/ManPages_iPhoneOS/man2/fsync.2.html, accessed 09.04.

[4]. Rajimwale A., Chidambaram V. et al. Coerced Cache Eviction and discreet mode journaling: Dealing with misbehaving disks. In Proc. of the IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN), 2011, pp. 518-529.

[5]. Linux man pages: fdatasync(2). Available at: https://linux.die.net/man/2/fdatasync, accessed 28.01.2023.

[6]. FreeBSD Manual Pages: fdatasync(2). Available at: https://man.freebsd.org/cgi/man.cgi?query=fdatasync&sektion=2, accessed 28.01.2023.

[7]. PostgreSQL's handling of fsync() errors is unsafe and risks data loss at least on XFS. Available at: https://postgrespro.ru/list/thread-id/2379543, accessed 28.01.2023.

[8]. Fsync Errors - PostgreSQL wiki (https://wiki.postgresql.org/wiki/Fsync_Errors#Open_source_kernels), accessed 09.04.2023

[9]. silent data loss with ext4 / all current versions. Available at: https://www.postgresql.org/message-id/56583BDD.9060302@2ndquadrant.com, accessed 28.01.2023.

[10]. strace. Available at: https://strace.io/, accessed 28.01.2023.

[11]. ptrace(2) - Linux manual page. Available at: https://man7.org/linux/man-pages/man2/ptrace.2.html, accessed 09.04.2023.

[12]. ptrace - FreeBSD Manual Pages. Available at: https://man.freebsd.org/cgi/man.cgi?query=ptrace, accessed 09.04.2023.

[13]. ptrace(2) - OpenBSD manual pages. Available at: https://man.openbsd.org/ptrace.2, accessed 09.04.2023.

[14]. RSoC: Implementing ptrace for Redox OS - part 5 - Redox - Your Next(Gen) OS. Available at: https://www.redox-os.org/news/rsoc-ptrace-5/, accessed 09.04.2023.

[15]. dtrace.org. About DTrace. Available at: (http://dtrace.org/blogs/about/, accessed 09.04.2023.

[16]. userfaultfd(2) - Linux manual page. Available at: https://man7.org/linux/man-pages/man2/userfaultfd.2.html, accessed 09.04.2023.

[17]. Debian Manpages: io_uring(7). Available at: https://manpages.debian.org/unstable/liburing-dev/io_uring.7.en.html, accessed 28.01.2023.

[18]. Vangoor B.K.R., Tarasov V., Zadok E. To FUSE or Not to FUSE: Performance of User-Space File Systems. In Proc. of the 5th USENIX Conference on File and Storage Technologies, 2017, pp. 59-72.

[19]. The reference implementation of the Linux FUSE (Filesystem in Userspace) interface. Available at: https://github.com/libfuse/libfuse, accessed 05.03.2023.

[20]. macFUSE. Available at: https://osxfuse.github.io/, accessed 28.01.2023.

[21]. Windows File System Proxy. Available at: https://winfsp.dev/, accessed 28.01.2023

[22]. GitHub - ligurio/unreliablefs: A FUSE-based fault injection filesystem. Available at: https://github.com/ligurio/unreliablefs, accessed 09.04.2023

[23]. Linux manual page: bpf(2). Available at: https://man7.org/linux/man-pages/man2/bpf.2.html, accessed 05.03.2023.

[24]. The Linux Kernel documentation: Using the Linux Kernel Tracepoints. Available at: (https://docs.kernel.org/trace/tracepoints.html), accessed 05.03.2023.

[25]. GitHub - microsoft/ebpf-for-windows: eBPF implementation that runs on top of Windows. Available at: https://github.com/microsoft/ebpf-for-windows, accessed 09.04.2023

[26]. Linux source Code. Available at: https://github.com/torvalds/linux/blob/master/include/trace/events/io_uring.h#L315, accessed 28.01.2023.

[27]. FoundationDB 7.2: Simulation and Testing. Available at: https://apple.github.io/foundationdb/testing.html, accessed 28.01.2023.

[28]. Navarro Leija O.S., Shiptoski K. et al. Reproducible containers. In Proc. of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, 2020, pp. 167-182.

[29]. Rust Programming Language. Available at: https://www.rust-lang.org, accessed 05.03.2023.
[30]. Costa G. Introducing Glommio, a Thread-per-Core Crate for Rust & Linux | Datadog. Available at: https://www.datadoghq.com/blog/engineering/introducing-glommio/, accessed 05.03.2023.
[31]. Jepsen - Distributed Systems Safety Research. Available at: https://jepsen.io/, accessed 02.03.2023
[32]. Project Gemini: An Open Source Automated Random Testing Suite for ScyllaDB and Cassandra Clusters. Available at: https://www.scylladb.com/2019/12/11/project-gemini-an-open-source-automated-random-testing-suite-for-scylla-and-cassandra-clusters/, accessed 02.03.2023.
[33]. Bornholt J., Kaufmann A. et al. Specifying and checking file system crash-consistency models. In Proc. of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems, 2016, pp. 83-98.
[34]. Chen H., Ziegler D. et al. Using Crash Hoare logic for certifying the FSCQ file system. In Proc. of the 25th Symposium on Operating Systems Principles, 2015, pp. 18-37.
[35]. AddressSanitizer - Clang 17.0.0git documentation. Available at: https://clang.llvm.org/docs/AddressSanitizer.html, accessed 09.04.2023.
[36]. ThreadSanitizer - Clang 17.0.0git documentation. Available at: https://clang.llvm.org/docs/ThreadSanitizer.html, accessed 09.04.2023.
[37]. The Kernel Address Sanitizer (KASAN) - The Linux Kernel documentation. Available at: https://www.kernel.org/doc/html/v4.14/dev-tools/kasan.html, accessed 09.04.2023.
[38]. parsley/instrument_fs at master LizardWizzard/parsley. Available at: https://github.com/LizardWizzard/parsley/tree/master/instrument_fs, accessed 05.03.2023.

Информация об авторах / Information about authors

Дмитрий Кириллович РОДИОНОВ – аспирант ИСП РАН. Сфера научных интересов: высоконагруженные приложения, архитектура систем управления данными, методы тестирования, распределенные системы.

Dmitry Kirillovich RODIONOV – post-graduate student of ISP RAS. Research interests: high-load applications, architecture of data management systems, testing methods, distributed systems.

Сергей Дмитриевич КУЗНЕЦОВ – доктор технических наук, профессор, главный научный сотрудник ИСП РАН, профессор кафедр системного программирования МГУ, МФТИ и ВШЭ. Научные интересы: управление данными, архитектуры систем управления данными, модели и языки данных, управление транзакциями, оптимизация запросов.

Sergey Dmitrievich KUZNETSOV – Doctor of Technical Sciences, Professor, Chief Researcher at ISP RAS, Professor at the Departments of System Programming of MSU, MIPT, and HSE. Research interests: data management, architectures of data management systems, data models and languages, transaction management, query optimization.

DOI: 10.15514/ISPRAS-2023-35(1)-14

Исследование встречаемости небезопасно сериализованных программных объектов в клиентском коде веб-приложений

[1] *Д.Д. Миронов, ORCID: 0000-0002-0092-0806 <denis.mironov@solidwall.io>*
[1,2] *Д.А. Сигалов, ORCID: 0009-0005-2781-6493 <asterite@seclab.cs.msu.ru>*
[1,2] *М.П. Мальков, ORCID: 0009-0000-7019-7556 <wgh@seclab.cs.msu.ru>*

[1] *ООО «СолидСофт»,*
117312, Россия, Москва, ул. Вавилова, д. 47А
[2] *Московский государственный университет им. М.В. Ломоносова,*
119991, Россия, Москва, Ленинские горы, д. 1

Аннотация. В данной статье проведено исследование встречаемости случаев использования небезопасной десериализации при взаимодействии между клиентским кодом и серверной стороной веб–приложения. Особое внимание было уделено сериализованным объектам, отправляемым из JavaScript-кода. Были выявлены характерные особенности шаблонов использования сериализованных объектов внутри клиентского JavaScript-кода и составлены уникальные классы, главной целью которых является облегчение ручного и автоматического анализа веб-приложений. Было разработано и реализовано инструментальное средство, выявляющее сериализованный объект в коде веб-страницы. Данный инструмент способен найти закодированные сериализованные объекты, а также сериализованные объекты, закодированные с помощью нескольких последовательно примененных кодировок. Для найденных экземпляров сериализованных объектов, инструмент определяет контекст, в котором находится найденный объект на странице. Для объектов, находящихся внутри JavaScript-кода, инструмент выявляет ранее упомянутые классы, при помощи сопоставления вершин абстрактного синтаксического дерева кода. После получения результатов исследования был проведен анализ серверных точек ввода данных на предмет десериализации найденных программных объектов на стороне сервера. В результате данной проверки были найдены ранее неизвестные публично уязвимости, о которых было сообщено разработчикам данного программного обеспечения. Одна из них получила идентификатор CVE-2022-24108. По результатам проведенного исследования был предложен метод, позволяющий облегчить как ручной, так и автоматизированный поиск уязвимостей типа “Десериализация недоверенных данных”. Предложенный алгоритм был протестирован на страницах более чем 50000 веб-приложений из списка Alexa Top 1M, а также на страницах 20000 веб-приложений из программ Bug Bounty.

Ключевые слова: небезопасная десериализация; веб-приложения; анализ клиентского кода; автоматизация анализа безопасности; поиск уязвимостей.

Для цитирования: Миронов Д.Д., Сигалов Д.А., Мальков М.П. Исследование встречаемости небезопасно сериализованных программных объектов в клиентском коде веб-приложений. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 223-236. DOI: 10.15514/ISPRAS-2023-35(1)-14

Research into Occurrence of Insecurely-Serialized Objects in Client-Side Code of Web-Applications

[1] *D.D. Mironov, ORCID: 0000-0002-**0092**-0806 <denis.mironov@solidwall.io>*
[1,2] *D.A. Sigalov, ORCID: 0009-0005-2781-6493 <asterite@seclab.cs.msu.ru>*
[1,2] *M.P. Malkov, ORCID: 0009-0000-7019-7556 <wgh@seclab.cs.msu.ru>*

[1] *SolidSoft,*
47A, st. Vavilova, Moscow, 117312, Russia
[2] *Lomonosov Moscow State University,*
GSP-1, Leninskie Gory, Moscow, 119991, Russia

Abstract. This paper studies the occurrence of insecure deserialization in communication between client-side code and the server-side of a web application. Special attention was paid to serialized objects sent from JavaScript client-side code. Specific patterns of using serialized objects within the client-side JavaScript code were identified and unique classes were formulated, whose main goal is to facilitate manual and automatic analysis of web applications. A tool that detects a serialized object in the client-side code of a web page has been designed and implemented. This tool is capable of finding encoded serialized objects as well as serialized objects encoded using several sequentially applied encodings. For found samples of serialized objects, the tool determines the context in which the found object appears on the page. For objects inside JavaScript code, the tool identifies the previously mentioned classes by mapping the vertices of the abstract syntax tree (AST) of the code. Web application endpoints were checked for whether programming objects were deserialized on the server side, after obtaining the results of the study. As a result of this check, previously unknown vulnerabilities were found, which were reported to the developers of this software. One of them was identified as CVE-2022-24108. Based on the results of this research, a method was proposed to facilitate both manual and automated searches for vulnerabilities of the "Deserialization of untrusted data". The proposed algorithm was tested on more than 50,000 web application pages from the Alexa Top 1M list, as well as on 20,000 web application pages from Bug Bounty programs.

Keywords: deserialization of untrusted data; web-applications; client-side code analysis; security analysis automation; vulnerabilities.

For citation: Mironov D.D., Sigalov D.A., Malkov M.P. Research into Occurrence of Insecurely-Serialized Objects in Client-Side Code of Web-Applications. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 223-236 (in Russian). DOI: 10.15514/ISPRAS-2023-35(1)-14

1. Введение

Веб-технологии стремительно развиваются в последние десятилетия, особенно ощутимый скачок произошел при переходе от так называемого “статического” веба к “динамическому” [1]. Раньше взаимодействие клиентской стороны веб-приложения с серверной было ограничено достаточно узким набором технологий, например: отправка GET-запроса с query-параметрами, содержащимися в URL (с помощью тегов <a>, <link>, и т.д.); отправка формы (с помощью тега <form>) с методом, указанным в атрибуте method, и данными из самой формы, которые могли отправляться в нескольких форматах, например: application/x-www-form-urlencoded, multipart/form-data (формат для отправки файлов).

В современных веб-приложениях, помимо вышеописанных способов и форматов взаимодействия клиента и сервера, появилось большое количество новых технологий. Для отправки запросов часто используются интерфейсы fetch и XMLHttpRequest, предоставляемые языком JavaScript. Сами же запросы зачастую могут содержать более сложные форматы данных, например: текстовые форматы, такие как JSON, XML; бинарные форматы, такие как protocol buffers, сериализованные объекты при помощи serialize в языке PHP, сериализованные объекты при помощи pickle в языке Python, сериализованные объекты при помощи Serializable в языке Java.

Такое разнообразие обусловлено несколькими факторами:

- технологии разработки и встроенные в них возможности обработки разных форматов данных;
- большое разнообразие данных, посредством которых пользователь должен коммуницировать с серверной частью веб-приложения;
- безопасность обработки пришедших данных;
- удобность их дальнейшего использования;

С появлением новых веб-технологий, помимо удобства, разработчики и пользователи столкнулись с новыми угрозами безопасности. В 2006 году появился термин “десериализация недоверенных данных”[1]. Этот недостаток также называют “небезопасной десериализацией”. Недостаток такого типа имеет высокий уровень опасности и может приводить к серьезным уязвимостям, таким как выполнение произвольного кода, отказ в обслуживании, чтение и запись локальных файлов [2]. Данный недостаток по сей день входит в классификацию OWASP Top Ten, в которой описаны наиболее часто встречаемые недостатки веб–приложений [3].

2. Методы поиска уязвимости типа “небезопасная десериализация”

Уязвимость типа “небезопасная десериализация” – это уязвимость серверной части приложения. Методы, позволяющие автоматически обнаруживать уязвимости такого рода, можно разделить на два класса.

- Анализ серверного кода, работающий в модели “белого ящика” (“white box”). Анализаторы такого типа имеют доступ к серверному коду; они, как правило, осуществляют преимущественно статический анализ.
- Анализ, работающий без доступа к серверному коду, в режиме “чёрного ящика” (“black box”). Средства, работающие таким образом, могут анализировать серверную часть, лишь взаимодействуя с ней – то есть, в случае веб-приложений, посылая на сервер запросы, и анализируя пришедшие ответы. Таким образом, это всегда динамический анализ.

Исследователи отмечают, что для современных средств поиска уязвимостей в веб-приложениях (как первого, так и второго типов) задача поиска уязвимостей типа "небезопасная десериализация" до сих пор остаётся сложной, существующие средства не предоставляют общего решения этой задачи [4], [5].

Анализаторы первого типа имеют полный доступ к информации об устройстве серверной части приложения; в связи с этим они имеют заведомо большее покрытие кода, чем анализаторы второго типа, и имеют больше шансов найти присутствующие недостатки. К анализаторам такого типа относятся SerialDetector [4], RIPS [6], а также такие промышленные инструменты, как PT Application Inspector [7], Solar appScreener [8], Fortify Static Code Analyzer [9], Checkmarx [10]. Недостатком данного типа анализаторов является необходимость предоставления кода серверной части приложения – в реальных условиях это не всегда возможно. На основе инструмента такого типа невозможно проводить эксперименты с поиском потенциально присутствующих уязвимостей на большой массе реальных сайтов, так как в большинстве случаев невозможно получить доступ к их серверному коду.

Анализаторы второго типа работают, как уже было упомянуто, в модели "чёрного ящика". В данной статье задача рассматривается именно в такой постановке – в условиях, когда доступ к исходному коду серверной части отсутствует. К анализаторам второго типа относятся такие средства, как PT BlackBox Scanner [11], Acunetix [12], Burp Suite Pro Scanner[13], HCL AppScan [14], Detectify [15], Qualys Web Application Scanning [16]. В связи с тем, что

[1] CWE-502: Deserialization of Untrusted Data (https://cwe.mitre.org/data/definitions/502.html)

анализаторы этого типа работают без доступа к коду сервера, они сталкиваются с рядом сложностей.

Для валидации данного недостатка нужно автоматически, по ответу сервера, распознавать факт выполнения десериализации данных на сервере. Помимо этого, для генерации вектора атаки необходимо иметь какие-то сведения о технологиях, используемых на серверной стороне, как минимум требуется знать, какой использовался язык программирования и какие существуют программные классы [17, 18]. Без доступа к серверному коду это нетривиально.

Следующим важным пунктом является поиск точек ввода данных на сервер, то есть URL конечной точки, а также параметры, тело запроса, и, возможно, заголовки. Современные средства могут найти все точки ввода данных, отправляемых из HTML-разметки, но не из JavaScript-кода [19. 20]. Для решения данной задачи обычно используют два подхода: динамический или статический анализ клиентского кода. Оба подхода имеют свои достоинства и недостатки, однако для современных веб-приложений ни один из этих подходов не решает задачу в общем виде [21, 22].

В данной статье предлагается иной подход, заключающийся в создании синтаксических шаблонов классов и разработке инструмента, выявляющего данные классы в клиентском коде веб-приложений.

3. Описание метода

В исходном клиентском коде веб-страницы выявляются сериализованные объекты, в том числе закодированные одной или более кодировок. Предполагается, что, если сериализованный объект находится в клиентском коде, значит он должен быть отправлен на сервер и десериализован. В общем случае это не всегда так, и в ходе исследования были обнаружены случаи, когда сериализованные объекты, присутствующие на странице, на самом деле не будут отправлены на сервер или не будут десериализованы после получения сервером. Тем не менее, в данной работе мы будем в дальнейшем исходить из того предположения, что наличие сериализованного объекта на клиентской стороне говорит о том, что скорее всего он будет отправлен на сервер, и в большей части выявленных классов (о них речь пойдёт ниже) это действительно так.

Далее, для сериализованных объектов, находящихся внутри JavaScript-кода, составляются синтаксические шаблоны, которые содержат какие-то характерные признаки для кода, в котором использован сериализованный объект. Затем, с помощью разработанного инструментального средства, полученные шаблоны выявляются в коде других веб-приложений.

Преимуществом предложенного подхода является то, что, при отнесении приложения к уже существующему классу, становится известна большая часть HTTP-запроса, который нужно выполнить, чтобы передать данные в функцию десериализации на сервере. Путь URL и ключи параметров обычно совпадают, так как используется одно и то же программное обеспечение. Преимуществом подхода также является то, что уже есть некоторые данные о сервере, и нет необходимости отправлять большое множество векторов атаки.

3.1 Алгоритм выявления сериализованных объектов внутри веб-страницы

Для большинства форматов сериализации можно написать достаточно точные сигнатуры. Например, сериализованные Java-объекты начинаются на байты “\xAC\xED\x00\x05”. То есть сигнатуре достаточно проверить, начинаются ли данные с “магической” константы. Сериализованные PHP-объекты (функция serialize) выглядят примерно таким образом: a:1:{i:20041001103319;s:4:\"test\";}. Для такого формата можно написать как “честный” детектор, работающий по тем же правилам, что и реализация unserialize из PHP, так и

несложную эвристическую сигнатуру, которая будет правильно работать в подавляющем числе случаев.

Эти объекты, когда они встречаются в веб-приложениях, обычно присутствуют не в “сыром” виде. Например, cookie допускают только печатные ASCII-символы, в то время как многие форматы сериализации могут содержать произвольные 8-битные байты. Также некоторые объекты могут быть очень большими, и поэтому их сжимают, чтобы они могли поместиться в те же cookie.

Таким образом, нередки подобные цепочки вложенности: Java Serializable → gzip → Base64. Вместо Base64 может быть Base32, URL encoding, 16-ричное кодирование (hex encoding), и т.д. Для сжатия может также использоваться zlib, DEFLATE, и т.д.

Для автоматического поиска сериализованных объектов с учетом таких “вложенностей” был разработан алгоритм, который будет сейчас описан. Входными данными для алгоритма являются HTTP-ресурсы (HTML-страницы, JavaScript-файлы) и их заголовки.

На первом шаге алгоритм выделяет строки-“кандидаты”, которые могут содержать в себе сериализованные объекты. В случае cookie это их значения непосредственно. JavaScript-файлы разбиваются на лексемы, и из них выделяются строковые константы. HTML-страницы синтаксически разбираются, из них извлекаются значения атрибутов (наиболее интересными являются атрибуты с именем “value” у элементов управления форм и атрибуты с именами, начинающимися с префикса “data-”). Инлайновые JavaScript-программы (текст которых непосредственно содержится в HTML-разметке страницы) обрабатываются вышеуказанным способом.

Для каждой такой строки-кандидата применяется рекурсивный алгоритм, псевдокод которого показа на листинге 1.

```
def find_chain(s, callback, max_depth):
    if max_depth <= 0:
          return
    for trans in TRANSFORMATIONS:
        if res := trans(s):
            find_chain(res, callback, max_depth-
1)
    for check in CHECKS:
        if check(s):
            callback(s)
```

Листинг 1: Алгоритм поиска цепочек
Listing 1: Chain search algorithm

`CHECKS` содержит определения функций, проверяющих строку на содержание конкретного типа сериализованного объекта: Java-объект, PHP-сериализованный объект, т.д. `TRANSFORMATIONS` содержит функции, производящие преобразования: вышеупомянутые кодирования вроде Base64, алгоритмы сжатия, синтаксический разбор JSON, и т.д.

То есть, например, если строка может быть раскодирована каким-то образом, то алгоритм вызывается рекурсивно для раскодированной строки. Если строка содержит что-то, что распознает сигнатура сериализованного объекта, то проверка завершается успехом, и вызывается функция `callback`.

H4sIAAAAAAAAAFvzloG1oLiIQTArs ... BZvHUM4CIisqAOqbFhijAQAA (base64)
'\x1f\x8b\x08\x00\x00\x00\x00\x00\x00\ … \xa3\x01\x00\x00' (gzip)
'\xac\xed\x00\x05psr\x00\x11java.util.HashMap\x05\x07\ … \x00\x00\x00w\x04\x00\x00\x00\x00xx' (Java Serializable)

3.2 Выявление контекста сериализованного объекта внутри веб–страницы

В проведенном исследовании важную роль играет анализ JavaScript-кода, подключенного на страницу. Поэтому неотъемлемым шагом исследования являлась реализация алгоритма, определяющего контекст нахождения сериализованного объекта внутри исходного кода веб-страницы. С помощью разработанного алгоритма были получены веб-страницы, на которых были найдены сериализованные объекты внутри JavaScript-кода для выполнения следующего шага работы.

Алгоритм принимает на вход сериализованный объект и контент веб-ресурса. Веб-ресурс представляет из себя HTML-разметку страницы или JavaScript-программу, которая подключается на веб-страницу.

При получении JavaScript-кода алгоритм выявляет наличие сериализованного объекта с помощью вхождения подстроки.

При получении HTML-разметки, она разбивается на лексемы, и происходит обход всех элементов. Проверяются значения всех атрибутов на равенство сериализованному объекту. При встрече тега script, подключенный на страницу JavaScript-код преобразуется в абстрактное синтаксическое дерево при помощи парсера Babel[2]. В полученном дереве происходит проверка всех строковых литералов на равенство сериализованному объекту. Такой способ необходим из-за того, что некоторые символы могут быть экранированы или закодированными специальными сущностями внутри HTML-разметки.

Предложенный алгоритм был применен на страницах более 50000 веб-приложений из списка Alexa Top 1 Million, а также на страницах 20000 веб-приложений из программ Bug Bounty. После выполнения данного шага была получена статистика того, в каких местах располагаются сериализованные объекты на страницах веб-приложения (табл. 1).

Табл 1. Статистика контекста найденных сериализованных объектов
Table 1. Statistics of found serialized objects contexts

Контекст сериализованного объекта	**Количество**
Сериализованные объекты в JavaScript-коде	12997
Сериализованные объекты внутри тега input	9804
Сериализованные объекты внутри тегов data*	2293
Сериализованные объекты внутри тега option	3
Сериализованные объекты внутри тегов script с	74
Сериализованные объекты внутри других тегов	476
Сериализованные объекты в других местах	46

Как видно из статистики табл. 1, 50% сериализованных объектов находится внутри JavaScript-кода, что подтверждает интерес к исследованию сериализованных объектов, найденных в JavaScript-коде.

[2] Babel Parser spec. (https://babeljs.io/docs/en/babel-parser)

3.3 Формирование и выявление классов сериализованных объектов внутри JavaScript-кода

Для классификации сериализованных объектов проводился ручной просмотр найденных примеров и окружающего их кода. В этом коде выделялись повторяющиеся в разных веб-приложениях шаблоны, например:

- уникально названная переменная, содержащая сериализованный объект;
- определенная функция, вызываемая с переменной, содержащей сериализованный объект.

```
var ajaxurl = '/wp-admin/admin-ajax.php ';
var true_posts =
'a:63:{s:14:"posts_per_page";i:10..."order "; s :4:"DESC";}';
var current_page = 1;
var max_pages = '3 ';
...
$('#true_loadmore').click(function() {
    $(this).text('Loading ...') ;
    var data = {
        'action': 'loadmore',
        'query': true_posts,
        'page': current_page
    };
    $.ajax({
           url: ajaxurl,
        data: data,
        type: 'POST',
        success: function(data) {
            ...
        }
    });
});
```

Листинг 2: Пример кода одного из веб-приложений класса LoadMore
Listing 2: Code example from web-application with LoadMore class

В листинге 2 характерными являются следующие особенности:

- сериализованный объект содержится в переменной с идентификатором `true_posts`;
- переменная `true_posts` используется внутри объекта, отправляемого на сервер с помощью функции `$.ajax()`.

Для разметки всех найденных сериализованных объектов выполнялся итеративный процесс:

- выявление особенностей кода;
- создание синтаксического шаблона для выявления найденных особенностей;
- применение классификации для разметки элементов выборки, попадающих под созданные шаблоны;
- процесс повторяется для тех объектов, которые остались неразмеченными.

Алгоритм классификации заключается в следующем:

- происходит обход дерева;
- встречая вершины определенных типов, алгоритм производит сопоставление найденных вершин с разработанными шаблонами;
- при успешном сопоставлении классификатор подает на выход основному алгоритму название класса, к которому относится полученный экземпляр сериализованного объекта.

Предложенный алгоритм был протестирован на страницах, полученных в ходе работы алгоритма из подраздела 3.2. В табл. 2 приведена статистика наиболее интересных классов. Интересными были сочтены, прежде всего, классы, в которых были обнаружены недостатки. Кроме того, как таковые были выбраны наиболее представительные классы. А конкретнее, те, в которые входит большое (> 5) количество приложений. Наконец, интересными были также сочтены классы, имеющие нетривиальную цепочку кодировок (более чем одна кодировка перед сериализацией) – поскольку обнаружение сериализованных объектов с такой цепочкой требует более сложного алгоритма поиска.

Всего было выделено 60 классов, из которых 23 являются нетривиальными (более одного приложения) и 37 тривиальных. Более полную статистику можно найти в репозитории на GitHub[3].

Табл 2. Статистика найденных классов
Table 2. Statistics of found classes

Класс	Приложений	Страниц	Цепочка кодировок
LoadMoreWordPress	65	2273	urldecode, phpser
JCCatalogBitrixOnlineShopSoft	33	128	base64, phpser
Settings	17	33	phpser
QuizSoft	15	919	phpser
SimpleAjaxManagerWordPress	7	935	base64, phpser
MsgLogVAR	4	1528	base64, base64, base64, phpser
InitState	1	101	base64, zlib, phpser
GdnMeta	1	67	urldecode, urldecode, json, phpser

3.4 Описание некоторых классов

В табл. 3, как пример, приведено описание некоторых классов, которые были выбраны как интересные выше, в подразделе 3.3. Представленные сигнатуры описаны на псевдоязыке на основе JavaScript, при этом используются следующие специальные обозначения:

- * – на месте этого символа может быть любое количество (может быть нулевым) любых символов, допустимых в идентификаторе;
- | – один из предложенных вариантов;
- <*> - любое обращение к свойствам (например, .property, [*]);
- SERIALIZED_OBJ – сериализованный объект в той кодировке, в которой он был найден на странице;
- OBJ_WITH_SERIALIZE_OBJ_INSIDE – JavaScript-объект содержащий внутри себя SERIALIZED_OBJ.

[3] Репозиторий со статистикой: https://github.com/miron6/Insecurely-serialized-objects-research

Табл 3: Примеры описания некоторых классов
Table 3: Examples of descriptions of some classes

LoadMoreWordPress	
Сигнатура	**Пример кода**
(true_posts* | ajax_query* | loadmore*) = SERIALIZED_OBJ | OBJ_WITH_SERIALIZE_OBJ_INSIDE	var true_posts = 'a:63:{s:13:"category_name";s:7:"betsoft";...; s:5:"order";s:4:"DESC";}';
JCCatalogBitrixOnlineShopSoft	
Сигнатура	**Пример кода**
obbx_* = new JCCatalog*(OBJ_WITH_SERIALIZE_OBJ_INSIDE)	var obbx_117848907_56410 = new JCCatalogElement({...,'SKU_PROPS':'YTozOntpOjA7czo5OiJWWVNPVEFfTU0iO2k6MTtzOjc6IlNUT1JPTkEiO2k6MjtzOjU6IlRTVkVUIjt9', ...});
Settings	
Сигнатура	**Пример кода**
setting = SERIALIZED_OBJ	setting = 'a:75:{s:6:"action";s:9:"save_edit";s:4:"name" ...s:8:"moduleid";s:3:"154";}'
QuizSoft	
Сигнатура	**Пример кода**
window.qmn_quiz_data<*> = OBJ_WITH_SERIALIZE_OBJ_INSIDE	window.qmn_quiz_data["1"] = {..., "answer_array":"a:5:{i:0;a:3:{i:0;s:25:"...";i:1;d:0;i:2;i:0;}}", ...};
SimpleAjaxManagerWordPress	
Сигнатура	**Пример кода**
samAjax = OBJ_WITH_SERIALIZE_OBJ_INSIDE	var samAjax = {..."clauses":"YTo0OntzOjI6IldDIjtzOjExMzg6IihJRihzYS5hZF91c2VycyA9IDAsIFRSVUUsIChzYS5hZF91… IDApIjt9","doStats":"1",...};
MsgLogVAR	
Сигнатура	**Пример кода**
MSLOG_var = SERIALIZED_OBJ	var MSLOG_var = "V1ZSdmVFN...UFE9PQ==";
InitState	
Сигнатура	**Пример кода**

`___INITIAL_STATE__ = OBJ_WITH_SERIALIZE_OBJ_INSIDE`	`___INITIAL_STATE__={"intl":{"defaultLocale":"en", ...}, ..., "description":"eJyN...k5mw==",...};`
GdnMeta	
Сигнатура	**Пример кода**
`gdn.meta = OBJ_WITH_SERIALIZE_OBJ_INSIDE`	`gdn.meta={"AnalyticsTask": "tick",..., "[{\"HashType\":\"md5\",\"TestMode\":false,\"Trusted\":\"1\",..."}`

4. Поиск уязвимостей

4.1 Методология, использованная при поиске уязвимостей

Для валидации того, происходит ли на сервере десериализация клиентских данных, использовались следующие методы.

- Отправка оригинального объекта, найденного на странице, а также некорректного, с точки зрения формата сериализации, объекта.
- Отправка измененного, но синтаксически корректного объекта. Например, добавление нового поля в словарь или изменение строкового литерала незначительным образом. Данная проверка нужна, чтобы отбросить случаи, когда на серверной части приложения сериализованный объект проходит проверку на строгое равенство с константной строкой.

При использовании всех методов выше сравнивались и анализировались HTTP-ответы.

Для веб-приложений, входящих в программу Bug Bounty, производилась попытка эксплуатации уязвимостей на самих приложениях. Для остальных приложений производился поиск компонентов с открытым исходным кодом, на основе которых они реализованы. В случае нахождения таковых проверялась возможность эксплуатации уязвимости путем ручного анализа кода и атака на стендовые приложения, сделанные на их основе.

В случае, когда подобные проверки не давали однозначных результатов, отправлялись сериализованные объекты стандартных классов используемого языка программирования. Например, в случае PHP таковым являлся класс DateTime. Отправлялся корректный и некорректный сериализованные объект. Десериализация некорректного PHP-объекта DateTime приводит к фатальной ошибке сервера, что являлось индикатором происходящей на сервере десериализации.

4.2 Найденные уязвимости

Класс Settings: было выявлено, что в данный класс входят приложения, сделанные на основе PHP-фреймворка OpenCart с использованием одного и того же плагина. В данном плагине присутствует недостаток небезопасной сериализации.

- Было реализовано стендовое приложение с использованием тех же технологий.
- Была найдена цепочка гаджетов с использованием классов базового фреймворка, позволяющая выполнить произвольный код на сервере.
- Данная уязвимость не была публично известной, поэтому разработчики продукта были оповещены о наличии данной уязвимости, а также был получен идентификатор

уязвимости CVE-2022-24108[4].

Класс SimpleAjaxManagerWordpress: было выявлено, что приложения данного класса используют CMS WordPress (является одной из самых распространенных CMS в интернете [23, 24]) и плагин Simple Ajax Manager.

- Было реализовано стендовое приложение с использованием тех же технологий.
- Были найдены две уязвимости: небезопасная десериализация и SQL-инъекция через поля сериализованного объекта.
- Найденная SQL-инъекция не была публично известной уязвимостью, поэтому был отправлен запрос на получение идентификатора CVE в организацию WPScan. Уязвимость не получила нового идентификатора уязвимости, но информация о ней была добавлена в существующую запись о недостатке “Десериализации недоверенных данных”[5].

Класс LoadMoreWordpress: приложения данного класса используют CMS WordPress, но технология, где используется небезопасная десериализация, не является плагином. Код, где выполняется уязвимый запрос, был приведен на одном из форумов для разработчиков.

- Страницы данного класса были найдены на ряде веб-приложений, входящих в программу Bug Bounty.
- При помощи эксплуатации небезопасной десериализации получилось добиться выполнения произвольного кода на серверной стороне для всех этих приложений.
- Был отправлен отчет об уязвимости на один из главных агрегаторов Bug Bounty программ HackerOne [25].

5. Заключение

В данной статье предложен метод поиска недостатка “Десериализация недоверенных данных” при помощи поиска сериализованных объектов внутри исходного кода веб-страницы, создания синтаксических шаблонов, основанных на особенностях кода, содержащего в себе сериализованный объект, и последующей классификации страниц веб-приложений.

Данный метод был протестирован на страницах более чем 70000 веб-приложений. Получены синтаксические шаблоны для 60 классов, которые могут быть использованы при дальнейшем автоматизированном анализе веб-приложений. На приложениях трех найденных классов получилось установить наличие искомого недостатка. В ходе исследования был получен идентификатор уязвимости – CVE, дополнена информация о другой общеизвестной уязвимости и отправлен отчет в программу Bug Bounty для целого ряда приложений, имеющих схожий недостаток.

Список литературы / References

[1] Nath K., Dhar S., Basishtha S. Web 1.0 to Web 3.0 - Evolution of the Web and its various challenges. In Proc. of the International Conference on Reliability Optimization and Information Technology (ICROIT), 2014, pp. 86-89.

[2] Koutroumpouchos N., Lavdanis G. et al. ObjectMap: detecting insecure object deserialization. In Proc. of the 23rd Pan-Hellenic Conference on Informatics (PCI'19), 2019, pp/ 67-72.

[3] Bach-Nutman M. Understanding The Top 10 OWASP Vulnerabilities. arXiv preprint arXiv:2012.09960, 2020, 4 p.

[4] CVE-2022-24108 (https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-24108)

[5] Wordpress: Simple Ads Manager vulnerability page (https://wpscan.com/vulnerability/38787b49-c19c-49db-925a-69e2c9cf7a43)

[4] Shcherbakov M., Balliu M. SerialDetector: SerialDetector: Principled and Practical Exploration of Object Injection Vulnerabilities for the Web. In Proc. of the Network and Distributed Systems Security (NDSS) Symposium, 2021, 18 p.
[5] Sabatini A. Evaluating the Testability of Insecure Deserialization Vulnerabilities via Static Analysis. Tesi di Laurea Magistrale in Computer Science and Engineering. Politecnico Milano, 2022, 72 p.
[6] Dahse J., Holz T. Simulation of Built-in PHP Features for Precise Static Code Analysis. In Proc. of the Network and Distributed System Security (NDSS) Symposium, 2014, 15 p.
[7] PT Application Inspector. Available at: https://www.ptsecurity.com/ru-ru/products/ai/, accessed 02.04.2023.
[8] Solar appScreener. Available at: https://rt-solar.ru/products/solar_appscreener/, accessed 02.04.2023.
[9] Fortify Static Code Analyzer. Available at: https://www.microfocus.com/en-us/cyberres/application-security/static-code-analyzer, accessed 02.04.2023.
[10] Checkmarx. Available at: https://checkmarx.com/, accessed 02.04.2023.
[11] PT BlackBox Scanner. Available at: https://bbs.ptsecurity.com/, accessed 02.04.2023.
[12] Acunetix. Available at: https://www.acunetix.com/, accessed 02.04.2023.
[13] Burp Suite's web vulnerability scanner. Available at: https://portswigger.net/burp/vulnerability-scanner, accessed 02.04.2023.
[14] HCL AppScan. Available at: https://www.hcltechsw.com/appscan, accessed 02.04.2023.
[15] Detectify Web Application Scanning. Available at: https://detectify.com/product/application-scanning, accessed 02.04.2023.
[16] Qualys Web Application Scanning. Available at: https://www.qualys.com/apps/web-app-scanning/, accessed 02.04.2023.
[17] Esser S. Shocking News in PHP Exploitation. Slides of the Presentation at the Power of Community Conference (POC), 2009. Available at: https://www.suspekt.org/wp-content/uploads/2019/12/POC2009-ShockingNewsInPHPExploitation.pdf, accessed at 04.03.2023.
[18] Esser S. Utilizing code reuse or return oriented programming in PHP applications. Slides of the Presentation at the BlackHat USA Conference, 2010. Available at: Available at: https://media.blackhat.com/bh-us-10/presentations/Esser/BlackHat-USA-2010-Esser-Utilizing-Code-Reuse-Or-Return-Oriented-Programming-In-PHP-Application-Exploits-slides.pdf, accessed at 04.03.2023.
[19] Сигалов Д.А., Хашаев А.А., Гамаюнов Д.Ю. Обнаружение серверных точек взаимодействия в веб-приложениях на основе анализа клиентского JavaScript-кода. Прикладная дискретная математика вып. 53, 2021 г., стр. 32-54. / Sigalov D.A., Khashaev A.A., Gamayunov D.Yu. Detecting server-side endpoints in web applications based on static analysis of client-side JavaScript code. Prikladnaya Diskretnaya Matematika, issue 53, 2021, pp. 32-54 (in Russian).
[20] Раздобаров А.В., Петухов А.А., Гамаюнов Д.Ю. Проблемы обнаружения уязвимостей в современных веб-приложениях. Проблемы информационной безопасности. Компьютерные системы, вып. 4, 2015 г., стр. 64-69. / Razdobarov A.V., Petukhov A.A., Gamayunov D.Yu. Problems overview for modern web applications vulnerabilities discovery. Information Security Problems. Computer Systems, issue 4, 2015, pp. 64-69 (in Russian).
[21] Pradel M., Schuh P., Sen K. TypeDevil: Dynamic type inconsistency analysis for JavaScript. In Proc. of the IEEE/ACM 37th IEEE International Conference on Software Engineering, 2015, pp. 314-324.
[22] Park C., Ryu S. Scalable and Precise Static Analysis of JavaScript Applications via Loop-Sensitivity. In Proc. of the 29th European Conference on Object-Oriented Programming (ECOOP), 2015, pp. 735-756.
[23] Lin J., Sayagh M., Hassan A.E. The Co-evolution of the WordPress Platform and its Plugins. ACM Transactions on Software Engineering and Methodology, vol. 32, issue 1, 2023, article no. 19, 24 p.
[24] Patel S.K., Rathod V.R., Prajapati J.B. Performance Analysis of Content Management Systems - Joomla, Drupal and WordPress. International Journal of Computer Applications, vol. 21, issue 4, 2011, pp 39-43.
[25] Walshe T., Simpson A. An Empirical Study of Bug Bounty Programs. In Proc. of the IEEE 2nd International Workshop on Intelligent Bug Fixing (IBF), 2020, pp. 35-44.

Информация об авторах / Information about authors

Денис Дмитриевич МИРОНОВ – исследователь проблем безопасности. Сфера научных интересов: статический анализ программ для задач безопасности приложений, безопасность веб-приложений.

Denis Dmitrievich MIRONOV – Security Researcher. Research interests: static program analysis for application security, web application security.

Даниил Алексеевич СИГАЛОВ – младший научный сотрудник лаборатории математических проблем компьютерной безопасности факультета ВМК МГУ, исследователь проблем безопасности в ООО «Солидсофт». Сфера научных интересов: безопасность веб-приложений, статический и динамический анализ программ для задач безопасности приложений.

Daniil Alekseevich SIGALOV – Junior Researcher of Laboratory of Mathematical Problems of Computer Cecurity at the CMC Faculty of Lomonosov Moscow State University, Security Researcher at SolidSoft LLC. Research interests: web application security, static and dynamic program analysis for application security.

Максим Петрович МАЛЬКОВ – ведущий программист лаборатории математических проблем компьютерной безопасности факультета ВМК МГУ, исследователь безопасности в ООО «Солидсофт». Сфера научных интересов: безопасность веб-приложений.

Maxim Petrovich MALKOV – Lead Programmer of Laboratory of Mathematical Problems of Computer Security at the CMC Faculty of Lomonosov Moscow State University, Security Researcher at SolidSoft LLC. Research interests: web application security.

DOI: 10.15514/ISPRAS-2023-35(1)-15

Сравнение графовых векторных представлений исходного кода с текстовыми моделями на основе архитектур CNN и CodeBERT

В.А. Романов, ORCID: 0000-0003-3772-0039 <v.romanov@innopolis.ru>
В.В. Иванов, ORCID: 0000-0003-3289-8188 <v.ivanov@innopolis.ru>

Университет Иннополис,
420500, Россия, г. Иннополис, ул. Университетская, д. 1.

Аннотация. Одним из возможных способов уменьшения ошибок в исходном коде является создание интеллектуальных инструментов, облегчающих процесс разработки. Такие инструменты часто используют векторные представления исходного кода и методы машинного обучения, заимствованные из области обработки естественного языка. Однако такие подходы не учитывают специфику исходного кода и его структуру. Данная работа посвящена исследованию методов предварительного обучения графовых векторных представлений исходного кода, где граф представляет структуру программы. Результаты показывают, что графовые векторные представления позволяют достичь точности классификации типов переменных программ, написанных на языке Python, сравнимой с векторными представлениями CodeBERT. Более того, одновременное использование текстовых и графовых векторных представлений в составе гибридной модели позволяет повысить точность классификации типов более чем на 10%.

Ключевые слова: исходный код; классификация типов переменных; Python; графовые нейронные сети; CodeBERT

Для цитирования: Романов В.А., Иванов В.В. Сравнение графовых векторных представлений исходного кода с текстовыми моделями на основе архитектур CNN и CodeBERT. Труды ИСП РАН, том 35, вып. 1, 2023 г., стр. 237-264. DOI: 10.15514/ISPRAS–2023–35(1)–15

Благодарности: Исследование выполнено при поддержке гранта Российского научного фонда (проект № 22-21-00493, https://rscf.ru/project/22-21-00493/).

Comparison of Graph Embeddings for Source Code with Text Models Based on CNN and CodeBERT Architectures

V.A. Romanov, ORCID: 0000-0003-3772-0039 <v.romanov@innopolis.ru>
V.V. Ivanov, ORCID: 0000-0003-3289-8188 <v.ivanov@innopolis.ru>

Innopolis University,
1, Universitetskaya Str., Innopolis, 420500, Russia

Abstract. One possible way to reduce bugs in source code is to create intelligent tools that make the development process easier. Such tools often use vector representations of the source code and machine learning techniques borrowed from the field of natural language processing. However, such approaches do not take into account the specifics of the source code and its structure. This work studies methods for pretraining graph vector representations for source code, where the graph represents the structure of the program. The results show that graph embeddings allow to achieve an accuracy of classifying variable types in Python programs that is comparable to CodeBERT embeddings. Moreover, the simultaneous use of text and graph embeddings as part of a hybrid model can improve the accuracy of type classification by more than 10%.

Keywords: source code; variable type prediction; Python; graph neural networks; CodeBERT

For citation: Romanov V.A., Ivanov V.V. Comparison of Graph Embeddings for Source Code with Text Models Based on CNN and CodeBERT Architectures. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 1, 2023. pp. 237-264 (in Russian). DOI: 10.15514/ISPRAS-2023-35(1)-15

Acknowledgements: The study was supported by a grant from the Russian Science Foundation (project no. 22-21-00493, https://rscf.ru/project/22-21-00493/).

1. Введение

В настоящий момент эталонной нейро-архитектурой для обработки исходного кода является трансформер [1]. Большинство моделей данной архитектуры принимают на вход исходный код в виде последовательности токенов. При этом специфические свойства исходного кода, такие как нелинейность исполнения, не учитываются. Примерами моделей, реализованных с использованием данной архитектуры, являются CuBERT и CodeBERT, показавшие свою эффективность для решения целевых задач [2, 3].

Альтернативой таким подходам являются предварительно обученные модели на основе графов, построенных из исходного кода. Такие модели могут учитывать зависимости в программе, даже если исходный код разделён на несколько файлов. Использование представления исходного кода в виде графа перспективно и исследовалось в нескольких работах, в том числе посвященных предварительному обучению [4, 5]. Тем не менее подходы, основанные на графах, новы и недостаточно изучены. В данной работе исследуется два подхода создания предварительно обученных графовых векторных представлений: на основе методов тренировки реляционных векторных представлений и на основе графовых нейронных сетей. Качество полученных векторных представлений оценивается на задаче классификации типов переменных.

Одна из проблем существующих подходов анализа исходного кода с помощью методов обработки текста – ограниченность обрабатываемого контекста. Модели машинного обучения, которые принимают исходный код в виде последовательности токенов (модели на архитектуре трансформер), могут обрабатывать за раз ограниченное количество токенов. При этом релевантные для решаемой задачи части кода могут быть размещены в нескольких файлах. По этой причине актуальными становятся методы создания предварительно обученных моделей на основе графов, построенных из исходного кода.

Идея создания предварительной обученных моделей для исходного кода, основанных на графовых нейро-сетевых моделях (GNN) не нова [5-9]. Большинство таких подходов тренируются путём решения задачи предсказания наличия связей между различными элементами программы и их типов. Модели, использующие графовые нейронные сети, уже показали свою эффективность при решении целевых задач. Однако в существующих исследованиях не проведено сравнение предварительно обученных моделей, реализованных с помощью графовых нейронных сетей и с помощью архитектуры трансформер. Проведение такого сравнения является целью данной работы.

Представления исходного кода в виде последовательности токенов и в виде графа могут дополнять друг друга, например, при создании статического анализатора типов в языках программирования с динамической типизацией (JavaScript, Python). Как правило, статические анализаторы типов полагаются на наличие подсказок в коде (type hints). Однако формальные методы не всегда способны предоставить однозначный ответ. Эту проблему можно частично решить, если механизм вывода типов сможет ранжировать кандидатов на основе дополнительных данных, таких как имена переменных или характер их использования [10]. Для решения задачи классификации типов переменных ранее уже демонстрировалась эффективность отдельно представлений исходного кода в виде последовательности токенов и в виде графа. В данной работе показано, что точность

классификации типов можно улучшить путём применения гибридной модели, которая использует одновременно два вида представлений исходного кода.

Новизна данной работы заключается в следующем.

1) Разработан метод преобразования исходного кода на языке Python в граф, содержащий глобальные связи, такие как вызовы функций и импортирования модулей, и отображает структуру программы.
2) Предложен подход для предварительной тренировки реляционных векторных представлений для исходного кода на основе графа с добавлением k-hop рёбер.
3) Предложен подход предварительной тренировки графовой нейросетевой модели для исходного кода, целью которой является создание векторных представлений. В качестве задач предварительного обучения используются предсказание имён, предсказание и классификация связей в графе, классификация типов узлов в графе.
4) Проведено исследование применимости предварительно обученных векторных представлений для решения задачи классификации типов переменных для программ, написанных на языке Python. Проведено сравнение графовых векторных представлений с векторными представлениями CodeBERT.

Исходный код для получения результатов опубликован на GitHub[1].

2. Обзор литературы

2.1 Предварительно обученные модели для исходного кода

Предварительно обученные модели позволяют сократить время тренировки. Они используются для инициализации моделей машинного обучения при решении самых разных задач. С появлением архитектуры трансформер многие работы исследовали её применение к исходному коду [2, 11, 12]. Один из самых базовых подходов для предварительного обучения – маскирующая модель (MLM). В некоторых работах используются дополнительные задачи предварительного обучения, разработанные специально для исходного кода. К ним относятся перевод между языками программирования, генерация текстового описания для кода и генерация кода из текстового описания [13]. Существуют модификации, использующие информацию из графа программы, например, графа потока данных [4].

В последние годы появляется всё больше работ, исследующих использование модальности исходного кода в виде дерева или графа [4, 14, 15], а также графовые нейронные сети для обучения векторных представлений для исходного кода [5, 13]. Тем не менее сравнение предварительно обученных моделей на основе архитектуры трансформер и графовых нейронных сетей всё ещё не проведено. Одна из причин – сложность оценки качества предварительно обученных моделей.

2.2 Методы оценки предварительно обученных моделей для исходного кода

В последнее время предварительно обученные модели для исходного кода всё чаще оцениваются путем решения таких целевых задач, как обнаружение неправильно используемых переменных [2, 11], предсказание имён переменных и функций [5, 11], поиск исходного кода [4], а также перевод между языками программирования [4, 13]. Иногда задачи направлены на то, чтобы понять, какие свойства исходного кода можно извлечь из предварительно обученных векторных представлений. Примерами таких задач могут быть классификация узлов абстрактного синтаксического дерева программы, оценка цикломатической сложности, оценка длины кода и обнаружение неправильных типов [16].

[1] https://github.com/VitalyRomanov/method-embedding

2.3 Методы решения задачи классификации типов переменных

Использование предварительно обученных моделей часто позволяет сократить время тренировки и количество требуемых данных при решении целевых задач. На настоящий момент не существует успешной предварительно обученной модели для исходного кода, использующей графовые нейронные сети. Тем не менее существует множество работ, в которых такие нейронные сети применяются для решения целевых задач. Одним из примеров является задача классификации типов переменных в программах, написанных с использованием языков программирования с динамической типизацией (Python, JavaScript).

При решении задачи классификации типов часто исходный код, подаваемый на вход модели машинного обучения, представляют в виде последовательности токенов [10, 17, 18]. Такая задача имеет смысл для динамических языков программирования, для которых формальный статический анализатор не всегда может предложить однозначный ответ. При решении этой задачи с помощью машинного обучения, для классификации типа зачастую используется не только информация о структуре исходного кода, но также документация и имена переменных [19, 20]. Результат классификации может затем передаваться в качестве рекомендаций статическому анализатору. В одной из работ был представлен подход под названием TypeWriter, основанный на рекуррентных нейронных сетях (RNN) [21]. Он сочетает в себе вероятностную оценку возможных типов и дальнейшую верификацию предложенных кандидатов. Благодаря второму шагу, такой подход может гарантировать корректность полученного типа.

Самая ранняя работа, посвящённая задаче классификации типов, исследовала применение машинного обучения для классификации типов переменных программ, написанных на языке JavaScript. При этом исходный код был представлен в виде графа потока управления [22]. Недавно был предложен подход под названием Typilus [23]. Авторы этого подхода использовали представление исходного кода в виде графа потока управления и данных. В отличие от предыдущих работ, где тип мог принимать одно из заранее выбранных значений, в этом подходе новые типы могут быть добавлены даже после обучения. В последние годы чаще можно найти работы, в которых основной моделью для классификации типов является графовая нейронная сеть [24-26].

2.4 Существующие подходы преобразования исходного кода в граф

Целью данной работы является исследование применения графовых векторных представлений для решения целевых задач исходного кода. Формат графа может иметь существенный вклад в качество финальных векторных представлений. Далее рассмотрены несколько форматов: в виде абстрактного синтаксического дерева (AST), графа потока управления и графа потока данных, и межпроцедурного графа. Эти виды представлений могут быть комбинированы в разных сочетаниях.

Представление в виде AST получается непосредственно из исходного кода программы. Узлы обозначают элементы исходного кода, а ребра – зависимости. Такое представление получить проще всего, однако оно обладает рядом недостатков: наличие узлов с повторяющейся функциональностью (например узлы циклов for и while), зависимость от языка программирования, большое количество узлов в дереве. Несмотря на это, представление исходного кода с помощью AST на сегодняшний день широко применяется в исследовательских работах [27-29].

При использовании графов потока управления узлы обозначают выражения, а ребра – передачу управления между выражениями. Часто процедуры при таком представлении имеют входные и выходные узлы. Графы потока управления содержат меньше узлов и могут обеспечить представление программы более независимое от языка программирования. Существует множество работ, использующих граф потока управления для анализа исходного

кода с помощью машинного обучения [30-35]. Иногда, вместо использования полноценного графа потока управления, представление AST дополняется рёбрами потока управления.

Графы потока данных получаются путём извлечения зависимостей между переменными. Такие представления могут не содержать условных выражений и, как следствие, операторов управления. Представление в виде графа потока данных получить труднее всего, но его польза для решения задач машинного обучения была не раз продемонстрирована [31, 32, 34-36].

При обработке исходного кода графы потока управления или данных обычно строятся для одной процедуры или функции. Межпроцедурные графы соединяют несколько отдельных графов в один через входные и выходные узлы, которые определены для каждой процедуры. Существующие инструменты позволяют получить такое представление только для узкого круга языков программирования. Один из способов построения такого графа использует информацию от компилятора программы [37].

Формат представления в виде графа полезен прежде всего потому, что он позволяет запечатлеть структурные зависимости в исходном коде. Однако не все зависимости могут быть использованы существующими методами машинного обучения. Особенно это справедливо для узлов в графе, расположенных на большом удалении друг от друга. Чтобы сократить расстояние в графе можно использовать дополнительные ребра, которые явным образом представляют полезные связи между удалёнными узлами. Так, представление AST часто дополняется вспомогательными рёбрами. Они могут быть из числа рёбер потока управления или данных, а также выполнять сугубо вспомогательные функции. Встречающиеся типы дополнительных ребер можно разбить на несколько групп:

- рёбра, обозначающие тип данных (Type, Inherits) [25, 30, 38];
- рёбра упоминания (LastUse, LastWrite) [38];
- рёбра вызова функций (FunctionCall) [6, 39];
- рёбра зависимости данных (NextUse) [23, 28];
- рёбра следования (NextExpression, NextArgument и NextToken);
- рёбра возврата (ReturnTo) [29, 31, 38];
- рёбра соседних узлов (Sibling) [38];
- рёбра аттрибутов (NodeName и NodeType) [28, 40];
- обратные рёбра [41].

При выборе представления графа возникает естественный вопрос: какие типы рёбер в графе важны для достижения высокого качества моделей машинного обучения на выбранной задаче? К сожалению, удалось найти только две работы, затрагивающие данный вопрос [23, 41], в которых проводилась оценка вклада различных типов рёбер при решении задач идентификации неправильно используемых переменных, классификация имён переменных и классификация типов переменных.

2.5 Векторные представления для графов

Современные графовые нейронные сети масштабируется до миллионов и миллиардов узлов [42], что открывает возможности для их применения к широкому кругу задач. В одной из работ был разработан метод под названием M-GNN, предназначенный для создания иерархических векторных представлений для графов [43]. Иерархия создаётся путём упрощения графа за счёт объединения узлов в суперузлы. Чем выше уровень иерархии, тем меньше узлов в графе.

Архитектура графовой нейросети под названием R-GCN была разработана для обработки гетерогенных графов [44]. Она может обрабатывать графы с несколькими типами рёбер.

Данная модель является автокодировщиком, цель которого восстановить информацию о рёбрах в графе. В одной из работ авторы решили использовать декодировщик с меньшим количеством тренируемых параметров [45]. В качестве критерия существования ребра они использовали целевую функцию RotatE. В другой работе R-GCN-подобная архитектура была модифицирована для работы с механизмом внимания [46].

Графовые векторные представления всё чаще находят своё применение при анализе исходного кода с помощью машинного обучения. Далее приводится описание используемого в данной работе подхода преобразования исходного кода в граф и методов тренировки графовых векторных представлений.

3. Методология

Далее описываются применяемые в данной работе методы преобразования исходного кода в граф, тренировки графовых векторных представлений и тестирования полученных векторных представлений на задаче классификации типов переменных.

3.1 Преобразование исходного кода в граф

В рамках данной работы к графовому представлению исходного кода выдвигаются следующие требования: 1) переменные с одинаковым именем, встречающиеся в теле одной функции, должны интерпретироваться как один узел в графе, что позволит более эффективно извлекать информацию о позициях в коде, где используется данная переменная; 2) для выражений, определённых в теле условного оператора if, циклов for и while, блока обработки исключений try, в графе должны присутствовать связи с упомянутыми выше операторами (например, для выражения в блоке оператора if должна присутствовать связь с узлом, соответствующим данному оператору), что позволит увеличить степень вершин графа и сократить его диаметр; 3) в графе должен отражаться порядок исполнения выражений; 4) связи с импортируемыми модулями, вызываемыми функциями, и наследуемыми классами должны однозначно разрешаться; 5) значения констант в исходном коде (чисел и строк) должны не отображаться в графе для сокращения числа уникальных узлов; 6) для уменьшения количества уникальных имён, имена функций и переменных должны быть токенизированы.

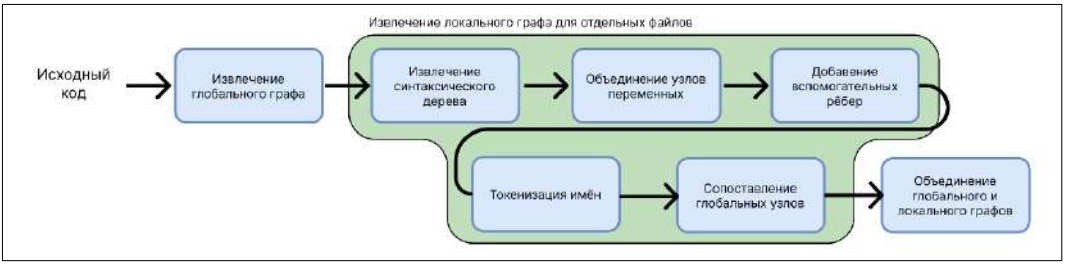

Рис. 1. Процедура преобразования исходного кода в граф
Fig. 1. The procedure for converting the source code into a graph

Стоит отметить, что для некоторых задач исходного кода сформулированное выше представление может быть избыточным. В частности, в разделе 4.8 проводятся эксперименты по проверке необходимости использования глобальных связей, сабтокенизации имён и наличия информации о типах связей между узлами при решении задачи классификации типов переменных. Так как цель работы заключается в проверке качества предварительно обученной модели, для изначальных экспериментов, производится построение графа в соответствии с упомянутыми выше требованиями. Построение графа осуществляется путём использования процедуры, приведённой на рис. 1.

На первом шаге осуществляется извлечение глобальных взаимосвязей с помощью утилиты Sourcetrail[2]. В процессе индексирования Sourcetrail создает базу данных глобальных связей. Примерами могут быть связи с вызываемыми в коде функциями, импортированными модулями, а также отношения наследования. Граф $g_{global} = (N_{global}, E_{global})$ представляет собой множество глобальных узлов и взаимосвязей между ними. Утилита Sourcetrail позволяет извлекать зависимости, даже когда происходит импортирование из сторонних пакетов. В результате для часто используемых библиотек собирается информация о том, как именно они используются. В дополнение, Sourcetrail сохраняет соответствие между исходным кодом и узлами в графе, что позволяет позднее для решения задачи классификации типов одновременно использовать совместное представление исходного кода в виде последовательности токенов и в виде графа.

Далее, обработка исходного кода в кодовой базе осуществляется на уровне отдельных файлов. Для извлечения синтаксического дерева программы используется модуль ast (Python 3.8). Далее происходит формирование локального графа исходного кода $g_{local} = (N_{local}, E_{local})$, который создаётся для отдельно взятого файла. На шаге объединения имён переменных происходит преобразование синтаксического дерева программы в граф. Переменные с одинаковым именем внутри тела функции объединяются в один узел. Затем происходит добавление вспомогательных рёбер. Для выражений, определённых в теле условного оператора if, циклов for и while, блока обработки исключений try, добавляются связи с упомянутыми выше операторами (например, для выражения в блоке оператора if добавляется связь с узлом, соответствующим данному оператору). Для отображения порядка исполнения программы в графе используются дополнительные рёбра next и prev.

Последний шаг в процессе создания локального графа – токенизация имён. На этом шаге в граф добавляются узлы и рёбра типа subword, которые обозначают токены имён. Узлы, представляющие токены, являются общими для всех файлов в кодовой базе. Без токенизации количество уникальных имён растет за счет неологизмов по мере добавления нового кода в кодовую базу [47]. Одним из самых популярных инструментов для токенизации является sentencepiece [48]. Он основан на алгоритмах сжатия, находит наиболее часто встречающиеся подстроки в кодовой базе и использует их в качестве токенов.

На последнем этапе обработки файла в кодовой базе происходит сопоставление глобальных узлов. К локальному графу добавляются ребра типа global_mention, связывающие узлы локального графа с соответствующими им узлами глобального графа. Эти ребра можно представить графом $g_{global\_mention} = \left(N_{local} \cup N_{global}, E_{global\_mention}\right)$. Результатом сопоставления является граф $g = g_{local} \cup g_{global} \cup g_{global\_mention}$. После этого происходит объединение графа g для отдельного файла с общим графом G, который объединяет в себе все обрабатываемые пакеты.

Пример графа, построенного из исходного кода, показан на рис. 2. Исходный код содержит определения двух функций и вызов функции. В ходе извлечения абстрактного синтаксического дерева создаются узлы определений функций (FunctionDef) и аргументов (arg), выражений return, if и (Call). Константы, такие как числа и строки, заменяются специальным узлом Constant. Далее происходит объединение узлов переменных. Так, оба упоминания переменной *c* в функции condition обозначаются одним узлом.

[2] https://github.com/CoatiSoftware/Sourcetrail

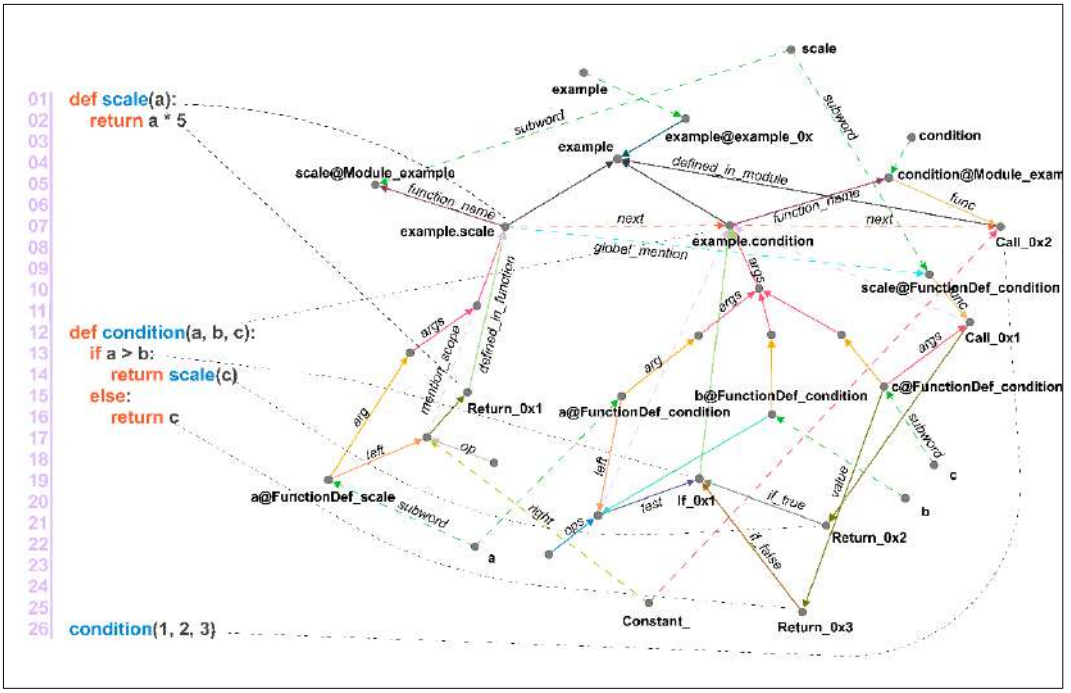

Рис. 2. Пример графа исходного кода, построенного для двух функций. Исходный код показан слева. Пунктирные черные линии ведут от операторов в коде к узлам графа, которые представляют эти операторы. Пунктирные цветные линии представляют собой дополнительные ребра, добавленные при аугментации. Имена некоторых узлов и некоторых ребер, а также всех обратных ребер опущены для облегчения понимания графа

Fig. 2. An example of a source code graph built for two functions. The source code is shown on the left. Dashed black lines lead from the statements in the code to the graph nodes that represent those statements. The dotted colored lines represent additional edges added during the augmentation. The names of some nodes and some edges, as well as all back edges, have been omitted to make the graph easier to understand

Рёбра, представленные сплошными линиями, взяты непосредственно из абстрактного синтаксического дерева. Ребра, представленные пунктирными линиями, получены на этапе добавления вспомогательных рёбер. Например, тип ребра mention_scope в функции scale соединяет узел, представляющий операцию умножения, с узлом, представляющим определение функции. Другие вспомогательные ребра используются для соединения следующих друг за другом выражений в коде (next, prev), контекста условного оператора (if_true, if_false). На этапе токенизации имён создаются узлы и рёбра типа subword. На этапе сопоставления глобальных узлов добавляются рёбра типа global_mention. Все ребра, кроме subword и op, соединяющих операторы, имеют парные обратные рёбра, которые не показаны в данном примере.

Узлы глобального графа можно отделить от общего графа. Глобальный узел представляет собой конкретный класс или функцию, которая может появляться как в различных файлах, так и в различных пакетах. Глобальные узлы связаны с любыми их упоминаниями в локальном графе с помощью отношений типа global_mention. Таким образом, локальный и глобальный графы можно отделить друг от друга, если удалить связи этого типа.

Этот факт можно использовать для предварительной тренировки. Например, в будущем, в качестве одной из целевых функций для предварительной тренировки может быть исследовано предсказание глобальных отношений на основе локального графа.

Атрибуты узлов имеют важную роль при предварительном обучении. В предложенном формате графа каждый узел ассоциируется с именем и типом.

Для большинства узлов, полученных из абстрактного синтаксического дерева, имя определяется типом узла. Для узлов, обозначающих имена переменных, функций, или аргументов, имя берётся из исходного кода. Имена для глобальных узлов извлекаются утилитой Sourcetrail и представляют собой полный путь от пакета до имени элемента (например, для метода __init__ класса string пакета builtins имя будет иметь вид builtins.string.__init__).

3.2 Предварительное обучение графовых векторных представлений

Для создания предварительно обученных векторных представлений предложено два подхода. Первый подход использует методы создания реляционных векторных представлений, разработанных для графов знания. Как и граф знания, представление исходного кода в виде графа содержит сущности (элементы программы) и отношения между ними. Такой способ не требует большого количества вычислительных ресурсов, и поэтому рассматривается в данной работе. Второй подход использует графовые нейронные сети.

3.3 Метод обучения реляционных векторных представлений

Метод обучения реляционных векторных представлений заключается в следующем. Цель тренировки – восстановление троек отношений (h, r, t), где h и t – головной и хвостовой объекты, а r – тип отношения. Цель тренировки заключается в том, чтобы максимизировать правдоподобие для правильных троек и минимизировать – для неправильных. В данной работе рассматривается несколько стандартных подходов для тренировки:

- TransR [49];
- DistMult [50];
- RESCAL [51];
- ComplEx [52];
- RotatE [53].

Одной из особенностей графа исходного кода является низкая связность узлов в графе, которая затрудняет обучение реляционных векторных представлений. Для того чтобы улучшить процесс тренировки, предлагается метод модификации графа, состоящий из следующих шагов:

1) убрать обратные рёбра, так как они нужны только для процесса обмена сообщениями;
2) создать k-hop ребра $k = 1..3$, которые соединяют узлы на расстоянии k и повышают степень связности графа;
3) добавить узлы, обозначающие типы, что позволяет сделать узлы одного типа ближе друг к другу независимо от используемой целевой функции для тренировки векторных представлений.

3.4 Метод обучения векторных представлений с помощью графовых нейронных сетей

В данной работе используется реляционная графовая свёрточная сеть (R-GCN), которая использовалась в недавних работах [5, 54]. Вначале все узлы инициализируются с помощью векторных представлений имён узлов. Затем осуществляется несколько итераций передачи сообщений. Агрегирование сообщений производится после каждой итерации путём усреднения векторных представлений, полученных от соседей. Рекуррентное уравнение для обновления состояния узла, которое часто встречается в литературе, можно записать в виде

$$\boldsymbol{h_i^{n+1}} = \boldsymbol{\sigma}\left(\boldsymbol{h_i^n} + \sum_{\boldsymbol{r \in R(i)}} \sum_{\boldsymbol{j \in N(i,r)}} \boldsymbol{f_r}\left(\boldsymbol{h_j^n}\right) \right), \quad (\mathbf{1})$$

где h_i^n – векторное представление узла i после слоя n; $R(i)$ возвращает типы отношений, в которых участвует узел i; $N(r,i)$ – множество соседей узла i связанных через отношение r; f_r – функция преобразования, зависящая от типа отношения, параметризованная нейронной сетью; σ – функция активации.

В данной работе в качестве целевых задач предварительного обучения исследуются задачи, основанные на: 1) предсказании имён переменных и функций (GNN-NamePred); 2) предсказании связей между узлами графа (GNN-EdgePred); 3) классификации типов связей в графе (GNN-TransR); 4) классификации типов узлов в графе (GNN-NodeClf). Разметку для обучения можно сгенерировать автоматически с помощью простых правил.

Задача предварительного обучения GNN-NamePred использует две составляющие: векторные представления узла и имени. Для вычисления представления узла используется рекурсивная формула (1). Для того чтобы увеличить утилизацию параметров, осуществлена токенизация имён. Каждый токен имени представлен своим уникальным вектором. Токены, участвующие в графе, и токены, полученные из имён, представлены разными векторами. Векторное представление имени вычисляется с помощью выражения

$$v_{name} = \sum_{s \in S_{name}} v_s,$$

где S_{name} – набор токенов для данного имени, а v_s — векторное представление токена S.

В качестве целевой функции используется MarginLoss, ранее применяемая в [42]. В качестве негативных примеров используются имена, присутствующие в кодовой базе, на которой построен граф. Целевая функция имеет вид

$$loss(v_h, v_t, y) = \begin{cases} 1 - \cos(v_h, v_t), & if \;\; y = 1 \\ \max(0, \cos(v_h, v_t)) - margin, & if \;\; y = -1 \end{cases},$$

где y – метка положительного или отрицательного примера.

Задача предварительного обучения GNN-EdgePred типична при обучении автокодировщика на основе графовой нейронной сети. Используя векторное представление узла, необходимо определить, с какими другими узлами в графе он связан. Подобная целевая функция ранее использовалась в других работах посвященных тренировке векторных представлений для исходного кода [5]. В задаче GNN-EdgePred может быть несколько правильных кандидатов. В процессе тренировки позитивный пример выбирается согласно равномерному распределению из списка соседей $N(i)$ узла i. Негативные примеры выбираются случайным образом. При тренировке используется та же целевая функция, что и для задачи GNN-NamePred.

Задача предварительного обучения GNN-TransR заимствована из методов тренировки реляционных векторных представлений. Сами векторные представления вычисляются с помощью рекурсивной формулы (1), а в качестве целевой функции используется TransR ($-|M_r v_h + v_r - M_r v_t|$, где v_h и v_t – векторные представления узлов, соединенных отношением r, v_r – векторное представление отношения, M_r – дополнительная матрица параметров). Этот подход предложен в данной работе и ранее не исследовался для создания предварительно обученных векторных представлений для исходного кода.

Задача предварительного обучения GNN-NodeClf тренируется предсказывать типы узлов в графе. Целевая функция для данной задачи определена выражением

$$loss(v_i, y) = -\sum_{c \in C} y \log(f_c(v_i)),$$

где v_i это векторное представление узла, y – тип узла, C – множество возможных типов узлов, $f_c(v_i)$ – функция оценки правдоподобия принадлежности узла i к классу c. В большинстве случаев тип узла возможно определить по типам связей с соседними узлами. Целевая функция GNN-NodeClf подходит для обучения векторных представлений, которые кодируют локальную структуру в графа.

3.5 Классификация типов переменных

Задачу классификации типов можно формализовать следующим образом. Дан набор токенизированных функций F с разметкой типов переменных L. Типы определены для переменных, обозначенных в сигнатуре функции. Для одной переменной разметка представлена в виде тройки $(start, end, type)$, где $start$ это стартовый токен, end – конечный токен, а $type$ – тип переменной. Набор типов определён в момент тренировки и не может быть расширен. Для каждого токена в соответствие поставлен идентификатор узла в графе, к которому он относится. Цель задачи – определить правильный тип переменной из числа заранее определённых типов.

Для решения этой задачи предложено два подхода. Первый предполагает использование ранее обученных векторных представлений узлов, ассоциируемых с переменными, для классификации типов этих переменных. Второй подход, предполагает одновременное использование представления исходного кода в виде последовательности токенов и в виде графа, и соответствующих векторных представлений. В данной работе сравниваются эти два похода.

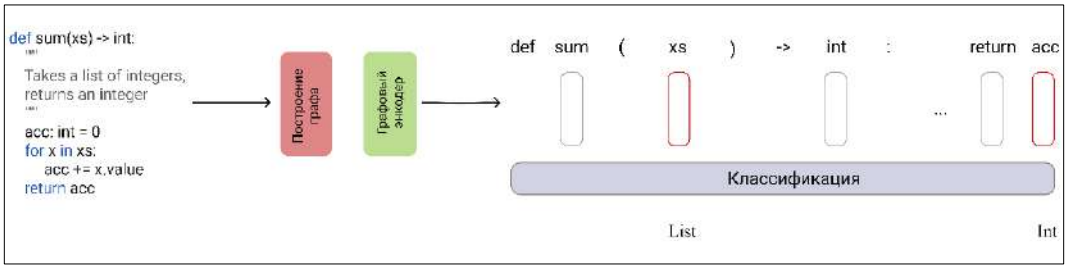

Рис. 3. Архитектура модели TypeClf-Graph для классификации типов переменных. Исходный код представлен в виде графа. Векторные представления переменных (обозначены красной рамкой) передаются на вход классификатора типов

Fig. 3. Architecture of the TypeClf-Graph model for classifying variable types. Source code is represented in the form of a graph. Vector representations of variables (indicated by a red frame) are passed to the input of the type classifier

Первый подход, названный TypeClf-Graph, заключается в классификации типов на основе предварительно обученных векторных представлений. Для классификации типа переменной используется лишь векторное представление соответствующего узла. Выбор узла для классификации является простой задачей, так как всегда известно какие именно узлы в графе соответствуют переменным. Векторное представление подаётся на вход простого классификатора, реализованного при помощи полносвязной нейронной сети. При этом сами векторные представления на этапе тренировки классификатора не обновляются. Архитектура такой модели показана на рис. 3.

Второй подход, названный TypeClf-Hybrid, заключается в классификации типов на основе гибридной модели. Такая модель объединяет в себе представление исходного кода в виде последовательности и в виде графа, что позволяет использовать существующие модели из области обработки естественного языка. Схема работы предложенной гибридной модели представлена на . . В основе лежит текстовый кодировщик, на вход которого исходный код поступает в виде последовательности токенов. Каждому токену в соответствие может быть

поставлен узел из графа. Токены, не несущие семантической нагрузки, могут не иметь соответствующих им узлов в графе (например скобки). В качестве модели обработки токенов могут использоваться любые модели для обработки текста. В данной работе проведены эксперименты со свёрточной моделью TextCNN, основанной на работе [55], и с предварительно обученной моделью CodeBERT, в том числе качестве базовых моделей.

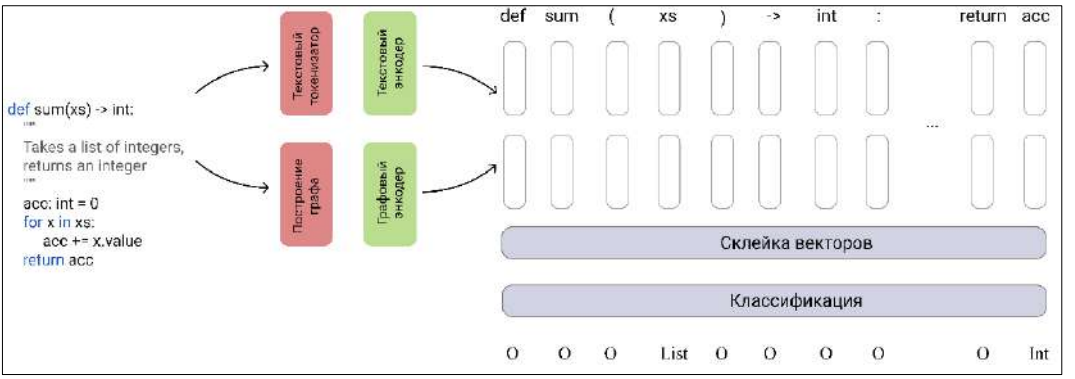

Рис. 4. Архитектура гибридной модели TypeClf-Hybrid. Исходный код представлен в виде последовательности токенов и в виде графа. Два вида векторных представлений конкатенируются и передаются на вход классификатора типов

Fig. 4. Architecture of the TypeClf-Hybrid model. Source code is represented in the form of a sequence of tokens and in the form of a graph. Two kinds of vector representations are concatenated and passed to the input of the type classifier

4. Результаты экспериментов

4.1 Описание наборов данных

В рамках данной работы были созданы два набора данных. Процедура получения наборов данных реализована в составе программного комплекса SourceCodeTools. Новые наборы данных создавались потому, что существующие не содержат информации о глобальных связях в исходном коде. Первый – составлен на основе популярных пакетов Python (далее обозначен PopularPackages, процедура сбора описана в [56]). Второй – основан на наборе данных CodeSearchNet[3] (далее обозначен CSN-Graph). Оба набора данных опубликованы в свободном доступе[4]. Статистика наборов данных приведена в табл. 1. Предварительная тренировка занимает существенное количество времени. Для увеличения количества поведённых экспериментов в основном используется набор данных PopularPackages, который значительно меньше по размеру.

Табл. 1. Статистика наборов данных
Table 1. Dataset statistics

Параметр	Значение
Набор данных PopularPackages	
Количество пакетов в обучающей выборке	142
Средняя степень вершины	6,45
Общее число узлов	2 652 787
Общее число рёбер	10 587 447
Средняя глубина AST	8,38

[3] https://github.com/github/CodeSearchNet
[4] https://disk.yandex.ru/d/GUxvRSPvFxop7g

Параметр	Значение
Набор данных CSN-Graph	
Количество пакетов в обучающей выборке	9 625
Средняя степень вершины	-
Общее число узлов	46 479 185
Общее число рёбер	216 697 812
Средняя глубина AST	16,76

4.2 Набор данных для классификации типов переменных

Для тренировки модели классификации типов переменных был подготовлен ещё один набор данных, основанный на наборе данных PopularPackages. Это решение обусловлено тем, что в числе прочих проводятся эксперименты с реляционными векторными представлениями, которые не имеют возможности обобщаться на новые данные. Чтобы предотвратить утечку информации, все аннотации типов и значения по умолчанию исключены из кодовой базы, на основе которой строится граф. В экспериментах используется две версии набора данных. Первая содержит все возможные типы переменных, а вторая – только 20 самых частых типов, которые составляют 86% всего набора данных.

Финальный набор данных для классификации типов переменных содержит 2938 примеров аннотаций типов. Количество уникальных типов велико из-за того, что в Python существует возможность определения параметрических типов. Чтобы увеличить среднее количество примеров для каждого уникального типа, решено в качестве целевого класса использовать имя основного типа без учёта параметров. Например, тип List[int] упрощается до List. В результате получен набор данных, содержащий 2767 примеров и 89 уникальных меток классов. Аналогичный подход для упрощения типов был применен в [23]. Метрикой, используемой для оценки качества, является HITS@1, показывающая насколько часто правильный тип является первым среди списка предлагаемых типов. Данная метрика эквивалентна точности классификации.

4.3 Используемые вычислительные ресурсы

Для тренировки моделей машинного обучения использовался компьютер с процессором Intel Core i7-7700K, 32Гб оперативной памяти и видеокартой NVIDIA 1080ti (12 Гб). Время предварительной тренировки модели графовой нейронной сети составляет около 1 часа для набора данных PP и 10 дней для набора данных CodeSearchNet-Graph. Время тренировки одной модели реляционных векторных представлений на наборе данных PP размерностью 500 – 8 дней. Время тренировки гибридной модели (TypeClf-Hybrid) без дообучения (300 эпох) – 4 часа.

4.4 Оценка графовых векторных представлений с помощью модели TypeClf-Graph

В данном эксперименте проводится оценка полезности графовых векторных представлений для решения задачи классификации типов. Нейро-классификатор TypeClf-Graph используется для того, чтобы определить, содержат ли векторные представления узлов информацию, ассоциируемую с типами переменных. Классификатор типов представляет собой простую полносвязную нейронную сеть с двумя скрытыми слоями размером 30 и 15. В качестве функции активации используется ReLU. Результат обучения модели классификатора TypeClf-Graph с использованием различных графовых векторных представлений показан в табл. 2. Максимальная точность, достигаемая классификатором реляционных векторных представлений, составляет 52,85 (RotatE). Максимальная точность,

достигаемая классификатором векторных представлений GNN, составляет 59,01 (GNN-NamePred). В качестве базовых моделей используются случайно сгенерированные векторные представления, а также векторные представления FastText [57] (натренированы в рамках данной работы на наборе данных CodeSearchNet со стандартными параметрами, используя библиотеку Gensim), рассчитанные для имён классифицируемых переменных. Все предварительно обученные векторные представления фиксированы и не обновляются в процессе тренировки.

Табл. 2. Точность классификации типов (Hits@k) с помощью модели TypeClf-Graph. Эксперименты повторялись 5 раз. В качестве базовых моделей используются случайно инициализированные вектора (без обучения), а также векторные представления FastText. Размерность всех векторных представлений равна 100

Table 2. Accuracy of type classification (Hits@k) using the TypeClf-Graph model. The experiments were repeated 5 times. Randomly initialized vectors (without training) as well as FastText vector representations are used as base models. The dimension of all vector representations is 100

Метод предварительного обучения	**Hits@1**	**Hits@3**	**Hits@5**
Все типы			
Случайные вектора	12.95±1.72	26.14±3.89	36.14±5.55
FastText	**61.29±0.71**	**79.28±0.95**	**85.86±0.42**
DistMult	45.72±1.99	67.02±2.48	76.98±2.02
RotatE	52.22±1.13	72.11±0.96	80.40±1.09
ComplEx	47.50±2.20	69.58±1.94	77.95±2.06
DistMult $k-hop$	50.51±1.58	72.11±1.46	79.99±2.30
RotatE $k-hop$	**52.85±0.59**	**73.34±1.55**	**80.82±0.90**
ComplEx $k-hop$	49.77±0.94	72.15±2.74	80.40±0.83
GNN-NamePred	**59.01±1.15**	**74.36±0.89**	**80.31±0.95**
GNN-EdgePred	56.81±0.83	73.85±1.08	79.88±1.21
GNN-TransR	46.18±2.34	64.40±2.42	73.58±2.12
GNN-NodeClf	49.05±1.94	68.58±2.51	75.98±1.49
Частые типы			
Случайные вектора	23.88±3.53	44.47±5.13	55.59±7.11
FastText	**65.99±0.97**	**84.25±1.16**	**90.11±1.05**
DistMult	52.33±1.07	74.74±2.13	84.72±1.47
RotatE	59.12±1.58	79.77±1.92	88.78±1.88
ComplEx	53.84±1.12	77.04±1.25	86.41±0.77
DistMult $k-hop$	57.99±1.02	79.34±1.24	88.50±0.63
RotatE $k-hop$	**60.21±1.36**	80.17±2.29	88.58±1.51
ComplEx $k-hop$	56.78±1.41	**80.42±1.95**	**89.02±1.42**
GNN-NamePred	**64.86±1.91**	78.35±1.11	83.83±0.68
GNN-EdgePred	60.79±1.26	**78.45±1.25**	**84.21±1.50**
GNN-TransR	52.92±1.63	71.56±2.55	81.45±2.06
GNN-NodeClf	54.47±1.83	74.84±1.96	82.15±0.86

Из результатов видно, что реляционные векторные представления справляются с задачей классификации типов хуже, чем векторные представления GNN. Можно предположить, что одной из причин для этого является лежащая в основе GNN парадигма передачи сообщений. Переменные, используемые в схожих контекстах, получают сообщения от похожих узлов.

Как следствие, переменные с одинаковым шаблоном использования, имеют схожие векторные представления. Векторные представления FastText, подсчитанные для имён классифицируемых переменных, позволяют достичь самых лучших показателей классификации типов.

4.5 Анализ точности классификации типов

Первая часть анализа заключается в определении зависимости между точностью классификации типа и его частотой (см. рис. 5). Наблюдается три кластера типов. Первый содержит типы вроде object, List, float, Dict, str. Для данного кластера наблюдается увеличение качества классификации с увеличением частотности типа. Второй кластер содержит типы Union, Optional, Any, bytes, bool. Для этих типов характерна низкая точность классификации несмотря на высокую частотность. Третий кластер содержит типы Resolver, Writer, CodeWrite и др. Эти типы являются редкими, но для них наблюдается идеальная точность классификации.

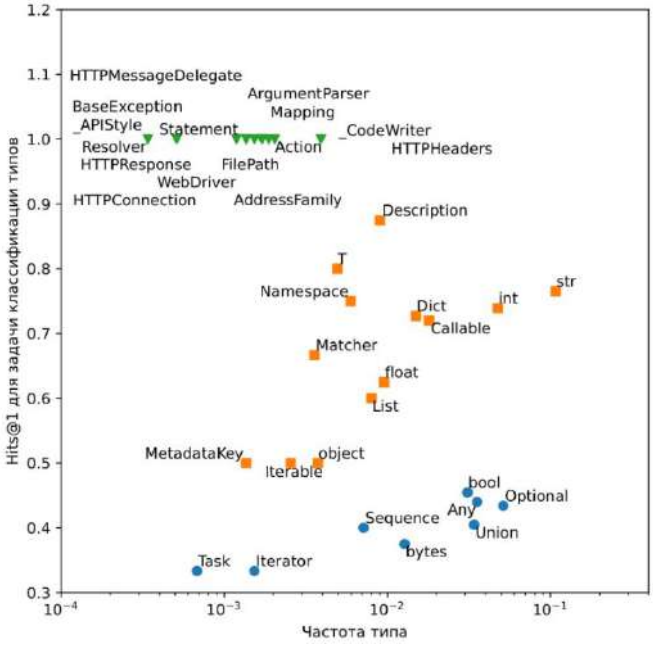

Рис. 5. Точность классификации типов в зависимости от частоты типа. Наблюдаются три кластера: 1) редкие типы, для которых точность классификации высока (FilePath, WebDriver); 2) частые типы, для которых точность классификации низкая (Union, bytes); 3) типы, для которых наблюдается увеличение точности с увеличением частоты (List, int)

Fig. 5. Type classification accuracy depending on type frequency. Three clusters are observed: 1) rare types for which the classification accuracy is high (FilePath, WebDriver); 2) frequent types for which the classification accuracy is low (Union, bytes); 3) types for which there is an increase in accuracy with increasing frequency (List, int)

Вторая часть анализа заключается в анализе матрицы расхождения. Ошибки часто совершаются в пользу частых типов, например, str (22% от всех аннотированных примеров), а также в пользу неоднозначных типов, таких как Optional (10%), Any (7%), Union (6%), Sequence (1.4%), T (0.9%), object (0.7%). Было решено протестировать модель классификации типов после исключения неоднозначных типов. Результат такого теста показан в табл. 3. В результате исключения неоднозначных типов, для векторных представлений GNN-NamePred точность классификации выросла на 35%, а для векторных представлений FastText на 32%.

Табл. 3. Метрика Hits@k классификации частых типов за исключением неоднозначных типов
Table 3. Hits@k metric for classifying common types except for ambiguous types

Метод предварительного обучения	Hits@1	Hits@3	Hits@5
GNN-NamePred	79.93	88.48	91.11
FastText	**81.24**	**93.22**	**96.44**
GNN-NamePred, размерность 500	78.88	87.56	91.77
FastText, размерность 500	82.17	92.96	95.46
CodeBERT, размерность 786	**84.53**	**94.21**	**96.25**

4.6 Анализ влияния размерности на точность классификации типов

В рамках данного эксперимента сравнивается точность классификации переменных при использовании графовых векторных представлений разной размерности. Увеличение количества параметров модели может приводить к улучшению качества работы. В качестве базовой модели используются векторные представления CodeBERT, имеющие размерность 768. Модель CodeBERT доступна из репозитория библиотеки transformer. Результаты данного эксперимента приведены на рис. 6.

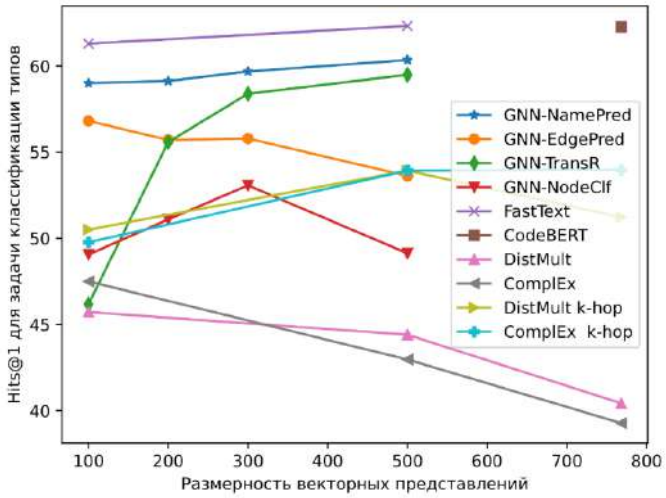

Рис. 6. Точность классификации типов в зависимости от размерности векторных представлений (оценка на всех типах)
Fig. 6. Accuracy of type classification depending on the dimension of vector representations (assessment on all types)

Результаты экспериментов показывают, что качество решения задачи классификации типов улучшается при увеличении размерности векторных представлений GNN-NamePred и GNN-TransR. Для векторных представлений GNN-EdgePred и GNN-NodeClf стабильного улучшения качества классификации при увеличении размерности не наблюдается. Для реляционных векторных представлений при увеличении размерности наблюдается улучшение качества при использовании графа с k-hop рёбрами, и ухудшение – при использовании стандартного графа.

В табл. 4 приведено значение метрик классификации Hits@k для всех типов и для частых типов при использовании векторных представлений размерностью 500. Среди реляционных векторных представлений лучшее качество классификации достигается при обучении на графе, содержащим k-hop рёбра (метод тренировки RotatE не рассматривался ввиду длительного времени тренировки). Среди векторных представлений GNN лучшие показатели

всегда достигаются при использовании GNN-NamePred. Тем не менее векторные представления FastText и CodeBERT позволяют достичь наилучших результатов классификации типов.

Табл. 4. Точность классификации типов Hits@k для графовых векторных представлений (размерность 500), FastText (размерность 500) и векторных представлений CodeBERT (размерность 768)

Table 4. Classification accuracy of Hits@k types for graph vector representations (dimension 500), FastText (dimension 500) and CodeBERT vector representations (dimension 768)

Метод предварительного обучения	Hits@1	Hits@3	Hits@5
Все типы			
DistMult 500	42.45±0.67	60.81±2.75	69.99±2.41
ComplEx 500	41.85±1.55	60.48±1.93	68.77±1.4
DistMult $k-hop$ 500	53.93±1.62	**74.27±1.98**	**81.32±1.78**
ComplEx $k-hop$ 500	**54.19±1.35**	73.75±1.93	80.66±2.00
GNN-NamePred 500	**60.35±0.95**	**76.18±1.12**	**81.49±0.93**
GNN-EdgePred 500	53.62±1.65	69.40±1.74	76.45±0.99
GNN-TransR 500	59.48±0.59	75.70±0.79	81.41±1.00
GNN-NodeClf 500	49.13±2.85	69.21±2.33	76.88±1.19
FastText 500	62.31±0.54	79.52±1.18	**86.33±0.63**
CodeBERT	**65.66±0.76**	**79.60±0.89**	85.00±1.10
Частые типы			
DistMult 500	50.58±2.23	72.01±2.16	80.73±1.35
ComplEx 500	50.19±1.01	71.57±2.17	81.59±2.13
DistMult $k-hop$ 500	61.29±0.98	**81.99±0.90**	**89.19±1.21**
ComplEx $k-hop$ 500	**61.42±0.92**	81.68±1.05	88.75±1.02
GNN-NamePred 500	**65.94±0.31**	**80.88±0.97**	**86.55±0.48**
GNN-EdgePred 500	57.79±1.49	75.36±1.76	82.34±1.54
GNN-TransR 500	62.01±1.77	78.26±1.33	84.49±1.50
GNN-NodeClf 500	55.40±4.12	75.40±1.63	83.37±1.69
FastText 500	65.14±0.72	82.33±2.67	88.13±1.33
CodeBERT	**68.29±1.54**	**84.44±1.13**	**90.63±1.12**

4.7 Оценка влияния длительности тренировки на качество классификации типов

В данном эксперименте выявляется как длительность предварительной тренировки влияет на качество классификации типов. Вначале такой анализ проводится для реляционных векторных представлений. Зависимость точности классификации типов от длительности предварительной тренировки представлена на рис. 7. С увеличением времени тренировки точность классификации типов улучшается, что говорит об улучшении качества векторных представлений. Векторные представления, предварительно обученные методами TransR и RESCAL, демонстрируют самую низкую точность классификации. Самые лучшие результаты получены методом тренировки RotatE при использовании графа с k-hop рёбрами. Однако тренировка такой модели требует значительных ресурсов. По этой причине модель RotatE редко использовалась в других экспериментах. Можно заметить, что точность классификации типов, как правило, выше при использовании графов с k-hop рёбрами

(исключение составляют векторные представления RotatE, натренированные в течение 100 эпох).

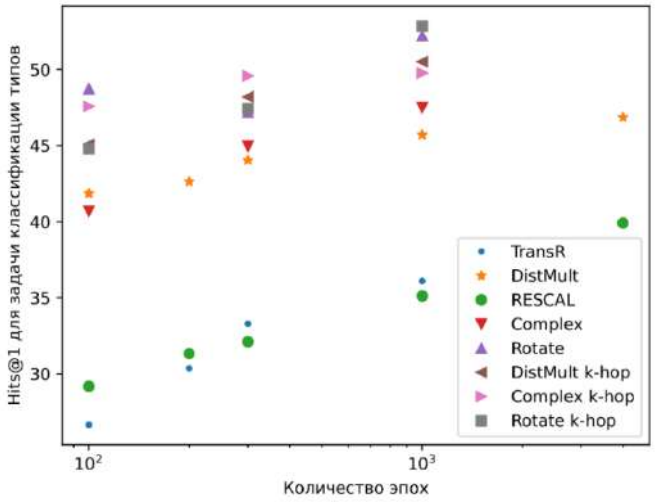

Рис. 7. Зависимость точности классификации типов с помощью реляционных векторных представлений от длительности предварительной тренировки. С увеличением времени тренировки точность классификации улучшается. Точность классификации, как правило, лучше при использовании k-hop векторов (исключение – вектора RotatE, натренированные в течение 100 эпох)
Fig. 7. Relationship between the accuracy of type classification using relational vector representations and the duration of the pretraining. As the pretraining time increases, the classification accuracy improves. Classification accuracy is generally better when using k-hop vectors (the exception is RotatE vectors trained for 100 epochs)

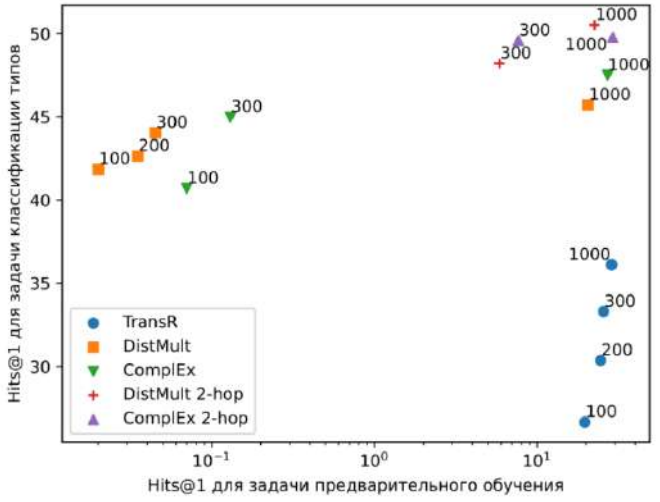

Рис. 8. Влияние длительности предварительного обучения реляционных векторных представлений на точность решения задачи предварительного обучения и задачи классификации типов. Число возле маркера обозначает количество эпох, использованное при тренировке
Fig. 8. Relationship between the duration of pretraining of relational vector representations and the accuracy of solving the pretraining task and the task of type classification. The number next to the marker indicates the number of epochs used during pretraining

Для проверки того, являются ли используемые задачи предварительного обучения полезными для тренировки векторных представлений для исходного кода, проведена проверка зависимости точности решения задачи предварительного обучения и точности классификации типов с использованием полученных векторных представлений. Данная зависимость для реляционных векторных представлений представлена на рис. 8. Наблюдается тенденция, при которой увеличение длительности тренировки (количество эпох) ведёт к увеличению точности решения задачи предварительной тренировки и точности классификации типов.

Далее, подобный анализ был проведён для векторных представлений GNN-NamePred, GNN-EdgePred и GNN-NodeClf (см. рис. 9). Исследование не проводилось для векторных представлений GNN-TransR ввиду ограниченных вычислительных ресурсов.

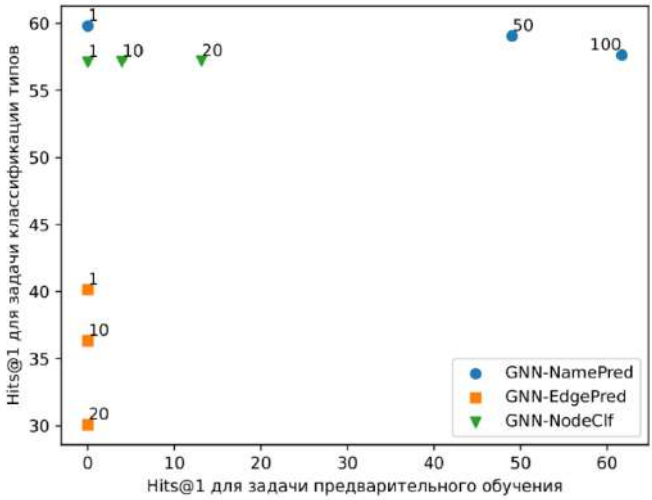

Рис. 9. Влияние длительности предварительного обучения GNN векторных представлений на точность решения задачи предварительного обучения и задачи классификации типов. Число возле маркера обозначает количество эпох, использованное при тренировке

Fig. 9. Relationship between the duration of pre-training of GNN vector representations and the accuracy of solving the pretraining task and the task of type classification. The number next to the marker indicates the number of epochs used during pretraining

Для векторных представлений GNN-NamePred наблюдается незначительный тренд, при котором качество решения задачи предварительного обучения улучшается, а задачи классификации типов незначительно ухудшается или остаётся неизменным.

Для векторных представлений GNN-NodeClf не наблюдается значительного улучшения точности классификации типов при улучшении точности решения задачи предварительно тренировки. В данном эксперименте в процессе предварительной тренировки использовалась скорость обучения, отличная от той, что использовалась для получения значений в табл. 2. С учётом значений из этих таблиц, при достижении точности решения задачи предварительного обучения близкой к единице (0,96), качество решения задачи классификации типов падает до 49,05.

Для векторных представлений GNN-EdgePred точность решения задачи предварительного обучения улучшается незначительно, и в то же время точность классификации типов падает. Данный результат указывает на интересный аспект векторных представлений GNN. Целевая задача GNN-EdgePred похожа на целевые задачи, используемые для тренировки реляционных векторных представлений, тем, что цель тренировки – научиться предсказывать наличие связей в графе на основе векторных представлений узлов. Одно из отличий заключается в том, что методы тренировки реляционных векторных представлений хранят

векторные представления узлов в явном виде, в то время как при использовании графовых нейронных сетей, векторные представления вычисляются на основе окружения узла. С учётом того, что GNN модель не справляется с решением задачи предварительного обучения, можно выдвинуть гипотезу о том, используемая графовая нейронная модель демонстрирует высокое смещение и низкую дисперсию.

4.8 Влияние составляющих графа на качество классификации типов

Граф, используемый для тренировки графовых векторных представлений, содержит большое количество различных типов рёбер. Не все эти рёбра имеют одинаковое влияние на качество финальных векторных представлений. В данной части исследования проводится изучение того, как исключение составляющих графа влияет на качество решения задачи классификации типов. Эксперименты проводились только для векторных представлений GNN. После удаления части графа, предварительная тренировка осуществлялась заново.

Результаты экспериментов представлены в табл. 5. В рамках первого эксперимента из графа были исключены метки типов рёбер. Таким образом, модель графовой нейронной сети R-GCN стала больше похожа на классическую графовую свёрточную нейронную сеть. При этом не наблюдается значительное изменение качества решения задачи классификации типов переменных. Данный факт может сигнализировать о том, что информация о типах рёбер в данный момент недостаточно утилизируется, или типы рёбер не так важны при решении задачи классификации типов. Данные результаты могут отличаться при проверке на других целевых задачах.

Табл. 5. Влияние исключения составляющих графа перед тренировкой графовых представлений GNN на качество решения задачи классификации типов
Table 5. Influence of excluding graph components before training GNN graph representations on the quality of solving the task of type classification

Модификация графа	Hits@1	Hits@3	Hits@5
Стандартный граф	59.01±1.15	74.36±0.89	80.31±0.95
Без глобальных связей	57.12±0.96	74.76±0.83	**80.82±1.32**
Без сабтокенов	58.26±0.46	74.44±0.54	80.43±1.4
Без типов рёбер	**59.33±0.45**	**74.8±1.08**	79.96±0.81

В рамках второго эксперимента из графа были исключены сабтокены. Без сабтокенов, в графе отсутствует информация об именах переменных. Несмотря на это, точность решения задачи классификации типов изменилась незначительно. Данный факт указывает на то, что классификация типов переменных осуществляется в основном за счёт использования структурных признаков, полученных из графа.

В третьем эксперименте из графа были исключены глобальные связи. Изначально предполагалось, что глобальные связи добавляют ценную информацию об использовании частых функций и методов, и могут значительно улучшить качество решения целевых задач. Результаты эксперимента показывают, что исключение глобальных рёбер из графа привело к снижению точности классификации типов, однако данное снижение мало для того, чтобы считать его значительным.

4.9 Влияние набора данных

В предыдущих экспериментах для предварительного обучения использовалась та же кодовая база, что и для классификации типов. Чтобы проверить, могут ли GNN модели для исходного кода обобщаться на новые данные, был подготовлен набор данных CSN-Graph. В данном эксперименте, из-за длительного времени, необходимого для обучения, было проведено сравнение только для моделей, обученных на задаче Name Prediction.

После предварительного обучения был проведён эксперимент по классификации типов с помощью подхода TypeClf-Graph. Результат Hits@1 для всех типов составил 58.85±0.71, а для частых типов – 63.20±0.67. Эти результаты практически совпадают с результатами, полученными при совместном использовании кодовой базы для предварительного обучения и классификации типов (см. табл. 6).

Табл. 6. Оценка метрики Hits@1 на наборе данных для классификации типов, используя векторные представления, предварительно обученные на наборах данных PP и CSN-Graph. Размерность векторных представлений равна 100
Table 6. Estimating the Hits@1 metric on the type classification dataset using vector representations pretrained on the PP and CSN-Graph datasets. The dimension of vector representations is 100

Метод предварительного обучения	Все типы	Частые типы
GNN-NamePred, Набор данных PP	**59.01±1.72**	**64.86±1.91**
GNN-NamePred, Набор данных CSN-Graph	58.85±0.71	63.20±0.67

4.10 Классификация типов с использованием гибридной модели

В данном эксперименте оценивается качество работы гибридной модели TypeClf-Hybrid для классификации типов. Гибридная модель использует одновременно текстовый кодировщик и графовые векторные представления. Были проведены эксперименты с двумя текстовыми кодировщиками: CNN и CodeBERT. На вход CNN модели подаются токены, их префиксы и суффиксы. В качестве векторных представлений для токенов используются вектора FastText, обладающие размерностью 100, предварительно обученные на Python программах из набора данных CodeSearchNet. Вектора для суффиксов и префиксов имеют размерность 50. Они обучаются во время тренировки модели. CNN модель состоит из трёх свёрточных слоёв, обладающих размерностью 40. Допустимое количество токенов в одной последовательности, подаваемой на вход текстового кодировщика, равно 512. Ширина окна CNN равна 10. Векторные представления, полученные на выходе текстового кодировщика, склеиваются с графовыми векторными представлениями. Были проведены эксперименты и с моделями CNN, обладающими большим количеством параметров. Однако увеличение числа параметров приводит к переобучению.

Качество работы гибридной модели TypeClf-Hybrid рассматривается на двух задачах: классификация типов и локализация + классификация. В первой задаче известно местоположение переменных, которые нужно классифицировать. Во второй задаче нужно сначала определить, какие токены должны быть классифицированы.

Ожидается, что гибридная модель с графовыми векторными представлениями должна работать не хуже, чем модель TypeClf-Graph. Кроме того, точность решения задачи локализация + классификация должна быть ниже, чем при решении только задачи классификации.

Табл. 7. Эффективность классификации типов с помощью гибридной модели и реляционных векторных представлений. Реляционные векторные представления имеют размерность 500
Table 7. Type classification using a hybrid model and relational vector representations. Relational vector representations have a dimension of 500

Модель классификации типов	Hits@1 Все типы	Hits@1 Частые типы
CNN, C	58.01±2.1	66.61±0.9
CNN, CL	55.01±1.7	63.33±2.5
CNN + DistMult, C	48.13±2.2	57.26±4.3
CNN + DistMult $k-hop$, C	**52.67±2.0**	**59.89±1.3**
CNN + ComplEx, C	51.27±6.2	55.97±3.9
CNN + ComplEx $k-hop$, C	52.12±2.2	58.98±1.4

Модель классификации типов	**Hits@1 Все типы**	**Hits@1 Частые типы**
CNN + DistMult, CL	-	-
CNN + DistMult $k-hop$, CL	-	-
CNN + ComplEx, CL	-	-
CNN + ComplEx $k-hop$, CL	-	-
CodeBERT, C	62.27±0.2	72.76±0.6
CodeBERT, CL	56.75±0.5	67.48±1.1
CodeBERT + DistMult, C	66.44±0.2	71.54±0.5
CodeBERT + DistMult $k-hop$, C	**66.78±0.4**	71.68±0.4
CodeBERT + ComplEx, C	66.30±0.7	**71.81±0.3**
CodeBERT + ComplEx $k-hop$, C	66.40±0.5	71.64±0.4
CodeBERT + DistMult, CL	55.92±0.3	60.99±0.9
CodeBERT + DistMult $k-hop$, CL	**56.08±0.6**	61.46±1.1
CodeBERT + ComplEx, CL	56.29±0.1	**61.70±0.6**
CodeBERT + ComplEx $k-hop$, CL	55.90±0.5	61.13±0.3

В табл. 7 приведены результаты оценки гибридной модели, использующей реляционные векторные представления (размерность векторных представлений 500). В качестве базовой модели используются текстовая модель CNN и CodeBERT. Рассматривались только модели графовых векторных представлений DistMult и ComplEx, так как они показали лучшие результаты при оценке модели TypeClf-Graph. Векторные представления RotatE с размерностью 500 не проверялись, так как их тренировка занимает слишком много времени.

Результаты показывают, что при использовании реляционных векторных представлений в связке с моделью CNN, лучшая точность классификации достигается при использовании модели DistMult, натренированной на графе с k-hop рёбрами. Не удалось натренировать гибридную модель, использующую CNN кодировщик и графовые векторные представления для решения задачи локализация + классификация, из-за нестабильностей в процессе тренировки. Модель, использующая только CNN кодировщик достигла более высокой точности классификации типов. При использовании реляционных векторных представлений в связке с кодировщиком CodeBERT, результаты классификации типов похожи для разных подходов тренировки векторных представлений. Точность классификации выше, чем при использовании только CodeBERT кодировщика. Точность локализации + классификации не меняется при добавлении реляционных векторных представлений.

Табл. 8. Эффективность классификации типов с помощью гибридной модели и векторных представлений GNN. Графовые векторные представления имеют размерность 500
Table 8. Type classification using hybrid model and GNN vector representations. Graph vector representations have a dimension of 500

Модель классификации типов	**Hits@1 Все типы**	**Hits@1 Частые типы**
CNN, C	58.01±2.1	66.61±0.9
CNN, CL	55.01±1.7	63.33±2.5
CNN + GNN-NamePred, C	65.48±0.8	71.89±0.7
CNN + GNN-EdgePred, C	65.25±0.9	69.49±0.9
CNN + GNN-TransR, C	64.91±1.3	68.63±1.6
CNN + GNN-NodeClf, C	61.60±0.7	68.32±1.8

Модель классификации типов	Hits@1 Все типы	Hits@1 Частые типы
CNN + FastText, C	**67.78±1.2**	**77.06±0.0**
CNN + GNN-NamePred, CL	64.23±0.6	68.78±1.1
CNN + GNN-EdgePred, CL	62.47±0.9	67.04±1.9
CNN + GNN-TransR, CL	62.03±1.4	68.14±1.3
CNN + GNN-NodeClf, CL	58.35±1.0	66.27±0.7
CNN + FastText, CL	**65.60±3.9**	**72.80±0.6**
CodeBERT, C	62.27±0.2	72.76±0.6
CodeBERT, CL	56.75±0.5	67.48±1.1
CodeBERT + GNN-NamePred, C	68.50±0.2	74.53±0.2
CodeBERT + GNN-EdgePred, C	67.83±0.4	74.65±0.2
CodeBERT + GNN-TransR, C	65.66±0.4	73.97±0.3
CodeBERT + GNN-NodeClf, C	65.28±0.5	74.65±0.5
CodeBERT + FastText, C	**70.76±0.2**	**77.06±0.0**
CodeBERT + GNN-NamePred, CL	63.36±0.3	69.39±1.2
CodeBERT + GNN-EdgePred, CL	63.41±0.7	70.38±0.6
CodeBERT + GNN-TransR, CL	60.83±0.5	68.70±0.5
CodeBERT + GNN-NodeClf, CL	61.58±0.6	69.82±0.7
CodeBERT + FastText, CL	**66.41±1.1**	**71.56±0.6**

В табл. 8 приведены результаты экспериментов по оценке качества работы гибридной модели, использующей векторные представления GNN. Гибридная CNN модель достигает точности классификации, схожей с простым классификатором TypeClf-Graph. Более того, точность классификации на частых типах стабильно выше и сравнима с CodeBERT. При классификации всех типов, точность классификации гибридной модели выше, чем модели, использующей только CNN кодировщик или только CodeBERT. На частых типах разница в точности классификации гораздо меньше. Как и ожидалось, задача локализация + классификация является более сложной и приводит к более низким значениям метрики Hits@1. Добавление графовых векторных представлений дает незначительное улучшение при использовании CodeBERT в качестве текстового кодировщика. В целом, точность классификации гибридной моделью, использующей векторные представления GNN выше, чем при использовании реляционных векторных представлений. Однако при использовании векторных представлений FastText можно получить ещё более высокие результаты.

4.11 Влияние предварительного обучения на скорость тренировки

Рис. 10 показывает выигрыш от использования векторных представлений GNN для каждой эпохи. Было проведено сравнение всех гибридных моделей, представленных в табл. 8. Чтобы оценить влияние на динамику обучения, сравнились показатели Hits@1 для каждой эпохи. Для лаконичности приведены только данные о динамике обучения для векторных представлений GNN-NamePred. Для векторных представлений GNN, обученных другими целевыми функциями, динамика аналогична.

Можно заметить, что модели, которые используют графовые векторные представления обучаются быстрее. Для CNN модели, использование дополнительных векторов ускоряет тренировку на десятки эпох. После некоторого момента разница в качество работы перестает меняться, а модели, использующие GNN вектора, сходятся к более высокому значению точности классификации типов. Результаты CodeBERT улучшились более чем на 10%.

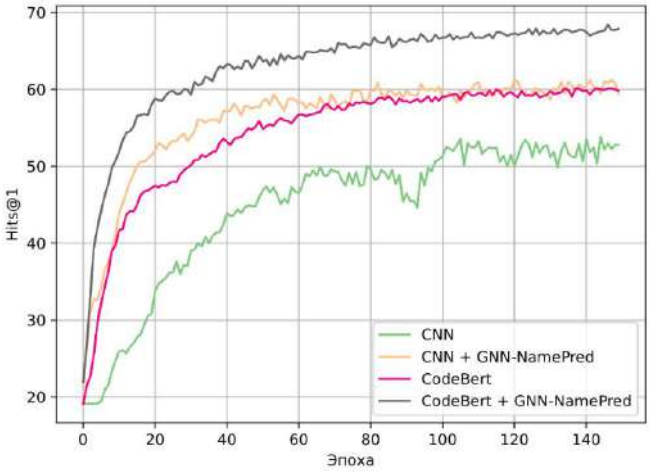

Рис. 10. Точность классификации типов моделью TypeClf-Hybrid в зависимости от эпохи для различных подходов тренировки векторных представлений GNN. Наблюдается улучшение от использования графовых векторных представлений вместе с текстовым кодировщиком по эпохам. При использовании векторов GNN модели обучаются быстрее, финальная точность выше
Fig. 10. Accuracy of type classification by the TypeClf-Hybrid model depending on the epoch for various approaches to training GNN vector representations. There is an improvement from using graphical vector representations along with a text epoch encoder. When using GNN vectors, models are trained faster, the final accuracy is higher

5. Заключение

Разработка предварительно обученных моделей является важным шагом на пути создания интеллектуальных приложений для анализа исходного кода. Большинство существующих предобученных моделей использует методы, созданные для обработки естественного языка. Графовые модели могу служить альтернативой используемым на данный момент подходам. Однако их свойства недостаточно изучены.

В данной статье проведено исследование применения предварительно обученных графовых векторных представлений для решения целевых задач, в частности классификации типов переменных в программах, написанных на языке Python. Было рассмотрено два типа векторных представлений: реляционных и обученных с помощью графовой нейронной сети.

Установлено, что при предварительном обучении графовых векторных представлений для исходного кода следует использовать графовые нейронные сети, так как они позволяют обобщаться на новые данные и показывают лучший результат по сравнению с реляционными векторными представлениями. При увеличении размерности, графовые векторные представления позволяют достичь точности классификации типов схожей с CodeBERT. Более того, совместное использование CodeBERT и графовых векторных представлений позволяет улучшить точность классификации.

Помимо CodeBERT, можно выделить и другие предварительно обученные модели для исходного кода, такие как GraphCodeBERT, UniXCoder и CodeT5. В отличие от CodeBERT, эти подходы в том или ином виде используют информацию из графа программы, которая может позволить сократить разрыв, наблюдаемый между CodeBERT и CodeBERT + GNN-NamePred. В данной работе проведено сравнение векторных представлений, полученных исключительно графовым и исключительно текстовым кодировщиками.

В дальнейшем следует оценить качество графовых векторных представлений на более широком круге задач, таких как поиск ошибок, поиск исходного кода, и генерация текстового описания.

Список литературы / References

[1] Vaswani A. Shazeer N. et al. Attention is all you need. In Proc. of the 31st Conference on Neural Information Processing Systems (NIPS), 2017, 11 p.

[2] Kanade A. Maniatis P. et al. Learning and evaluating contextual embedding of source code. In Proc. of the 37th International Conference on Machine Learning, 2020, pp. 5110–5121.

[3] Feng Z., Guo D. et al. CodeBERT: A Pre-Trained Model for Programming and Natural Languages. In Proc. of the Conference on Empirical Methods in Natural Language Processing, 2020, pp. 1536-.1547.

[4] Guo D., Ren S. et al. GraphCodeBERT: Pre-training Code Representations with Data Flow. In Proc. of the Ninth International Conference on Learning Representations, 2021, 18 p.

[5] Liu L., Nguyen H. et al. Universal Representation for Code. Lecture Notes in Computer Science, vol. 12714, 2021, pp. 16-28.

[6] Nguyen A.T., Nguyen T.N. Graph-Based Statistical Language Model for Code. In Proc. of the 37th IEEE International Conference on Software Engineering, 2015, pp. 858-868.

[7] Alon U., Sadaka R. et al. Structural language models of code. In Proc. of the 37th International Conference on Machine Learning, 2020, pp. 245-256.

[8] Yang Y., Chen X., Sun J. Improve Language Modelling for Code Completion by Tree Language Model with Tree Encoding of Context. In Proc. of the 31st International Conference on Software Engineering and Knowledge Engineering, 2019, pp. 675–680.

[9] Hellendoorn V.J., Sutton C. et al. Global Relational Models of Source Code. In Proc. of the Eighth International Conference on Learning Representations, 2020, 10 p.

[10] Pandi V., Barr E.T. et al. OptTyper: Probabilistic Type Inference by Optimising Logical and Natural Constraints. arXiv preprint arXiv:2004.00348, 2020, 29 p.

[11] Chirkova N., Troshin S. Empirical study of transformers for source code. In Proc. of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2021, pp. 703-715.

[12] Buratti L., Pujar S. et al. Exploring Software Naturalness through Neural Language Models. arXiv preprint arXiv:2006.12641, 2020, 12 p.

[13] Ahmad W.U., Chakraborty S. et al. Unified pre-training for program understanding and generation. In Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 2655–2668.

[14] Wang Y., Wang et al. CodeT5: Identifier-aware Unified Pre-trained Encoder-Decoder Models for Code Understanding and Generation. In Proc. of the Conference on Empirical Methods in Natural Language Processing, 2021, pp. 8696-8708.

[15] Guo D., Lu S. et al. UniXcoder: Unified Cross-Modal Pre-training for Code Representation. In Proc. of the 60th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, 2022, pp. 7212-7225.

[16] Karmakar A., Robbes R. What do pre-trained code models know about code? In Proc. of the 36th IEEE/ACM International Conference on Automated Software Engineering, 2021, pp. 1332-1336.

[17] Cui S., Zhao G. et al. PYInfer: Deep Learning Semantic Type Inference for Python Variables. arXiv preprint arXiv:2106.14316, 2021, 12 p.

[18] Hellendoorn V.J., Bird C. et al. Deep learning type inference. In Proc. of the 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2018, pp. 152-162.

[19] Malik R.S., Patra J., Pradel M. NL2Type: Inferring JavaScript Function Types from Natural Language Information, In Proc. of the IEEE/ACM 41st International Conference on Software Engineering (ICSE), 2019, pp. 304-315.

[20] Boone C., de Bruin N. et al. DLTPy: Deep Learning Type Inference of Python Function Signatures using Natural Language Context. arXiv preprint arXiv:1912.00680, 2019, 10 p.

[21] Pradel M., Gousios G. et al. Typewriter: Neural type prediction with search-based validation. In Proc. of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2020, pp. 209-220.

[22] Raychev V., Vechev M., Krause A. Predicting program properties from "Big Code". In Proc. of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, 2015, pp. 111-124.

[23] Allamanis M., Barr E.T. et al. Typilus: Neural type hints. Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2020, pp. 91-105.

[24] Peng Y., Gao C. et al. Static inference meets deep learning: a hybrid type inference approach for Python. In Proc. of the 44th International Conference on Software Engineering, 2022, pp. 2019-2030.
[25] Wei J., Goyal M. et al. LambdaNet: Probabilistic type inference using graph neural networks. In Proc. of the Eighth International Conference on Learning Representations, 2020, 11 p.
[26] Ye F., Zhao J., Sarkar V. Advanced Graph-Based Deep Learning for Probabilistic Type Inference. arXiv preprint arXiv:2009.05949, 2020, 25 p.
[27] Fernandes P., Allamanis M., Brockschmidt M. Structured Neural Summarization. In Proc. of the Seventh International Conference on Learning Representations, 2019, 18 p..
[28] Cvitkovic M., Singh B., Anandkumar A. Deep Learning On Code with an Unbounded Vocabulary. In Proc. of the Machine Learning for Programming (ML4P) Workshop at Federated Logic Conference (FLoC), 2018, 11 p..
[29] Dinella E., Dai H. et al. Hoppity: Learning Graph Transformations To Detect and Fix Bugs in Programs. In Proc. of the Eighth International Conference on Learning Representations, 2020, 17 p.
[30] Wang Y., Gao F. et al. Learning a static bug finder from data. arXiv preprint arXiv:1907.05579, 2019, 12 p.
[31] Zhou Y., Liu S. et al. Devign: Effective vulnerability identification by learning comprehensive program semantics via graph neural networks. In Proc. of the 33rd International Conference on Neural Information Processing Systems (NIPS), 2019, pp. 10197-10207.
[32] Brauckmann, A. Goens, S. Ertel and J. Castrillon. Compiler-based graph representations for deep learning models of code. In Proc. of the 29th International Conference on Compiler Construction, 2020, pp. 201-211.
[33] Wan Y., Shu J. et al. Multi-modal attention network learning for semantic source code retrieval. In Proc. of the 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2019, pp. 13-25.
[34] Wang W., Li G. et al. Detecting Code Clones with Graph Neural Networkand Flow-Augmented Abstract Syntax Tree. In Proc. of the IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER), 2020, pp. 261-271.
[35] Li Y., Wang S. et al. Improving Bug Detection via Context-Based Code Representation Learning and Attention-Based Neural Networks. Proceedings of the ACM on Programming Languages, vol. 3, issue OOPSLA, 2019, article no. 162, 30 p.
[36] Ben-Nun T., Jakobovits A.S., Hoefler T. Neural code comprehension: A learnable representation of code semantics. In Proc. of the 32nd International Conference on Neural Information Processing Systems (NIPS), 2018, pp. 3589-3601.
[37] DeFreez D., Thakur A.V., C. Rubio-Gonzalez A.V..Path-based function embedding and its application to error-handling specification mining, In Proc. of the 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2018, pp. 423-433, 2018.
[38] Brockschmidt M., Allamanis M. et al. Generative Code Modeling with Graphs. In Proc. of the Seventh International Conference on Learning Representations, 2019, 24 p.
[39] Lu D. Tan N. et al. Program classification using gated graph attention neural network for online programming service. arXiv preprint arXiv:1903.03804, 2019, 12 p.
[40] Zhang J., Wang X. et al. A Novel Neural Source Code Representation Based on Abstract Syntax Tree. In Proc. of the IEEE/ACM 41st International Conference on Software Engineering (ICSE), 2019, pp. 783-794.
[41] Allamanis M., Brockschmidt M., Khademi M. Learning to Represent Programs with Graphs. In Proc. of the 6th International Conference on Learning Representations (ICLR), 2018, 17 p.
[42] Hamilton W.L., Ying R., Leskovec J. Inductive representation learning on large graphs. In Proc. of the 31st International Conference on Neural Information Processing Systems (NIPS), 2017, pp. 1025-1035.
[43] Wang Z., Ren Z. et al. Robust embedding with multi-level structures for link prediction. In Proc.of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019, pp. 5240–5246.
[44] Schlichtkrull T.N., Kipf P. et al. Modeling Relational Data with Graph Convolutional Networks. Lecture Notes in Computer Science, vol. 10843, 2018, pp. 593-607.
[45] Cai, L. Yan B, et al. TransGCN: Coupling transformation assumptions with graph convolutional networks for link prediction. In Proc. of the 10th International Conference on Knowledge Capture (K-CAP), 2019, pp. 131-138.
[46] Liu X., Tan H. et al. RAGAT: Relation Aware Graph Attention Network for Knowledge Graph Completion. IEEE Access, vol. 9, 2021, pp. 20840–20849.

[47] Allamaras M., Chanthirasegaran P. et al. Learning continuous semantic representations of symbolic expressions. In Proc. of the 34th International Conference on Machine Learning, vol. 70, 2017, pp. 80–88.
[48] Kudo T., Richardson J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In. Proc. of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2018, pp. 66-71.
[49] Lin Y., Liu Z. et al. Learning entity and relation embeddings for knowledge graph completion. In. Proc. of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2181–2187.
[50] Yang B., Yih W. et al. Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575, 2014, 12 p.
[51] Nickel M., Tresp V., Kriegel H.-P. A three-way model for collective learning on multi-relational data. In Proc. of the 28th International Conference on International Conference on Machine Learning, 2011, pp. 809-816.
[52] Trouillon T., Welbl J. et al. Complex embeddings for simple link prediction. In Proc. of the 33rd International Conference on International Conference on Machine Learning, 2016, pp. 2071-2080.
[53] Sun Z., Deng Z.-H. et al. Rotate: Knowledge graph embedding by relational rotation in complex space. In Proc. of the Seventh International Conference on Learning Representations, 2019, 18 p.
[54] Ling X., Wu L. et al. Deep graph matching and searching for semantic code retrieval. ACM Transactions on Knowledge Discovery from Data (TKDD), vol. 15, issue 5, 2021, article no. 88, 21 p.
[55] Collobert R., Weston J. et al. Natural language processing (almost) from scratch. Journal of Machine Learning Research, vol. 12, 2011, pp. 2493-2537.
[56] Romanov V., Ivanov V., Succi G. Representing Programs with Dependency and Function Call Graphs for Learning Hierarchical Embeddings. In Proc. of the 22nd International Conference on Enterprise Information Systems (ICEIS), vol. 2, 2020, pp. 360-366.
[57] Bojanowski P., Grave E. et al. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, vol. 5, 2017, pp. 135-146.

Информация об авторах / Information about authors

Владимир Владимирович ИВАНОВ – кандидат физико-математических наук, доцент. Научные интересы: анализ данных, машинное обучение, разработка программного обеспечения, компьютерная лингвистика, обработка естественного языка, извлечение информации, анализ текста, надежность программного обеспечения и метрики программного обеспечения.

Vladimir Vladimirovich IVANOV – Candidate of Physical and Mathematical Sciences, Associate Professor. Research interests: data analysis, machine learning, software development, computer linguistics, natural language processing, information extraction, text analysis, software reliability and software metrics.

Виталий Анатольевич РОМАНОВ – аспирант. Работал исследователем и инструктором в университете Иннополис с 2016 года. Научные интересы: обработка естественного языка, компьютерная лингвистика, глубокое обучение, большие данные и генеративные модели.

Vitaly Anatolyevich ROMANOV – PhD student. Worked as a researcher and instructor at Innopolis University since 2016. Research interests: natural language processing, computational linguistics, deep learning, big data and generative models.