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.

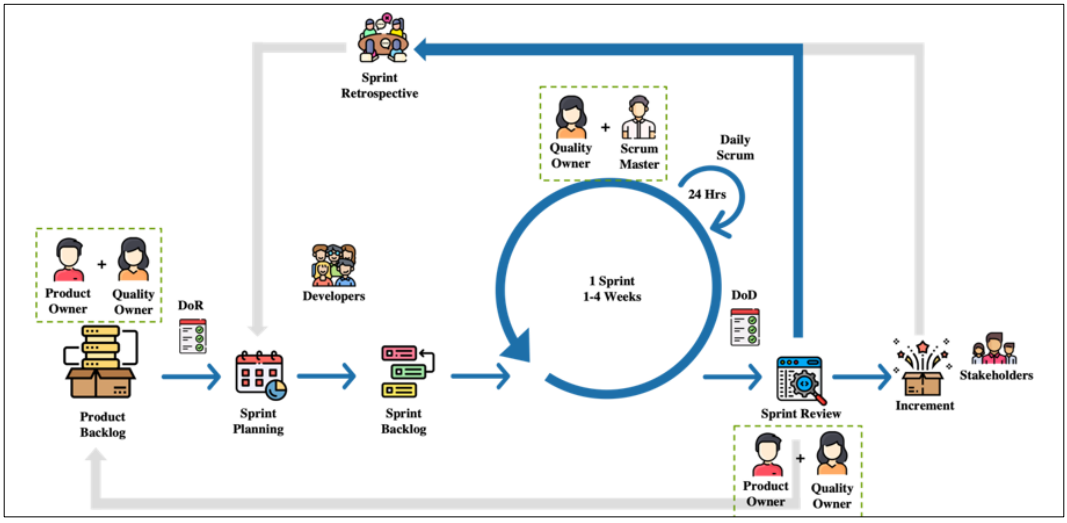

Fig. 1. Scrumlity Framework

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.