DOI: 10.15514/ISPRAS-2024-36(6)-6

Could an LLM Like chatGPT Perform a Functional Size Measurement using the COSMIC Method?

[1] F. Valdés-Souto, ORCID: 0000-0001-6736-0666 <fvaldes@ciencias.unam.mx>
[2] D. Torres-Robledo, ORCID: 0009-0002-7168-9709 <dtorres@ciencias.unam.mx>

[1] National Autonomous University of Mexico, Science Faculty, CDMX, Mexico.
[2] National Autonomous University of Mexico
Research Institute in Applied Mathematics and Systems, CDMX, Mexico.

Abstract. The process of developing software is intricate and time-consuming. Resource estimation is one of the most important responsibilities in software development. Since it is currently the only acceptable metric, the functional size of the program is used to generate estimating models in a widely accepted manner. On the other hand, functional size measurement takes time. The use of artificial intelligence (AI) to automate certain software development jobs has gained popularity in recent years. Software functional sizing and estimation is one area where artificial intelligence may be used. In this study, we investigate how to apply the concepts and guidelines of the COSMIC method to measurements using ChatGPT 4o, a large language model (LLM). To determine whether ChatGPT can perform COSMIC measurements, we discovered that ChatGPT could not reliably produce accurate findings. The primary shortcomings found in ChatGPT include its incapacity to accurately extract data movements, data groups, and functional users from the text. Because of this, ChatGPT's measurements fall short of two essential requirements for measurement: accuracy and reproducibility.

Keywords: COSMIC; CFP; functional size measurement; LLM; chatGPT; software engineering; AI; automatization.

For citation: Valdés-Souto F., Torres-Robledo D. Could an LLM like chatGPT perform a functional size measurement using the COSMIC method? Trudy ISP RAN/Proc. ISP RAS, vol. 36, issue 6, 2024, pp. 103-114. DOI: 10.15514/ISPRAS-2024-36(6)-6.

Может ли языковая модель на базе chatGPT измерять функциональный размер методом COSMIC?

[1] *Ф. Вальдес-Соуто, ORCID: 0000-0001-6736-0666 <fvaldes@ciencias.unam.mx>*
[2] *Д. Торрес-Робледо, ORCID: 0009-0002-7168-9709 <dtorres@ciencias.unam.mx>*

[1] *Национальный автономный университет Мексики, факультет Науки, Мехико, Мексика.*
[2] *Национальный автономный университет Мексики, Исследовательский институт прикладной математики и систем, Мехико, Мексика.*

Аннотация. Процесс разработки программного обеспечения является сложным и трудоемким. Оценка ресурсов является одной из наиболее важных обязанностей в разработке программного обеспечения. Поскольку в настоящее время это единственный приемлемый показатель, функциональный размер программы используется для генерации моделей оценки общепринятым способом. С другой стороны, измерение функционального размера требует времени. Использование искусственного интеллекта (ИИ) для автоматизации определенных рабочих мест разработчиков программного обеспечения набрало популярность в последние годы. Определение размеров и оценка функциональности программного обеспечения является одной из областей, в которой может использоваться искусственный интеллект. В этом исследовании мы исследуем, как применять концепции и рекомендации метода COSMIC к измерениям с использованием ChatGPT 4o, большой языковой модели (LLM). Чтобы определить, может ли ChatGPT выполнять измерения COSMIC, мы обнаружили, что ChatGPT не может надежно производить точные результаты. К основным недостаткам, обнаруженным в ChatGPT, относится его неспособность точно извлекать из текста движения данных, группы данных и функциональных пользователей. Из-за этого измерения ChatGPT не соответствуют двум основным требованиям к измерениям: точности и воспроизводимости.

Ключевые слова: метод измерения функционального размера COSMIC; функциональные точки COSMIC (CFP); измерение функционального размера; большая языковая модель (LLM); чат-бот chatGPT; программная инженерия; искусственный интеллект; автоматизация.

Для цитирования: Вальдес-Соуто Ф., Торрес-Робледо Д. Может ли языковая модель на базе chatGPT измерять функциональный размер методом COSMIC? Труды ИСП РАН, том 36, вып. 6, 2024 г., стр. 103–114 (на английском языке). DOI: 10.15514/ISPRAS–2024–36(6)–6.

1. Introduction

The competitive software development industry can address the estimating project problem by assessing functional size using a standard (COSMIC ISO/IEC 19761 is the only second-generation FSMM [1]); several estimation methods have been established, such as [2-3], and software development productivity can be measured.

For more than 70 years of research, software estimation has been a focus for numerous researchers since its inception in the 1950’s [4]. Precise estimation is a crucial component of software development and a key factor in project failure and has a significant effect on project planning and industrial budgets. [5-6], and [7].

The idea of automating software development chores with artificial intelligence (AI) has gained traction in recent years. Software functional sizing and estimation is one field in which artificial intelligence tries to demonstrate practical and accurate use. [6, 8-9]

The objective is to shorten the time required for measuring using the standard Functional Size Measurement Methods (FSMM). This will enable businesses to quickly estimate functional sizes by having the ability to measure user requirements accurately and promptly, most often provided in text format.

ChatGPT is one of the most advanced models of AI technology, offering some amazing and useful solutions in many fields, such as marketing [10], book creation/edition [11-12], graphic design [13], video creation/edition [14], music edition [15], and so forth. However, not every use has been effective; some attempts have led to pertinent failures or even instances of plagiarism [16].

In this article, we unbiasedly examine whether it is feasible to measure user requirements using ChatGPT 4o and determine whether it is not by providing a specific prompt that outlines the fundamentals of the COSMIC technique.

This paper's outline is as follows. Background information on software estimates and measurement, large language models, functional size measurement, and measurement repeatability is given in Section 2. The experimental protocol and its implementation are explained in Section 3. The data acquired were covered in Section 4, and the conclusions are finally covered in Section 5. Table 1 has the functional size measurement of the user requirements using the COSMIC method.

2. Background

2.1. Software measurement and estimation

The literature on software estimation has a wide range of techniques developed over more than six decades [4]. This has resulted in several estimation methods [2-3, 6-7], numerous classifications of these methods [5-7, 17-19], and various estimation process topologies [20-21]. Despite this extensive catalog of techniques, there is still no consensus on a single model that consistently produces accurate results for all industrial projects.

Even though regression-based estimating techniques based on reference databases predominate in the literature, it is not uncommon to find it difficult to reproduce research [9, 17, 22]. Several authors point out that measuring the size of the program is essential to the precision of approximations [23-26]. According to Fedotova et al. [4], the lack of a size variable may contribute to regression-based models' inability to perform well in estimation.

Neural Networks (NN) and other Machine Learning (ML) techniques have proven to be highly effective in producing accurate predictions, even in situations where noise has severely distorted the input data and the relationships between the inputs and outputs are complex [18].

The academic literature points to several difficulties in the subject, chief among them being the scarcity of real-world datasets (such as those from NASA, ISBSG, Desharnais, and COCOMO) [6]. The use of AI techniques is significantly hampered by this lack.

In the reviewed literature, only two approaches using AI to measure functional size were found, both employing the COSMIC standard [8, 27]. Ungan [8] has presented a technology that measures user requirements based on free-form text. To attain a "precise" measure, it necessitates clear and high-quality specifications, which makes it a closed source. Free-form requirements are by definition prone to being ambiguous, long, and incomplete, especially in the early stages of a project [28].

The other method involves measuring a reference case study using ChatGPT for the first time. This approach yields less than ideal results when applying COSMIC principles and rules; even in cases where the sizes are comparable, there are numerous errors in identifying data groups, data movements, and functional users based on the prompt requirements [27].

2.2 The LLM model: ChatGPT

A Large Language Model (LLM) is an artificial intelligence (AI) model designed to understand and generate human-like texts. LLMs are typically based on deep learning architectures, such as Transformers, and are trained on large amounts of text data to learn the patterns and structures of natural language. These models can perform a variety of language related tasks, including text generation, language translation, question answering, summarization, and more [29].

LLMs have demonstrated remarkable capabilities in "understanding" and generating text across different languages and domains. They are widely used in various applications, such as virtual assistants, chatbots, content generation, language translation services, and natural language processing tasks Examples of popular LLMs include OpenAI's GPT series (such as GPT-3, GPT3.5, and GPT-4) and Google's BERT.

ChatGPT is an LLM developed by OpenAI, specifically based on the GPT (Generative Pre-trained Transformer) architecture. It uses artificial intelligence to generate responses in text conversations [16]. The functioning of LLMs is based on two main phases: training and fine-tuning.

Text generation in LLMs, like ChatGPT, ***is based on the model's ability to predict the next word in a text sequence*** [16]. When given an input, the model evaluates the previous words and generates a list of possible next words and their associated probabilities. The word with the highest probability is selected, and the process repeats until the response is complete.

Considering the above, LLMs can perform ***a form of reasoning based on statistical and contextual patterns*** learned during training. The models do not have understanding or awareness but operate based on correlations and patterns in the training data.

2.3. Measurement of the functional size of software using COSMIC

Functional Size Measurement Methods (FSMM) are currently divided into two generations [1], with COSMIC ISO/IEC 19761 [28] being the only second-generation FSMM, the lessons learned from first-generation methodologies were the foundation for developing this standard [30]. The COSMIC Measurement Manual [28] presents all the guidelines, precepts, and examples required to carry out functional size measurements.

In real-world projects, approximating a functional size can be necessary in several situations [31]. These include: (1) when a size is required but not enough time or resources are available to measure using the standard method; (2) early in the project's life cycle, before the Functional User Requirements (FUR) have been detailed to the point where an accurate size measurement is possible; and (3) when the documentation quality of the actual requirements is inadequate for an accurate measurement. The functional size assessment can then be as accurate as feasible by using assumptions [32].

2.4. Reproducibility importance in metrology

In any scientific discipline, the validation of results is indispensable. Reproducibility allows other researchers to verify the findings of a study by replicating the same experiment or measurement under the same conditions. If the results can be reproduced, it reinforces the credibility and validity of the original work. This is particularly important in metrology, where the precision and accuracy of measurements directly impact the quality of technological products and services.

The reproducibility thus is crucial for the advancement of the discipline, actually software engineering is considered immature [1, 33].

3. Experimental procedure

To conduct this experiment, we utilized version 4o of ChatGPT (dated 08-08-2024) and the user requirements information from the C-Reg case study [34]. This case study provides the functional size measurement of the requirements for an actual system using the COSMIC method. For reference, the functional size of the C-Reg system is detailed in Table 1.

Fig. 1 illustrates the flow of the experimental procedure to obtain the results presented in Table 2. The first step involved creating a fine-tuned prompt based on the recommendations proposed by [16], which included incorporating some knowledge about COSMIC to be considered by ChatGPT.

The second step is to include the FUR for three functional processes described in the C-Reg system into the prompt and executed it using ChatGPT two times each.

Finally, we compare the measured size using chatGPT against the real measurements obtained in [34].

Table 1. C-Reg case study example for measuring COSMIC function points from requirements [34].

ID	*Functional Process*	CFP
1	Add teacher details	4
2	Consult a teacher's data	4
3	Modify a teacher's data	3
4	Delete data from a teacher	5
5	Consult the Course Offerings (Teacher)	7
6	Create assignments in Course Offerings	6
7	Modify assignments in Course Offerings	7
8	Delete assignments in Course Offerings	4
9	Add student data	4
10	Consult student data	4
11	Modify student data	3
12	Delete student data	4
13	Consult the Course Offerings (Student)	6
14	Create student schedule	6
15	Modify student schedule	8
16	Delete student schedule	6
17	Monitor Course Offering enrollment progress	6
18	Monitor enrollment progress on student schedules	5
19	Close registration	10
	Total	102

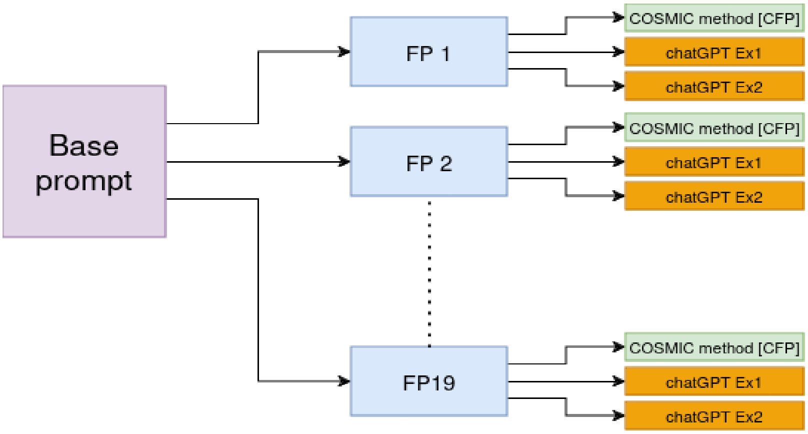

Fig. 1. Flow of experimental procedure diagram.

3.1 Fine-tuned prompt creation

To generate the prompt, it was necessary to describe some aspects of the COSMIC measurement method [28]. Firstly, we defined data groups, data movements, and the definition of functional users.

Creating and fine-tuning the ChatGPT prompt to create an effective functional size measurer involves carefully designing the model to categorize objects of interest, functional users, and the types of data movements [16].

Firstly, we ask to describe the given use case and to include the data groups and their movements according to the COSMIC measurement method [28].

Next, we describe and give examples of what a data movement is, what a functional user is, and what an object of interest is. The examples of these concepts improved how chatGPT classified the content of the use case.

Next, we gave hints on what systems mentioned in the requirements are beyond the measurement scope and must be considered as functional users (Course catalog system and Billing system).

Then there is a space where it is needed to insert any use case.

Finally, we ask that the data groups used, and their movements be explicitly included according to the COSMIC standard, count the times the data groups are moved in the functional process described above, and place them in a table.

Below is the prompt that was used to conduct the tests:

> *Describe the given case use and explicitly include the data groups used and their movements according to the COSMIC standard, the data movements move a group of data and can be read from the database, writing to the database, input from a functional user and output to a functional user, functional users are everything with which the system interacts (e.g. people who use the system, other systems with which it communicates, different systems from which it receives data), data groups describe an object of interest that can be a real world object or a conceptual object (e.g. user, payroll, catalogs, teachers, students, courses, workers).*
>
> *The Course Catalog system and the Billing System are functional users, so to interact with them, there must be exit and entry movements with these.*
>
> *Given the following use case:*
>
> *[**Insert a Functional Process description from C-reg (Table 1)**]*
>
> *Explicitly include the data groups used and their movements according to the COSMIC standard, the movements move a group of data and can be read from the database, written to the database, entry from a functional user, and output to a functional user (e. g. people who use the system, other systems with which it communicates), data groups describe an object of interest that can be a real-world object or a conceptual object (e. g. user, payroll).*
>
> *Additionally, count the times the data groups are moved in the functional process described above and place them in a table with the form: data group, movement, value.*

3.2 Prompts execution

Once the execution of distinct prompts was developed, the results were collected and shown in Table 2.

The functional process ID is shown in the first column, and the name of the selected functional process is shown in the second. The functional size derived from the COSMIC technique, as per Table 1, is shown in column three. The functional size obtained from the first prompt execution utilizing the developed prompt and the Magnitude of Relative Error (MRE) for that first prompt execution is displayed in column six.

Table 2. Comparison of the results of applying the COSMIC method to the C-Reg [34] requirements against the results obtained by chatGPT in two different executions.

ID	***Functional Process***	**Measured CFP**	**Measured CFP (ChatGPT)**	**Diff**	**MRE**
1	Add teacher details (Prompt1)	4	7	3	75.0%
1	Add teacher details (Promp2)	4	10	6	150%
5	Consult the Course Offerings (Teacher) (Prompt1)	7	6	1	14.2%
5	Consult the Course Offerings (Teacher) (Prompt2)	7	10	3	42.8%
15	Modify student schedule (Prompt1)	8	10	2	25.0%
15	Modify student schedule (Promp2)	8	6	2	25.0%

Using the information in Table 2, Table 3 contains the quality criteria for estimation the robustness of the model, which are Mean Magnitude of Relative Error (MMRE), MRE Standard Deviation (SDRMS), and the Prediction level at 10% (Pred 10%) was compiled.

Table 3. Quality criteria for estimation the robustness of the model.

MMRE:	0.553
RMSE:	3.240
SDRMS:	0.510
Pred(10%):	0

Based on the quality criteria, it can be mentioned that there is an average relative error of 5.53%, with a standard deviation of 0.510, and all the measures by ChatGPT are not within the 10% prediction level.

The results show that the size measured using ChatGPT 4o has a difference greater than 10% from the real size measured with the COSMIC method in every FP.

Since the COSMIC method is a standard, the goal of a functional size measure is to be reproducible and audited, any difference could put at risk the project success since there will be a difference in estimating the necessary resources.

4. Discussion

From Table 2, it was observed that like the findings in the article by Hartenstein et al. [27], ChatGPT exhibits some consistency in the total measurement value, in this experiment (only three functional processes) with variation. However, at the individual level, the functional processes yield different results, while functional process 1 shows a significant percentage variation. This observation suggests that there is no reproducibility based solely on the measurement value.

But after carefully examining ChatGPT's responses, we can see that – even in cases where the definition of the text has been given – it is inconsistent to identify data groups, data movements, and functional users straight from the text – elements that software measurers are familiar with. Refer to Table 4.

Table 4. Two ChatGPT functional size responses for the Add teacher details functional process.

Response 1			*Response 2*		
Data Group	*Movement*	*Value*	*Data Group*	*Movement*	*Value*
Command	*Entry*	*1*	*Command*	*Entry*	*1*
Form Template	*Exit*	*1*	*Form Template*	*Exit*	*1*
Teacher Data	*Entry*	*1*	*Teacher Data*	*Entry*	*1*
Teacher Data	*Read*	*1*	*Teacher Data*	*Read*	*1*
Error Messages	*Exit*	*2*	*Error Messages*	*Exit*	*2*
Teacher Data	*Write*	*1*	*Teacher Data*	*Write*	*1*
			Work Area Data	*Write*	*1*
			Course Catalog Data	*Exit*	*1*
			Billing Data	*Exit*	*1*
Total 7 CFP			*Total 10 CFP*		

One of the primary reasons for using a standard metric is to enable auditability of results. However, in this case, even though the results could be analyzed for accuracy (quality criteria like MMRE, STDEV, etc.), they cannot be audited due to the varying elements used to derive the size. Therefore, it is not possible to consider these results as measurements. At best, they could be considered an approximation approach with some considerations.

As an approximation approach, it has not been studied as extensively as other methods. The results do not provide a clear route for gathering or improvement; they seem more like guesses or luck.

From the observations made in this experiment, it becomes apparent that any approximation approach based on text may encounter similar challenges. It is easy to understand this because LLM models like ChatGPT can perform a form of reasoning based on statistical and contextual patterns learned during training. This implies that there should be identifiable patterns or repeated elements in a text that the model can recognize and utilize. However, this is an open question in the software requirement research field, predating the existence of LLM or natural language processing (NLP) technology.

Additionally, numerous subjective elements such as different local expressions, language variations, communication styles, abstractions, etc., could make this a more difficult task. These factors contribute to the complexity of accurately interpreting and analyzing text-based data, making it challenging to develop robust approximation approaches in software requirements. Therefore, addressing these challenges will require interdisciplinary efforts and innovative solutions to advance the state of the art in software measurement and estimation.

5. Conclusions

In this study, we proposed an experiment to determine whether ChatGPT could perform COSMIC measurements. However, we discovered that ChatGPT could not reliably produce accurate findings at the detailed level. The primary shortcomings found in ChatGPT include its incapacity to accurately extract data movements, data groups, and functional users from the text.

From this experiment, we observe that the results produced by the ChatGPT model are not consistent (reproducibly), leading to different results in different executions. This inconsistency generates erroneous measurements and incorrect information, a principal element for metrology, which could put at risk the success of the project.

We can observe that ChatGPT does not adhere to the COSMIC methodology, so the resulting measurements, although correct in some cases, are merely coincidental. If the measurement were audited, it would likely not pass the COSMIC method application, which is a significant issue in software contracting.

Based on the experiment, it is challenging to conceive that LLM models could accurately measure software using a FSMM because they operate on patterns and structured data. In contrast, FURs are often described in free text and depend on the individual writer, leading to variations in language, communication styles, and abstractions.

Indeed, our challenge extends beyond the capabilities of current technology like ChatGPT, it includes the inherent variability and subjectivity of human language. Despite the significant advancements in natural language processing, there is no replicable or consistent way to interpret Functional Unit Requirements (FURs) due to the lack of standardized descriptions.

The proposal by Gérançon et al. [35] offers a potential approach to address this issue. However, current tools, even the most advanced ones like ChatGPT, fall short in meeting the essential requirements for measurement using the COSMIC method directly from the FURs text descriptions: accuracy and reproducibility.

Therefore, it's crucial to recognize that the limitations and issues faced in using AI for estimation tasks extend to approximating functional size from text. Addressing these challenges requires advancements in AI technology, as well as a deeper understanding of the complexities of human language and the specific requirements of the software engineering domain.

6. Limitations

Since OpenAI's ChatGPT model, is not open source, and they can change the way the model responds to mitigate risky results according to their policy [29], the quality of the results in this proposal has been changing from the beginning of this article until the time of its publication (a reproducibility problem).

The steps to perform a functional size measurement using the COSMIC method require knowing the context of the system to be measured, such as the attributes of an object of interest. Thus, the way an LLM will group the attributes of an object of interest could be different from those identified in the measurement process.

How user requirements are obtained can vary, so it is a challenge to create a prompt that could cover everything.

7. Future work

Replicating the experiment with all the functional processes from the case study [34], along with additional real software applications, would provide valuable insights into the performance and limitations of ChatGPT in measuring functional size from text.

Using different or improved versions of LLM, such as ChatGPT 4, 4+, 4o, LLAMA 3.1, Gemini, and Bart, could offer further insights into the model's consistency and performance. To get consistent results under the same prompt in any model would be crucial in determining its reliability.

Finally, exploring the development of a system that leverages ChatGPT for specific parts of the measurement methodology, such as transforming user stories into functional user actions or wrapping attributes in objects of interest, could lead to creating a more accurate measurement model. A mixed approach, like the Retrieval-Augmented Generation (RAG) approach, could combine the strengths of ChatGPT's language processing capabilities with other techniques or models to enhance accuracy and reliability in functional size measurement.

References

[1]. ABRAN, Alain. Software metrics and software metrology. John Wiley & Sons, 2010. https://doi.org/10.1002/9780470606834.ch2.

[2]. Silhavy, R., Prokopova, Z. & Silhavy, P. Algorithmic optimization method for effort estimation. Program Comput Soft 42, 161–166 (2016). https://doi.org/10.1134/S0361768816030087.

[3]. Durán, M., Juárez-Ramírez, R., Jiménez, S. et al. User Story Estimation Based on the Complexity Decomposition Using Bayesian Networks. Program Comput Soft 46, 569–583 (2020). https://doi.org/10.1134/S0361768820080095.

[4]. O. Fedotova, L. Teixeira, A.H. Alvelos, Software effort estimation with multiple linear regression: Review and practical application, J. Inf. Sci. Eng. 29 (2013) 925–945.

[5]. T.K. Lee, K.T. Wei, A.A.A. Ghani, Systematic literature review on effort estimation for Open Sources (OSS) web application development, in: FTC 2016 - Proc. Futur. Technol. Conf., IEEE, San Francisco, California, USA, 2016: pp. 1158–1167. https://doi.org/10.1109/FTC.2016.7821748.

[6]. P. Sharma, J. Singh, Systematic literature review on software effort estimation using machine learning approaches, in: Proc. - 2017 Int. Conf. Next Gener. Comput. Inf. Syst. ICNGCIS 2017, IEEE, Jammu, India, 2018: pp. 54–57. https://doi.org/10.1109/ICNGCIS.2017.33.

[7]. C.E. Carbonera, K. Farias, V. Bischoff, Software development effort estimation: A systematic mapping study, IET Res. Journals. 14 (2020) 1–14. https://doi.org/10.1049/iet-sen.2018.5334.

[8]. E. Ungan, C. Hammond, A. Abran, Automated COSMIC Measurement and Requirement Quality Improvement Through ScopeMaster ® Tool, in: A.C. Murat Salmanoglu (Ed.), Proc. Acad. Pap. IWSM Mensura 2018 "COSMIC Funct. Points - Fundam. Softw. Effort Estim. Held Conjunction with China Softw. Cost Meas. Conf. (CSCM 2018), CEUR Workshop Proceedings (CEURWS.org), Beijing, China, 2018: pp. 1–13. doi: ISSN:1613-0073.

[9]. P. L. Braga, A. L. I. Oliveira and S. R. L. Meira, "Software Effort Estimation using Machine Learning Techniques with Robust Confidence Intervals," in 7th International Conference on Hybrid Intelligent Systems, Kaiserslautern, Germany, 2007.

[10]. Yaozhi Zhang, Nina Katrine Prebensen, Co-creating with ChatGPT for tourism marketing materials, Annals of Tourism Research Empirical Insights, Volume 5, Issue 1, 2024, 100124, ISSN 2666-9579, https://doi.org/10.1016/j.annale.2024.100124.

[11]. Altmäe, Signe Sola-Leyva, Alberto Salumets, Andres, Artificial intelligence in scientific writing: a friend or a foe?, Volume 47, Issue 1, 2023, ISSN 1472-6483 https://doi.org/10.1016/j.rbmo.2023.04.009.

[12]. Zuckerman, M., Flood, R., Tan, R. J. B., Kelp, N., Ecker, D. J., Menke, J., & Lockspeiser, T. (2023). ChatGPT for assessment writing. Medical Teacher, 45(11), 1224–1227. https://doi.org/10.1080/0142159X.2023.2249239.

[13]. T. Putjorn and P. Putjorn, "Augmented Imagination: Exploring Generative AI from the Perspectives of Young Learners," 2023 15th International Conference on Information Technology and Electrical Engineering (ICITEE), Chiang Mai, Thailand, 2023, pp. 353-358, doi: 10.1109/ICITEE59582.2023.10317680.

[14]. S. Bengesi et al., Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformers. arXiv preprint arXiv:2311.10242 (2023).

[15]. McKinsey & Company, What is ChatGPT, DALL-E, and generative AI? | McKinsey. McKinsey & Company (2023).

[16]. OpenAI and Josh Achiam and Steven Adler and Sandhini Agarwal, GPT-4 Technical Report, 2024. arXiv:2303.08774.

[17]. M. Jørgensen, M. Shepperd, A systematic review of software development cost estimation studies, IEEE Trans. Softw. Eng. 33 (2007) 33–53. https://doi.org/10.1109/TSE.2007.256943.

[18]. S. Bilgaiyan, S. Sagnika, S. Mishra, M. Das, A systematic review on software cost estimation in Agile Software Development, J. Eng. Sci. Technol. Rev. 10 (2017) 51–64. https://doi.org/10.25103/jestr.104.08.

[19]. N. Kinoshita, A. Monden, M. Tshunoda and Z. Yucel, "Predictability classification for software effort estimation," in Proceedings - 2018 IEEE/ACIS 3rd International Conference on Big Data, Cloud Computing, Data Science and Engineering, BCD 2018, Yonago, Japan, 2018.

[20]. R. Britto, V. Freitas, E. Mendes, M. Usman, Effort estimation in global software development: A systematic literature review, Proc. - 2014 IEEE 9th Int. Conf. Glob. Softw. Eng. ICGSE 2014. (2014) 135–144. https://doi.org/10.1109/ICGSE.2014.11.

[21]. F. Valdés-Souto, Validation of supplier estimates using cosmic method, CEURInternational Work. Softw. Meas. Int. Conf. Softw. Process Prod. Meas. (IWSM Mensura 2019). 2476 (2019) 15–30.

[22]. M. Shin and A. L. Goel, "Empirical Data Modeling in Software Engineering Using Radial Basis Functions," IEEE Transactions on Software Engineering, vol. 26, no. 6, pp. 567-576, 2000.
[23]. M. Linda and M. C. B. Laird, Software Measurement and Estimation: A Practical Approach, New York, N.Y., USA: Jonh Wiley & Sons, 2006.
[24]. S. Koch and J. Mitlöhner, "Software project effort estimation with voting rules," Decision Support Systems, vol. 46, no. 4, pp. 895-901, 2009.
[25]. De Lucia, E. Pompella and S. Stefanucci, "Assessing effort estimation models for corrective maintenance through empirical studies," Information and Software Technology, vol. 47, no. 1, pp. 3-15, 2005.
[26]. J. Hill, L. C. Thomas and D. E. Allen, «Experts' estimates of task durations in software development projects», International Journal of Project Management, vol. 18, nº 1, pp. 13-21, 2000.
[27]. Hartenstein, S., Johnson, S.L., Schmietendorf, A., ¨Towards a fast cost estimation Supported by large language models¨ (2024). URL: https://cosmic-sizing.org/publications/fast-cost-estimation-by-chatgpt/
[28]. The COSMIC Functional Size Measurement Method: Measurement Manual (2021), v. 5.0 ed., URL https://cosmic-sizing.org/measurement-manual/
[29]. OpenAI and Josh Achiam and Steven Adler and Sandhini Agarwal, GPT-4 System Card, 2024. arXiv:2303.08774
[30]. F. Vogelezang and H. v. Heeringen, Benchmarking: Comparing Apples to Apples (Apress, Berkeley, CA, 2019), pp. 205–217, ISBN 978-1-4842-4221-6.
[31]. Vogelezang, COSMIC Group, ¨Early Software Sizing with COSMIC, Practitioners¨ (2020), v.4.0.2, URL: https://cosmic-sizing.org/publications/early-software-sizing-with-cosmic-practitioners-guide/
[32]. Vogelezang, COSMIC Group, ¨Early Software Sizing with COSMIC: Experts Guide¨ (2020), v.4.0.2, URL: https://cosmic-sizing.org/publications/early-software-sizing-with-cosmic-experts-guide/
[33]. Sánchez Alonso, S., Sicilia Urban, M. Á., & Rodríguez García, D. (2011). Ingeniería del software : un enfoque desde la guía SWEBOK (1a ed., 1a reimp.). Garceta.
[34]. Symons, C.R., et al, Course Registration ('C-REG') System Case Study, v2.0.1 2018. https://cosmic-sizing.org/publications/course-registration-c-reg-system-case-study-v2-0-1/
[35]. Bruel Gérançon, Sylvie Trudel, Roger Kkambou, Serge Robert, Software Functional Sizing Automation from Requirements Written as Triplets, ICSEA 2021: The Sixteenth International Conference on Software Engineering Advances, 2021.

Информация об авторах / Information about authors

Франсиско ВАЛЬДЕС-СОУТО имеет степень PhD в области инженерии программного обеспечения по специальности "Измерение и оценка программного обеспечения" в Высшей технологической школе (ETS) в Канаде, две магистерские степени в Мексике и Франции. Президент COSMIC. Доцент факультета наук Национального автономного университета Мексики. Основатель Мексиканской ассоциации метрик программного обеспечения (AMMS). Более 25 лет опыта в разработке критически важного программного обеспечения. К настоящему времени опубликовал более 50 научных работ, включая статьи в индексированных журналах, трудах научных конференций, книгах и главах книг. Является главным промоутером проекта изучения формальных метрик программного обеспечения в Мексике, продвигая COSMIC (ISO/IEC 19761) в качестве национального стандарта. Член Национальной системы исследователей (SNI). Область научных интересов: измерение и оценка программного обеспечения, применяемого для управления проектами программного обеспечения, управление тематикой, производительностью и экономикой разработок программного обеспечения.

Francisco VALDÉS-SOUTO had a PhD in Software Engineering with a specialty in Software Measurement and Estimation at the École de Technologie Supérieure (ETS) in Canada, two master's degrees in Mexico and France. President of COSMIC. Associate Professor of the Faculty of Sciences of the National Autonomous University of Mexico (UNAM). Founder of the Mexican Association of Software Metrics (AMMS). More than 25 years of experience in critical software development. He currently has more than 50 publications including articles in Indexed Journals, Proceedings, books and book chapters. He is the main promoter of the topic of formal software metrics in Mexico, promoting COSMIC (ISO/IEC 19761) as a National Standard. Member of the National System of

Researchers (SNI). Research interests: software measurement and estimation applied to software project management, scope management, productivity and economics in software projects.

Даниэль ТОРРЕС-РОБЛЕДО – магистрант Исследовательского института в области прикладной математики и систем, имеет ученую степень по программированию от научного факультета Национального автономного университета Мексики.

Daniel TORRES-ROBLEDO – Master student at Research Institute in Applied Mathematics and Systems, degree in Computer Science from Science Faculty of the UNAM.