EAI Endorsed Transactions
on Scalable Information Systems Research Article **EAI.EU**

Software quality models: Exploratory review

Lloy Pinedo[1*], Miguel Valles-Coral[1], Jorge Raul Navarro-Cabrera[1], Richard Injante[1], Ángel Cárdenas-García[1], Fernando Ruiz-Saavedra[1], Cecilia García-Rivas-Plata[2] and César Augusto Flores-Tananta[3]

[1]Universidad Nacional de San Martin, Tarapoto, Peru
[2]Universidad Nacional Jose Maria Arguedas, Andahuaylas, Peru
[3]Universidad Cesar Vallejo, Tarapoto, Peru

Abstract

INTRODUCTION: The emerging industrialization of software promotes the continuous development of new evaluation models that adapt to the product's characteristics and the organizations' needs.
OBJECTIVES: This article aims to identify the software quality models proposed between 2016 and 2020.
METHODS: We conducted an exploratory systematic review in the SciELO, IEEE Xplore, and Scopus databases, resulting in 14 research papers proposing new models for assessing software quality.
RESULTS: 79% of the research papers were extracted from IEEE Xplore, and 50% of them were authored by individuals affiliated with European institutions. We presented various software quality models focused on assessing both product quality and usability for specific purposes. Additionally, we reported that the proposed models are based on common software quality metrics standards such as CMMI, ISO/IEC 9126, and others.
CONCLUSION: The review serves to update the state of the art regarding the software quality models proposed in recent years, which will be valuable for researchers and software developers when seeking evaluation models that align with their specific needs.

Keywords: product quality, quality of use, quality standards, software development, systematic review

Received on 12 March 2023, accepted on 15 September 2023, published on 26 September 2023

doi: 10.4108/eetsis.3982

1. Introduction

Callejas-Cuervo et al. [1] point out that the term "Software Quality" refers to the performance that a computer system must uphold throughout its lifecycle to ensure the efficiency of its functionalities. In a similar vein, Toro et al. [2] mention that software quality determines the success of companies that are currently striving for innovation and continuous improvement. Hence, the significance of the attention given to this subject.

This is why assessing the quality of software products has become a strategic element for organizations due to the impact it has on the competitiveness of developing companies. As a result, various models have been proposed to accommodate the size of companies and provide attributes for the evaluation and selection of software based on compliance with quality metrics [3].

According to Akbar et al. [4], developing high-quality software not only satisfies customer expectations but also benefits the organization by meeting production goals within the planned timeframe and cost estimates. To achieve this, as suggested by Quiroz-Martínez et al. [5], it is essential to prioritize the identification of requirements that determine the functionalities to be included in the product, considering key factors to meet software quality criteria.

On the other hand, certification processes issued by institutions based on quality standards for software products ensure efficiency, effectiveness, and add value to the products. They also instill confidence in customers and provide predictability in work. Therefore, their

*Corresponding author. Email: lpinedo@unsm.edu.pe

consideration is valuable for companies engaged in software development that aim to showcase the quality of their production through certifications with international impact [6].

According to Huda et al. [7], inherently large and complex software systems with numerous correlated metrics developed under different components make the construction of new software quality evaluation models very complicated. As a result, as indicated by Martinez-Fernandez et al. [8], the evaluation and improvement of software quality are primary objectives for the community of systems engineers, computer scientists, and computing professionals, leading to new proposals and the formulation of quality standards and models.

It is worth noting that globally, in the field of software production, multiple standards, methodologies, and models have been developed to improve their management and development, aiming for high-quality products [2,9]. However, Saini et al. [10] point out that these quality assurance approaches or traditional metrics have limitations or shortcomings because some of them solely focus on evaluating source code and cannot be used for quality prediction or fault detection.

Several systematic review studies have been conducted on software quality models [1,10–16] where researchers compare, classify, evaluate, and determine models that are in higher demand or provide greater reliability and effectiveness in their applications for assessing software quality. However, there is still a need to continue exploring new software quality models, as they are frequently emerging in scientific literature.

For this reason, this article sets out to identify software quality models proposed between the years 2016 and 2020. The goal is to compile and encompass models aimed at software quality evaluation (QS). To achieve this, we adapted an analysis matrix for models based on the studies of Villalta et al. [11] and Yan et al. [14], classifying them by years of development or launch, incorporation of standards, and the authors who designed them. The primary purpose of this review is to update and enrich the state of the art in terms of research conducted on QS models, providing researchers and developers with the ease of finding current models.

2. Methodology

In recent decades, the software industry has undergone a revolution. Initially, determining the quality of a product was solely based on its functionality. However, as the years have passed and the complexity of building robust systems has increased, new characteristics and sub-characteristics have been incorporated to assess quality in a broader and more rigorous manner. As a result, there has been a growing development of models that can adapt to the needs of a project [12]. In light of this perspective, we formulate the research question: ¿What software quality models have been proposed between the years 2016 and 2020?

To answer this question, we applied the methodology of Exploratory Systematic Review (ESR), which is a valuable tool for bibliographic reviews due to its systematic, transparent, and applicable nature in various scientific fields [17]. Taking into account the phases of ESR, we proceeded to conduct a literature search in electronic databases during the month of March 2021. We used SciELO, IEEE Xplore and Scopus as resources because they have a broad regional and international scope and also facilitate advanced searching through filters.

We used the terms "model", "quality" and "software" which we concatenated using the boolean operator AND, and we searched in the fields of title, abstract, and keywords. For SciELO, we conducted the search in both spanish and english.

Regarding the criteria for document selection, we included scientific articles, conference papers, and book chapters in both Spanish and English published between 2016 and 2020 that focused on new proposals for software quality evaluation models. We excluded review articles, opinion pieces, short communications, duplicates, and items classified as gray literature (technical reports, theses, monographs, etc.).

The results of the initial search without filters yielded 325, 17,935, and 65,430 documents according to SciELO, IEEE Xplore, and Scopus, respectively. After applying filters for the year range, document type, and language, we obtained 105, 4,051, and 16,198 documents. Subsequently, we reviewed the titles and abstracts of each article, excluding those that did not propose new models for evaluating software quality. This process was exhaustive to eliminate duplicates and avoid any bias. It is worth noting that a significant portion of the filtered literature focused on the evaluation of software product quality rather than on proposals or model development.

At the conclusion of the review process, you've gathered 14 papers focused on the development of software quality measurement models. Storing them in Mendeley for subsequent in-depth review is an excellent approach, as it allows you to organize the metadata such as title, authors, journal, year, volume, number, pages, URL, or DOI. Properly managing your bibliographic sources will be crucial as you progress in your research [18].

3. Results and discussion

3.1. Publications by database

After extracting the information, we identified that a large percentage of studies proposing the development of new software quality (QS) models are available on IEEE Xplore (79%), followed by SciELO (14%) and Scopus (7%) (Figure 1). This does not mean that scientific literature is limited to a single database but rather reflects that research is disseminated across different sources and may even be included in one another.

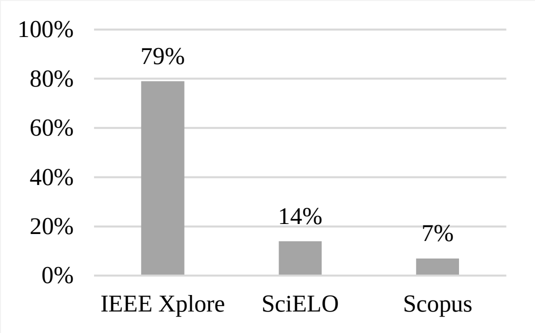

Figure 1. Publications by database

3.2. Regional distribution

We consider it important to identify the distribution of proposed software quality models by region. To do so, we took into account the institutional affiliation of the authors, obtaining a higher proportion in the European region (50%), followed by Asia (29%), and Latin America (21%) (Figure 2). These results reveal that a significant portion of the software quality models proposed between 2016 and 2020 were formulated by European researchers, although there is also scientific production evident from authors in other regions.

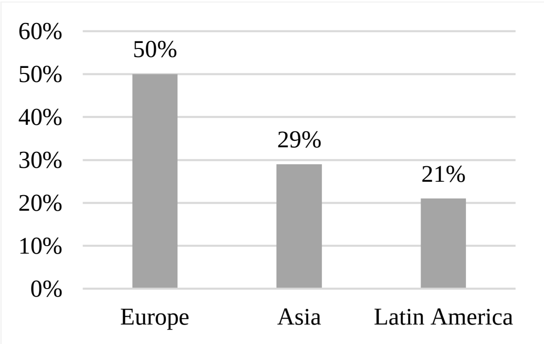

Figure 2. Publications by region

3.3. Software quality models (2016-2020)

To characterize the software quality models published between 2016 and 2020, we designed a data collection matrix from which we extracted information regarding the authors, model name, objective, standards used, and impact or results generated. These details are presented in Table 1 (next page).

Among the identified software quality models, we highlight the use of widely adopted industry standards, which served as a foundation or were expanded upon for new software quality evaluation proposals. Some of these standards include.

ISO/IEC 9126

ISO/IEC 9126 is an international standard developed for assessing and measuring the quality of software products. This standard focuses on six primary quality characteristics: functionality, reliability, usability, efficiency, maintainability, and portability. Each of these characteristics is further divided into subcharacteristics that provide specific criteria for evaluating quality in various aspects of the software [19].

ISO/IEC 9126 is widely used in the software industry to define quality requirements and establish metrics for measuring compliance with these requirements. For example, in the realm of functionality, it assesses the software's ability to meet user needs, while in efficiency, it measures performance and resource utilization.

ISO/IEC 25010

ISO/IEC 25010 is a revision and expansion of the ISO/IEC 9126 standard. This standard introduces a more comprehensive and up-to-date model for software quality. It divides software quality into eight quality characteristics, which are further broken down into subcharacteristics and quality attributes. In addition to the characteristics mentioned in ISO/IEC 9126, ISO/IEC 25010 includes aspects such as security, compatibility, and interoperability [20].

This more detailed model provides a more comprehensive view of software quality and helps organizations better understand user needs. It facilitates informed decision-making to enhance software quality, as it allows for a more precise assessment of the strengths and weaknesses of the product.

CMMI

CMMI (Capability Maturity Model Integration) is a maturity and capability model used to assess and enhance an organization's ability to consistently develop high-quality software. It focuses on key areas such as project management, software engineering, and process management [21]. CMMI provides a set of best practices and a gradual path for improving the quality and efficiency of software development processes [22].

Organizations that adopt CMMI can assess their maturity in terms of processes and receive guidance on how to improve. This helps them standardize and optimize their software development practices, which, in turn, leads to higher product quality and greater operational efficiency.

Table 1. Matrix of software quality models (2016-2020)

Autores	Denominación	Objetivo	Estándar	Impacto
Espejo-Chavarría et al. [23]	High-Level Quality Assurance Model	Develop a quality assurance model with a focus on human talent	CMMI	Quality cost reduction from 19.85% to 7.41%.
Gitto et al. [24]	First-tier Construction Quality Assurance Model	Establish a complex Quality Assurance (QA) model that integrates the client's perspective	ISO/IEC 9126	According to a survey, it provides reliability, security, and other aspects (38%)
González-Reyes et al. [12]	External Quality Assurance Evaluation Model	Propose an external quality model for evaluating software in early-stage organizations	ISO / IEC 25010	Applicable to any type of organization initiating their work in quality or the quality area
Akbar et al. [4]	AZ Model	Develop the Software Development Life Cycle (SDLC) model, known as the "AZ Model"	Not specified	The model combines the most important features of traditional methodologies, yielding optimal results in Quality Assurance (QA)
Kumar et al. [25]	Innovative Software Defect Prediction Model utilizing Probabilistic ABC	Design and implement a prediction model using probabilistic classification based on ABC	Not specified	The proposed model has a high detection rate compared to traditional models
Yuanxu et al. [26]	HMM-FNN	Introduce a software analysis model based on fuzzy data and Feedforward Neural Network (FNN), as well as Hidden Markov Model (HMM)	Not specified	The model not only increased prediction accuracy but also improved learning efficiency by modifying the error function algorithm during the learning phase
Mendonca et al. [27]	ELECTRE TRI	Present a multi-criteria analysis model	ISO 9126	It is observed as a highly useful model for analyzing service quality in terms of user satisfaction but still underutilized for analyzing software quality
He et al. [28]	Method of Evaluation Based on Characteristic Parameters	Demonstrate the reliability assessment of a software component using a characteristic-parameter-based evaluation method	Not specified	Preliminary results show that this method can be effectively applied in evaluating the reliability of software components
Utsunomiya et al. [29]	Not specified	Group classes based on influencing factors such as the number of roles, type of design pattern, and static associations between design patterns and anti-patterns	Not specified	Empirically, the failure density distributions among class groups influence the design pattern for classifying the classes
Legowo et al. [30]	QAS (Quality Assurance System)	Develop a quality assurance system using the agile development methodology with Scrum	BAN-PT / ISO 9001:2008	The accuracy of the QAS methodology in Quality Determination (QD) is verified
Cevallos- Lopez et al. [31]	JS (Joint System)	Create a JS (Joint System) model that serves as a specific guide in the domain of educational JS to ensure project quality	CMMI-Dev 1.3	The model extends the CMMI model by incorporating quality criteria and best practices obtained from JS studies
Moyo et al. [32]	Secure-SSDM (Secure Software Development Model)	Propose a software development methodology that promotes quality and security in software products	ISO / IEC 25010	The theoretical contribution is the addition of security practices to the individual software development knowledge base
Yan et al. [33]	SCT Model	Build a comprehensive evaluation system that assesses the scientific nature and accuracy of the proposal	CMMI	The quality of the methodology, using Backpropagation Neural Network (BP), significantly aligns with the comprehensive software evaluation in terms of quality
Ahmad et al. [34]	SOQEMM (Software Quality Engineering and Management Model)	Develop a software subcontracting model (SOQEMM) from the provider's perspective	Not specified	Suppliers can identify, examine, and address quality-related challenges effectively

The matrix provides a panoramic view of research in the field of software quality between 2016 and 2020. The results reveal a diversity of approaches and objectives adopted by researchers, reflecting the breadth and complexity of this field.

One noteworthy aspect is the multiplicity of approaches in the models. These models are designed to address various objectives in software quality according to the specific software product [35]. Some focus on quality assurance, while others are geared towards defect prediction, integrating security into development [36], or assessing educational projects [37]. This array of approaches underscores the need for adaptive and specific strategies to tackle different aspects of software quality.

Furthermore, several of these models are based on widely recognized quality standards such as CMMI, ISO/IEC 9126, and ISO/IEC 25010. This trend of using established standards suggests that there is significant value in following well-established frameworks to enhance software quality [1]. The implementation of these standards provides a solid and globally recognized foundation for optimizing software quality.

In terms of impact, some models have achieved favorable outcomes. For example, Espejo-Chavarría et al.'s high-level quality assurance model [23] reduced quality costs, demonstrating the practical value of implementing an approach based on recognized standards. Similarly, Kumar et al.'s software defect prediction model [25] stands out for its ability to detect defects with a higher success rate than traditional models. These results indicate that applying specific approaches can positively influence software quality and development process efficiency.

However, it is relevant to note that some models do not specify concrete standards, which raises questions about the replicability and applicability of their results in broader contexts. Additionally, the matrix highlights that while many models have focused on technical aspects of software quality, such as functionality and security, there are still opportunities to research approaches that address quality from more holistic perspectives, such as user satisfaction and project management.

This matrix of software quality models emphasizes the importance of ongoing research in this ever-evolving field. Technology advances, and user expectations constantly change, necessitating an agile and adaptable approach to address quality challenges in software development. Each model and approach presented in the matrix contributes to the knowledge in this field and provides valuable insights for future research and improvements in software quality.

3.4. Limitations of the review

As limitations of the present review, we highlight that the search was conducted in only three databases, and we restricted the year range from 2016 to 2020. For future research, we suggest expanding the search to include new software quality models developed between 2021 and 2023. Additionally, it would be beneficial to consider the inclusion of additional databases such as Web of Science and Google Scholar to ensure a comprehensive collection of relevant models.

Furthermore, our review primarily focused on identifying software quality models proposed within a 5-year period, emphasizing their objectives and generated results. In future research, it could be explored whether these models have been adopted and used in the software industry, as well as whether they have undergone evolutions or new versions. This analysis would provide a more comprehensive understanding of the applicability and effectiveness of these models in practical environments.

Conclusions

In this article, we have presented a exploratory systematic reviews that comprehensively addresses the software quality models developed or proposed up to the present. The results reflect the dynamic progress in creating new models that respond to the changing demands of software production projects and have the primary objective of ensuring customer satisfaction.

A noteworthy finding of this review is the prevalence of adopting established standards such as CMMI and ISO regulations as fundamental bases for the development of new proposals in the field of software quality. This highlights the importance of using internationally recognized frameworks to establish solid foundations in software quality management. The ability of these standards to serve as a starting point in creating customized models demonstrates their versatility and relevance in today's software industrialization.

Finally, this article aims to provide access to relevant and useful information for academics, students, and professionals interested in software quality. Quality models play a crucial role in improving software development processes and end-user satisfaction, and this review provides a solid foundation for future research and practical applications in the ever-evolving field of software quality.

References

[1] Callejas-Cuervo M, Alarcón-Aldana AC, Álvarez-Carreño AM. Modelos de calidad del software, un estado del arte. Entramado. 2017; 13(1):236–50. https://doi.org/10.18041/entramado.2017v13n1.25125.

[2] Toro A, Peláez LE. Validación de un modelo para el aseguramiento de la calidad del software en MIPYMES que desarrollan software en el Eje Cafetero. Entre Cienc e Ing. 2018; 12(23):84–92. https://doi.org/10.31908/19098367.3707

[3] Fernandez-Perez Y, Cruz-Corona C, Verdegay-Galdeano JL. A New Model Based on Soft Computing for Evaluation and Selection of Software Products. IEEE Lat Am Trans. 2018; 16(4):1186–92. https://doi.org/10.1109/TLA.2018.8362155

[4] Akbar MA, Sang J, Khan AA, Fazal-E-Amin, Nasrullah, Shafiq M, et al. Improving the Quality of Software Development Process by Introducing a New

Methodology–AZ-Model. IEEE Access. 2018; 6:4811–23. https://doi.org/10.1109/ACCESS.2017.2787981

[5] Quiroz-Martínez MÁ, Jurado-Antón OA, Plua-Moran DH, Leyva-Vazquez MY. Priorización de requisitos para una adecuada calidad de software. Ser Cien Univ Cien Inform. 2020; 13(6):135–45.

[6] Esterkin V, Pons C. Evaluación de calidad en el desarrollo de software dirigido por modelos. Ingeniare Rev Chil Ing. 2017; 25(3):449–63. https://doi.org/10.4067/S0718-33052017000300449

[7] Huda S, Alyahya S, Mohsin Ali M, Ahmad S, Abawajy J, Al-Dossari H, et al. A Framework for Software Defect Prediction and Metric Selection. IEEE Access. 2018; 6:2844–58. https://doi.org/10.1109/ACCESS.2017.2785445

[8] Martinez-Fernandez S, Vollmer AM, Jedlitschka A, Franch X, Lopez L, Ram P, et al. Continuously Assessing and Improving Software Quality With Software Analytics Tools: A Case Study. IEEE Access. 2019; 7:68219–39. https://doi.org/10.1109/ACCESS.2019.2917403

[9] Pancik J, Vemola A, Kledus R, Semela M, Bradac A. Auto recalls and software quality in the automotive sector. EAI Endorsed Trans Scalable Inf Syst. 2018; 5(17):154808. https://doi.org/10.4108/eai.29-5-2018.154808

[10] Saini GL, Panwar D, Kumar S, Singh V. A systematic literature review and comparative study of different software quality models. J Discret Math Sci Cryptogr. 2020; 23(2):585–93. https://doi.org/10.1080/09720529.2020.1747188

[11] Villalta A, Carvallo JP. Modelos de calidad de software: Una revisión sistemática de la literatura. Maskana. 2016; 6:107–17.

[12] González-Reyes A, Hernández-González A, André-Ampuero M. Modelo básico inicial de calidad externa para productos de software. Rev Cuba Ciencias Inform. 2016; 10:94–111.

[13] Morales-Arias JJ, Pardo-Calvache CJ. Revisión sistemática de la integración de modelos de desarrollo de software dirigido por modelos y metodologías ágiles. Inf Técnico. 2016; 80:87–99. https://doi.org/10.23850/22565035.326

[14] Yan M, Xia X, Zhang X, Xu L, Yang D, Li S. Software quality assessment model: a systematic mapping study. Sci China Inf Sci. 2019; 62(2):e191101. https://doi.org/10.1007/s11432-018-9608-3

[15] Adewumi A, Misra S, Omoregbe N, Crawford B, Soto R. A systematic literature review of open source software quality assessment models. Springerplus. 2016; 5(1):e1936. https://doi.org/10.1186/s40064-016-3612-4

[16] Sagar K, Saha A. A systematic review of software usability studies. Int J Inf Technol. 2017. https://doi.org/10.1007/s41870-017-0048-1

[17] Fernández-Sánchez H, King K, Enríquez-Hernández CB. Revisiones Sistemáticas Exploratorias como metodología para la síntesis del conocimiento científico. Enfermería Univ. 2020; 17(1):87–94. https://doi.org/10.22201/eneo.23958421e.2020.1.697

[18] Pinedo-Tuanama Ll, Valles-Coral M. Importancia de los referenciadores bibliográficos en la gestión de la información científica en tesistas universitarios. An Doc. 2021; 24(2). https://doi.org/10.6018/analesdoc.465091

[19] Al-Kilidar H, Cox K, Kitchenham B. The use and usefulness of the ISO/IEC 9126 quality standard. Int. Symp. Empir. Softw. Eng., IEEE; 2005. p. 122–8. https://doi.org/10.1109/ISESE.2005.1541821

[20] Mulyawan MD, Kumara INS, Swamardika IBA, Saputra KO. Kualitas Sistem Informasi Berdasarkan ISO/IEC 25010: Literature Review. Maj Ilm Teknol Elektro. 2021; 20(1):15–28. https://doi.org/10.24843/MITE.2021.v20i01.P02

[21] Selleri-Silva F, Furtado-Soares FS, Lima-Peres A, Monteiro-de-Azevedo I, Vasconcelos APLF, Kenji-Kamei F, et al. Using CMMI together with agile software development: A systematic review. Inf Softw Technol. 2015; 58:20–43. https://doi.org/10.1016/j.infsof.2014.09.012

[22] Iqbal J, Ahmad RB, Nasir MHNM, Niazi M, Shamshirband S, Noor MA. Software SMEs' unofficial readiness for CMMI®-based software process improvement. Softw Qual J. 2016; 24:997–1023. https://doi.org/10.1007/s11219-015-9277-3

[23] Espejo-Chavarría A, Bayona-Oré S, Pastor C. Aseguramiento de la Calidad en el Proceso de Desarrollo de Software utilizando CMMI, TSP y PSP. RISTI - Rev Ibérica Sist e Tecnol Inform. 2016; 20(12):62–77. https://doi.org/10.17013/risti.20.62-77

[24] Gitto JP, Bosch-Mauchand M, Durupt AP, Cherfi Z, Guivarch I. A Methodology for Complex System Quality Model Construction – First level. IFAC-PapersOnLine, vol. 49, Elsevier B.V.; 2016. p. 319–24. https://doi.org/10.1016/j.ifacol.2016.07.624

[25] Kumar PR, Saradhi-Varma GP. A novel probabilistic-ABC based boosting model for software defect detection. Int. Conf. Innov. Information, Embed. Commun. Syst. ICIIECS 2017, IEEE; 2018, p.1–6. https://doi.org/10.1109/ICIIECS.2017.8276059

[26] Yuanxu L, Chaowen C, Peisheng H, Guo W. A software development project quality analysis model based on HMM-FNN. Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS, IEEE; 2018. p.220–4. https://doi.org/10.1109/ICSESS.2017.8342900

[27] Mendonca J, Fernandes Lima AC, Di-Oliveira V, Bittencourt-Reis AC, Borges-Simao-Monteiro S. Classification model of the software quality level according to the user's perception. Int. Conf. Qual. Inf. Commun. Technol. QUATIC 2018, IEEE; 2018. p.297–8. https://doi.org/10.1109/QUATIC.2018.00054

[28] He R, Tang L, Han X, He W, Zhao Z. Software Component Reliability Evaluation Method Based on Characteristic Parameters. 18th Int. Conf. Softw. Qual. Reliab. Secur. Companion, QRS-C 2018, IEEE; 2018. p.235–41. https://doi.org/10.1109/QRS-C.2018.00051

[29] Utsunomiya H, Yamamoto S. Proposal of Software Quality Confirmation Method Using Feature Model. 7th Int. Congr. Adv. Appl. Informatics IIAI-AAI 2018, IEEE; 2018. p.769–74. https://doi.org/10.1109/IIAI-AAI.2018.00159

[30] Legowo MB, Indiarto B, Prayitno D. Agile Software Methodology with Scrum for Developing Quality Assurance System. 2nd Int. Conf. Comput. Informatics Eng. Artif. Intell. Roles Ind. Revolut. 4.0, IC2IE 2019, IEEE; 2019. p.104–9. https://doi.org/10.1109/IC2IE47452.2019.8940831

[31] Cevallos- Lopez AA, Santorum-Gaibor MO. A quality model for the serious games development process. Int. Conf. Inf. Syst. Softw. Technol. ICI2ST 2019, IEEE; 2019. p.1–8. https://doi.org/10.1109/ICI2ST.2019.00008

[32] Moyo S, Mnkandla E. A Novel Lightweight Solo Software Development Methodology with Optimum

Security Practices. IEEE Access. 2020; 8:33735–47. https://doi.org/10.1109/ACCESS.2020.2971000
[33] Yan B, Yao HP, Nakamura M, Li ZF, Wang D. A case study for software quality evaluation based on SCT model with BP neural network. IEEE Access. 2020; 8:56403–14. https://doi.org/10.1109/ACCESS.2020.2981872
[34] Ahmad J, Khan AW. Software Outsourcing Quality Evaluation Management Model (SOQEMM). 3rd Int. Conf. Comput. Math. Eng. Technol. Idea to Innov. Build. Knowl. Econ. iCoMET 2020, IEEE; 2020. p.1–10.https://doi.org/10.1109/iCoMET48670.2020.9073839
[35] Espinosa-Luna BH, Castillo-Oliva J, Montañez-Díaz BA, Mendoza-De-los-Santos A. Implementación de un chatbot basado en modelo de lenguaje de inteligencia artificial para responder preguntas frecuentes de estudiantes universitarios. Rev Cient Sist e Inform. 2023;3(2):e570. https://doi.org/10.51252/rcsi.v3i2.570
[36] Zanabria-Ortega M, Benites-Cazorla MV. Sistema NOTION para la administración de los hogares de los trabajadores del poder judicial de Puno post Covid-19. Rev Cient Sist e Inform. 2023; 3(2):e584. https://doi.org/10.51252/rcsi.v3i2.584
[37] Saldaña-Cerván EJ. Implementación de un aula virtual para mejorar la satisfacción de los estudiantes de secundaria en una institución educativa peruana. Rev Cient Sist e Inform. 2023; 3(1):e474. https://doi.org/10.51252/rcsi.v3i1.474