Revista Cubana de Ciencias Informáticas
Vol. 16, No. 3, Julio-Septiembre, 2022
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 51-70

Tipo de artículo: Artículo original
Temática: Ingeniería y gestión de software
Recibido: 21/06/2022 | Aceptado: 01/07/2022

Framework for evaluating performance efficiency from early stages in software development

Marco de trabajo para evaluar la eficiencia del desempeño desde etapas tempranas en el desarrollo del software

Maidelyn Piñero González[1] https://orcid.org/0000-0002-6534-132X
Aymara Marin Diaz[1] https://orcid.org/0000-0001-5101-7804
Yaimí Trujillo Casañola [1] https://orcid.org/0000-0002-3138-011x
Raidel Paez Llopiz [2] https://orcid.org/0000-0003-4541-9676

[1]University of Computer Science, Cuba. Highway San Antonio Km 2 1/2. {mpinero, amarin, yaimi}@uci.cu
[2]Itopia, Cuba. Calle 1 e/ 56 y 58 Artemisa. raidel.paez88@gmail.com

* Author for correspondence. (mpinero.uci.cu)

ABSTRACT

Nowadays, with the technological revolution, software has become part of everyday activities. The demand for software with quality and speed is a decisive factor for the competition between software development companies or organizations. ISO/IEC 25010:2011 mentions that performance efficiency is related to the performance relative to the amount of resources used under set conditions, based on their temporal behavior, the use of resources and the software's capacity to function. In Cuba, there is evidence of user dissatisfaction with the response times and connectivity of national mobile applications. On the other hand, software development companies consider testing associated with performance efficiency to be important; however, 71% carry it out in the final stages of system development. As a consequence, it is evident that defects associated with performance efficiency persist in the reviews. This research proposes a framework for incorporating the evaluation of performance efficiency from early stages in the development of software, which allows us to know the behavior of this characteristic during the software life cycle and detect defects as early as possible to correct them at the corresponding level and not escalate to higher levels, which makes it possible to increase the quality of the final product. The validation was carried out using the methods and techniques of high international reference.

Keywords: performance efficiency; quality activities; user satisfaction; quality control and quality assurance.

RESUMEN

La eficiencia del desempeño como característica de calidad del producto es un factor importante a tener en cuenta desde inicio del desarrollo del sistema, pues el mal desempeño de esta característica afecta la satisfacción del usuario. Por ello, es significativo conocer el comportamiento de la eficiencia del desempeño en cada etapa y nivel de desarrollo del software a través de diferentes actividades de calidad. En el presente artículo se analizan normas, estándares, modelos, metodologías y criterios de autores reconocidos a nivel internacional con el objetivo identificar las actividades de calidad que se realizan para conocer el comportamiento de la eficiencia del desempeño en el software. Se definen objetivos, precondiciones, frecuencia de uso, productos de trabajo y los resultados esperados de las actividades de calidad propuestas asociadas a la eficiencia del desempeño. Se considera la ejecución de estas actividades como parte del proceso de control y aseguramiento de la calidad en las instituciones y empresas desarrolladoras de software. Es necesario la incorporación de las actividades desde inicio del desarrollo y realizarlas de forma transversal a las actividades de análisis, diseño e implementación del software. Para valorar la contribución de la propuesta dada, se utiliza el método Delphi y criterios de expertos, siendo exitosamente aceptada y aprobada por los mismos.

Palabras clave: eficiencia del desempeño; actividades de calidad; satisfacción del usuario; control; aseguramiento.

Introduction

The increase in the development and implementation of software solutions has covered various areas and social sectors. In this context, the growing demand and interest on the part of the market and users for high quality software products is an essential element for their use and commercialization. (Estrada, García et al., 2011, Zaraket, 2016, Mina and Barzola, 2017, Paz, Gómez et al., 2017, Benjumea, 2019, Diaz, Casañola et al., 2020) Nowadays, systems are increasingly complex, which makes it very difficult to coordinate that all the services involved are available and stable because each of these subsystems is managed. The quality of performance efficiency is significantly reduced, incurring security and increased failures that may occur (FOCUS, 2013).

Not knowing how software behaves in a stressful situation, at the limit of resource usage and the affected response time, leads to dissatisfaction and dissatisfaction among end-users and they will reject software that is slow (Globe, 2017). Performance efficiency is a quality model characteristic defined by the International Organization for Standardization (ISO) in ISO/IEC 25010:2011 that relates to the performance of a device as a function of its temporal behavior, resource usage and the software's capacity or performance limits (ISO/IEC, 2011, Muñoz, 2018, Kaur, Grover et al., 2019).

The International Software Testing Qualification Board (ISTQB) in its report from 2013 to 2014 identified performance efficiency testing as one of the tests of the future (ISTQB, 2014). For 2017 and 2018, it showed that the trends for the software testing profession will be test automation, and that, of the most important tests for the organization, performance efficiency tests account for 60.7%. (ISTQB®, 2018, Diaz, Casañola et al., 2020, González, Diaz et al., 2021) In the report for 2019 and 2020, interest in performance testing certification increases with 40%. (International Software Testing Qualifications Board, 2020, International Software Testing Qualifications Board, 2020, González, Diaz et al., 2021)

In the report for 2019 and 2020, interest in performance testing increases with 40%. Moreover, MICRO FOCUS1 together with Sogeti2 and Capgemini in the 2019 and 2020 period edition. highlighted that organizations consider assigning responsibilities for technical quality in aspects of performance efficiency testing. On the basis that performance efficiency testing is the most used with 63%, performance efficiency engineering stands out among the skills in need of change with 48%, focusing on the demand for quality in speed and giving it the responsibility to ensure end-user satisfaction, this being the main objective within the software testing and quality assurance strategy.(Micro Focus, 2020, González, Diaz et al., 2021)

As part of the country's computerization process, Cuba has adopted standards to evaluate the quality of the product in companies, institutions or organizations. To this end, it has created the Quality Model for the Development of Computer Applications (MCDAI), which among the quality characteristics taken into account to evaluate the product is performance efficiency (Granma, 2017). In order to find out the trend in the use and evaluation of this characteristic, a diagnosis was carried out.

A survey of 28 users between 18 and 40 years of age who use national mobile applications to manage transport, food and cleaning, showed that response time and difficulty in connectivity generate dissatisfaction, with 100% of respondents mentioning this. Another survey of 37 specialists with more than 4 years of experience from 7 different entities that develop software in the country, showed that for 100% of them the use of performance efficiency tests is important for the acceptance of the product by users. However, 54% do not include performance efficiency testing in their project. Of the 17 respondents that include testing for software performance efficiency (46%), it was confirmed that 71% of them take it into account in the final stages of the product (37% do it at the end of development and another 34% in the closure phase); only 29% take it into account during the software development phase.

A documentary review of the quarterly and annual trend reports for 2017, 2018 and 2019, which are kept by the UCI Software Quality Directorate in order to know the progress of the products over a period of time, showed that in more than 50% of the applications reviewed annually, defects associated with HTTP requests, the response time of the software and the user concurrency. It became evident that on several occasions technical opinions have been carried out because in the fourth iteration of the systems defects associated with performance efficiency remain outstanding. It is shown that, when performance testing is carried out in advanced stages of the software, the effort to correct the error is greater and this leads to an impact on the delivery times planned with the client and even to discarded software. When reviewing the main causes of this situation, the development teams claim that in order to resolve these non-conformities, significant changes in the software are necessary, among which they mention: re-engineering, architecture redefinition, code optimization and even the lack of knowledge of the solution to the defects, which implies effort, cost and time.

Based on the aforementioned problems, the problem to be solved is defined as: How to detect defects associated with performance efficiency in early stages of software development and increase the quality of the final product? In order to solve the scientific problem posed above, the general objective is defined as follows: To develop a framework to evaluate the efficiency of performance from early stages in software development that includes internationally standardized practices for detecting defects and increasing the quality of the final product.

Computational Methods or Methodology

Several scientific methods were used to obtain information for the research. Among the theoretical methods, the following were used: analytical - synthetic for the analysis of the common elements provided by the literature on performance efficiency to identify the quality activities that are proposed and are relevant to know the behavior associated with this characteristic; the inductive - deductive used in the diagnosis of the trend of the use of quality activities associated with performance efficiency at national and international level; and the historical-logical analysis for the study of the historical trajectory on the performance efficiency characteristic that have been reflected by different authors, models, norms and standards, so as to provide a knowledge base from the historical to incorporate the use of activities that guarantee a correct performance efficiency in the software. Within the empirical methods, formal and informal interviews and the survey were used to obtain data and information to argue the problematic situation and selection of the quality activities associated to the efficiency of the performance according to experiences in the evaluation of the characteristic in the software development cycle.

Results and discussion

A framework provides a base structure for organizing the components of a process(CMMI® 2010, Sánchez, 2015, Diaz, Casañola et al., 2018). The use of a framework in the software development process has been a good practice in recent years, as it establishes a definite conceptual and technological support structure, usually with concrete artefacts or software modules, according to which the software project can be organized and developed (Sucre and Chirinos, 2014). The proposed solution constitutes a framework based on best practices established by international standards, guidelines and models, and on national experience, which integrates fundamental activities for the detection of defects associated with performance efficiency in early stages of software development. It is based on the testing process, continuously monitoring this feature and is guided by the SQuaRE Series of ISO/IEC 25000:2011(ISO/IEC, 2005) for assessing software performance efficiency.

The framework aims to:

1. define and improve mechanisms for understanding performance efficiency behavior in the software lifecycle,
2. early detection of defects associated with performance efficiency in software.

Pursue as objectives:

1. detect defects associated with performance efficiency at the stage closest to where they were introduced,
2. feedback and monitoring of performance efficiency behavior in the software lifecycle,
3. make decisions in order to improve the performance of the software,

4. improve the quality of the software.

The components that make up the framework are:

1. quality activities,
2. methods and techniques to be applied,
3. tools,
4. human resources,
5. guides.

The framework proposed to evaluate the efficiency of performance from the early stages of software development can be applied independently of the development methodology used (agile or robust) and/or the duration of the software (small, medium or large projects), as it all depends on the planning, selection and frequency of the activities to be carried out in the project. The levels at which the activities are to be carried out must be assessed on the basis of the organization is structure. The framework is composed of a sequence of logical steps grouped into four stages (see Fig. 1): Planning and Monitorization, Analysis and Design, Implementation and Execution, and Analysis and Communication of the Results of the activities associated with performance efficiency. The inputs and outputs are work products generated in the research.

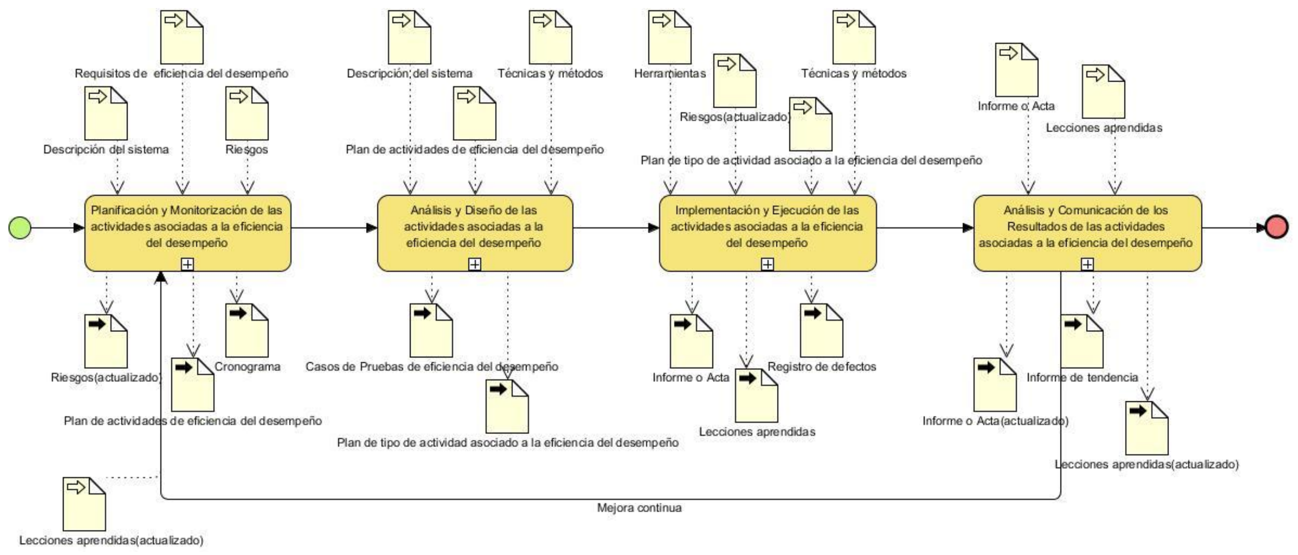
Requisitos de eficiencia del desempeño
Descripción del sistema
Riesgos
Descripción del sistema
Técnicas y métodos
Plan de actividades de eficiencia del desempeño
Herramientas
Riesgos(actualizado)
Técnicas y métodos
Plan de tipo de actividad asociado a la eficiencia del desempeño
Informe o Acta
Lecciones aprendidas
Planificación y Monitorización de las actividades asociadas a la eficiencia del desempeño
Análisis y Diseño de las actividades asociadas a la eficiencia del desempeño
Implementación y Ejecución de las actividades asociadas a la eficiencia del desempeño
Análisis y Comunicación de los Resultados de las actividades asociadas a la eficiencia del desempeño
Riesgos(actualizado)
Cronograma
Plan de actividades de eficiencia del desempeño
Casos de Pruebas de eficiencia del desempeño
Plan de tipo de actividad asociado a la eficiencia del desempeño
Informe o Acta
Registro de defectos
Lecciones aprendidas
Informe o Acta(actualizado)
Informe de tendencia
Lecciones aprendidas(actualizado)
Mejora continua
Lecciones aprendidas(actualizado)

Fig. 1 - Stages of the framework for assessing performance efficiency.

Font: Own elaboration.

The following preconditions must be met in order to implement the framework:

1. Have clearly identified requirements associated with performance efficiency,
2. Have defined roles and responsibilities in the project,
3. Have identified risks that may affect the efficiency of software performance.

In case requirements and risks are not defined, guidelines are proposed to identify these aspects from the beginning of the software.

Stage 1. Planning and Monitoring of activities associated with performance efficiency: In this stage, the objectives of the test are deduced, information and data associated with the activities of quality performance efficiency are collected. Risks and action plans are identified to address the occurrence of any problems in the performance of the component or system. It is recommended that, during the planning of the quality activities of the project and institution, these activities are taken into account as they depend on the allocation of human and technological resources for their fulfilment. As input artefacts, the description of the system, the performance efficiency requirements and the risks are necessary. As output are the updated risks, the test schedule and the plan of performance efficiency activities.

Stage 2. Analysis and design of the activities associated with performance efficiency: In this stage, the activities to be carried out to assess performance efficiency are analyzed and designed. Operational profiles are defined and created by analyzing load and stress profiles and monitoring time. As input artefact the system description, plan of performance efficiency activities and the selection of techniques and methods to be used are needed. The output is the design of performance efficiency test cases and the plan for each type of activity to be performed associated with performance efficiency.

Stage 3. Implementation and execution of the activities associated with performance efficiency: In this stage, the tools and environment involved in the execution of the activities are configured, and the correct functioning of the environment is deployed and verified. The defined techniques and methods are then used and the activities that assess the performance efficiency behavior are executed. As an input artefact, it is necessary to have updated risks, the plan for each type of activity to be carried out associated with performance efficiency, the techniques, methods and tools to be used. The output is a report with the results of the activities, the register of defects and the lessons learned during this process.

Stage 4. Analysis and Communication of the Results of the activities associated with performance effectiveness: In this stage, a comparative analysis of the results is carried out through the following steps and tables for a better understanding. Conclusions are drawn and recommendations are proposed. Analyze trends in activities related to performance efficiency and lessons learned. The results of the associated activities carried out are then communicated. The input artefact is the compliance report and the lessons learned. The output is an analysis of the lessons learned, an update of the final report or minutes and a tendency report.

Quality Activities

The selection of the quality activities to be used in the proposed framework for assessing performance efficiency from early stages of software development was based on the analysis of the literature review of standards, guidelines, international models and authors covering the subject. In addition, the cri- tics of specialists from national organizations with experience and knowledge on the subject in question were taken into account. For this purpose, experts were consulted by means of focus groups and questionnaires, which allowed their criteria to be ascertained. An important step for the results to be as accurate as possible is the appropriate selection of the experts. Therefore, for this research it was decided to use the curricular synthesis analysis for the selection of the experts, considering practical experience in the object of the research as the main factor to be taken into account. The results showed that the quality activities to be proposed in the framework are: software testing with 100% approval, formal and informal technical reviews with 70% and peer reviews as an informal review with 60%.

- Software testing

Objectives: to check that the software meets the performance efficiency requirements established at the beginning of the software development and to monitor the behavior of this characteristic in the software life cycle to prevent or detect defects.

Frequency: to be performed for each test level according to the test objective and the object to be tested during the software development.

Work products: software architecture, code fragments, components, software or system, database, services.

Expected results: early detection and resolution of defects associated with performance efficiency in the tests performed for resolution without going to the next level and knowledge that the performance efficiency behavior is acceptable in the software development lifecycle.

- Technical reviews

Objectives: technical review of work products identified in the institution as deliverable to the client and early detection of defective work products associated with performance effectiveness.

Frequency: to be performed at the conclusion of each work product or project milestone during software development.

Work products: software performance efficiency requirements, software design and architecture, code fragments, components, software or system, database.

Expected results: early resolution of detected defects in work products associated with performance efficiency.

- Peer reviews

Objectives: early detection of defects in work products associated with performance efficiency and prevention of defects in design, architecture or coding that affect software performance efficiency.

Frequency: To be performed at each test level during software development according to the purpose of the test and the object to be evaluated during software development.

Work products: Software performance efficiency requirements, software design and architecture, code fragments, components, software or system, database, services.

Expected results: Early resolution of detected defects in work products associated with performance efficiency.

The proposed quality activities are implemented at all levels of the institution, therefore, all those involved must be highly committed. It is proposed that these activities start from the project, which is the basic unit of software development. Then, if there is a management between the project and the institution, an intermediate level is proposed to check the efficiency of the software performance. Finally, the institutional level constitutes the last filter before delivery to the customer. For small organizations, it is proposed to carry out the project level, using the same process to carry out all quality activities.

Techniques and methods to be applied

For the selection of the techniques and methods to be used in the proposed framework to assess performance efficiency from early stages of software development, a bibliographic analysis of best practices for assessing performance efficiency in the software lifecycle was carried out according to standards, guidelines, models, etc., as well as a review of the best practices for the evaluation of performance efficiency in the software lifecycle and recognized authors in the field. In addition, a survey was conducted with 10 specialists with more than 8 years of experience in software production and practical knowledge of the research topic from XETID, CALISOFT, SEGURMÁTICA, DESOFT and UCI. By Finally, the criteria of the already selected experts were used to obtain the data necessary to identify the techniques and methods to be used through interviews.

Table 1 - Summary of methods and techniques by quality activity.

Activities	Methods and Techniques
Software testing	• Performance-related tests with a 100% pass rate. • Black box testing using the techniques: – Equivalent partitioning with 80% compliance.

	– Analysis of border values with 70% compliance. • Acceptance tests with 80% compliance.
Technical reviews	• Checklists with an 85% approval rate. • Documentary review with 75% approval., • Interviews with a 70% pass rate. • Proofs of concept with 70 (software architecture, database)
Peer reviews	• Checklists with an 80% pass rate. • Interviews with a 75% pass rate. • Component testing with a 70% pass rate.

Own elaboration.

Tools

The use of a tool for measuring and analyzing performance efficiency in the framework is necessary to evaluate this characteristic from early stages of software development as it would be very complex to manually re- present several test scenarios as it requires significant resources. There is a wide range of tools available for software performance testing, and the approach to testing varies according to the context and the object to be evaluated. For this reason, general characteristics are proposed that the tools must comply with in order to be used in projects and organizations that decide to evaluate performance efficiency. These are listed below:

1. Know the compatibility, scalability and capacity of the tool(s) for performance testing as it needs to be understood and established at the organizational level.
2. Have technical knowledge on the use of the tool to know the configuration requirements and data generation/loading capacity. This sometimes requires time for training and self-study.
3. Open source tool to generate low costs and with a large community to enable feedback and accessibility to information.
4. Multi-platform tool enabling its access and use to be applicable in various architectures and operating systems.

For the proposed framework it is decided to use Apache Jmeter as it fulfils the above-mentioned characteristics according to the research objective.

Human resources

Experienced human capital is meritorious as it creates an environment of trust and confidence that provides a vision of learning and improvement during the evaluation process. According to the quality activities to be developed, the roles and competences included in the proposed framework are defined.

Pool of experts: It is proposed that a pool of experts be set up to make use of the experience of the institution's best specialists to ensure the quality of the proposed activities.

Roles for Software Testing: Test Coordinator, Test Analyst/Test Specialist and Tester.

Roles for Technical Reviews: Lead Reviewer, Expert Reviewer and Reviewer.

Roles for Peer Reviews: Lead Reviewer and Reviewer.

Guides

In the proposed framework for assessing performance efficiency from early stages of software development, guidelines are defined to support the software development process. The proposed guidelines are:

1. Guidance for defining non-functional requirements associated with software performance efficiency.
2. Guidance on identifying and recording risks associated with software performance efficiency.
3. Guidance for planning quality activities associated with software performance efficiency.
4. Guidance for proposing methods, techniques, expected results, resources and environments for carrying out quality activities associated with software performance efficiency.
5. Guidelines for defining measures associated with software performance efficiency.
6. Guidance for the configuration of the Jmeter tool when performing performance tests.

The guides: Define requirements and Identify risks respectively, propose a list of questions to capture each of these elements from the initial stages. In addition, the requirements defined by CALISOFT(CALISOFT 2017, Alvarez, Aldana et al., 2018) are proposed as the base requirements, and risks associated with each subcharacteristic and good practices to prevent and mitigate them are defined. The proposed measures are associated with the quality measures defined in ISO/IEC 25023 (ISO/IEC, 2016).

The main element in the proposed planning guide is the frequency with which quality activities are proposed to be carried out and the levels at which they are to be implemented. In the case of a small organization, the implementation of the framework can be considered at the project level, using testing and peer review as the fundamental method for carrying out the activities. Depending on the type of quality activity to be performed and the artefact to be released, a guide is formulated with the specific methods, techniques, tools, expected results, tolerance, resources and environments to undertake the different quality activities. A guide is presented to explain the sequence of steps associated with the installation, configuration and administration of the JMeter tool for performance testing.

Validation of the proposal

In order to find out whether the framework contributes to the solution of the research problem, expert judgement was taken into account. For the selection of the experts, an analysis of their curricular synthesis was carried out, considering as selection criteria work experience in the software industry of 6 years or more, scientific production related to the evaluation of performance efficiency in software projects and scientific production focused on the object to be evaluated in the research. As a result of the above analysis, 15 experts were selected at the national level from the following countries institutions: Segurmática, DESOFT, XETID, UCI, CUJAE and CALISOFT. The Delphi method was then applied to take advantage of the common elements in the group of experts. Three rounds were carried out in order to combine and finalize the results. After the analysis of the answers, it became evident that the results were satisfactory, as all the categories were evaluated as Very High or High. The results of the votes cast by the experts are shown below:

Table 2 - Percentage of expert judgement.

Criterion	**Percentage**	**Fashion**
Relevance	93.3	5
Pertinence	86.6	5

Coherence	73.3	5
Understating	66.6	4
Accuracy	60.1	4

Font: Own elaboration.

Based on the above results, it is considered that the experts agree that the proposed framework for the evaluation of performance effectiveness from the early stages of software development enables the early detection of defects associated with this characteristic and an increase in the quality of the final product. In addition, the Iadov technique is applied to measure the variables:

1. Customer satisfaction: the institution's top management believes that the implementation of the framework increases the quality of the final product.
2. Applicability of the framework: Senior management believes that the components proposed in the framework allow for applicability in the different environments.
3. Usefulness of the framework: it allows early detection of defects associated with performance efficiency.

After the analysis of the questions and the position of each answer on the satisfaction scale, the result of the individual satisfaction of 38 users was as follows:

Table 3 - Individual results by level of satisfaction.

Satisfaction level	**Quantity**	**%**
Clear satisfaction	29	76.4
More dissatisfied than satisfied	1	2.6
Not defined	2	5.2
More satisfied than dissatisfied	5	13.2
Clear dissatisfaction	0	0
Contradictory	1	2.6

Own elaboration.

The Group Satisfaction Index (GSI) is calculated using the formula(Alonso, Hernández et al., 2020, Tinajero, Catota et al., 2021). In this case, the ISG value was 0.81, which indicates satisfaction with the proposed framework, as the result was between 0.5 and 1. This was followed by the two open-ended supplementary questions. These were:

1. Would you use the proposed framework for early detection of defects associated with performance efficiency?
2. What elements do you think you would add to the proposed framework? Which ones?

The feedback was positive and from the open questions users expressed that it would be very beneficial to include continuous integration tools for the execution of automated tests associated with performance, starting from the unit or component of the system to the integration of the code and the overall functioning of the software.

Conclusions

After carrying out the research, it is concluded that:

1. International models, norms, standards, methodologies and authors have enabled the incorporation of best practices to be mainstreamed for the early incorporation of performance efficiency in software development.
2. The national trends identified through a diagnosis revealed the need to establish quality activities, techniques and methods, roles, responsibilities, tools and guidelines to incorporate performance efficiency in the Cuban software industry from early stages.
3. A framework for incorporating early stage performance efficiency in software development was defined, which integrates good practices such as: quality activities, techniques, methods, tools, roles, responsibilities, guidelines and work products.
4. The application of the framework makes it possible to understand and evaluate the performance efficiency behavior of the systems, detecting at an early stage the shortcomings associated with this characteristic and the increase in the quality of the final product.

References

Alonso, L. A., K. M. Hernández, Y. A. J. N. C. Alvarado and M. Learning (2020). "IADOV neutrosófico para validar la concepción pedagógica del proceso de formación del léxico pedagógico de los estudiantes de la carrera Español-Literatura." 59.

Alvarez, A. R., L. M. Aldana and A. E. G. Rodríguez (2018). "PROPUESTA DE REQUISITOS DE USABILIDAD Y EFICIENCIA DE DESEMPEÑO PARA NORMA RAMAL: REQUISITOS DE CALIDAD PARA SISTEMAS INFORMÁTICOS Y PRODUCTOS DE SOFTWARE." Informática 2018: 8.

Benjumea, L. C. (2019). PROCESO DE TESTING FUNCIONAL DE SOFTWARE PARA LAS MIPYMES DESARROLLADORAS DE SOFTWARE DE LA CIUDAD DE PEREIRA. MAESTRÍA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE UNIVERSIDAD AUTONOMA DE MANIZALES

CALISOFT (2017). REQUISITOS DE LA CALIDAD PARA SISTEMAS INFORMÁTICOS Y PRODUCTOS DE SOFTWARE. CALISOFT. La Habana, Cuba, 03/2017**:** 12.

CMMI®, S. E. I. (2010). CMMI® for Development, Version 1.3.

Diaz, A. M., Y. T. Casañola and D. B. Hidalgo (2018). "Marco de Trabajo para gestionar actividades de calidad." **12, No. 2**.

Diaz, A. M., Y. T. Casañola and D. B. Hidalgo (2020). "Estrategia de pruebas para organizaciones desarrolladoras de software." Revista Cubana de Ciencias Informáticas **Vol. 14, No. 3**: 22.

Estrada, A. F., T. C. García, Y. L. Perdomo, A. V. Cintra and D. Martínez (2011). "Una experiencia novedosa para el testing desarrollada por un departamento de pruebas de software." Revista Cubana de Ciencias Informáticas: 15.

FOCUS, M. (2013). Application Performance Engineering. Application Delivery Management.

Globe. (2017). "Pruebas de rendimiento en eCommerce." 2020, from https://www.globetesting.com/2017/01/pruebas-de-rendimiento-para-ecommerce/.

González, M. P., A. M. Diaz, Y. T. Casañola and D. B. Hidalgo (2021). "Buenas prácticas para prevenir los riesgos de la eficiencia del desempeño en los productos de software." Revista Cubana de Ciencias

Informáticas **Vol. 15, No. 1**: Pág. 89-113.

Granma. (2017). "Un modelo para certificar el software cubano.", 2020, from http://www.granma.cu/cuba/2017-10-03/un-modelo-para-certificar-el-software-cubano-03-10-2017-22-10-32.

International Software Testing Qualifications Board, I. (2020). "International Software Testing Qualifications Board." International Software Testing Qualifications Board, 2020, from http://www.istqb.org.

International Software Testing Qualifications Board, I. (2020). ISTQB® Effectiveness Survey. I. S. T. Q. B. ISTQB®**:** 21.

ISO/IEC (2005). ISO/IEC JTC1/SC7 /N3163 25000:2005 Software product quality requirements and evaluation (SQuaRE) - Guide to SQuaRE. FCD 25000 - Software Engineering – Software product quality requirements and evaluation (SQuaRE) - Guide to SQuaRE. CANADA CANADA (SCC)**:** 52.

ISO/IEC (2011). ISO/IEC 25010:2011 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models, Switzerland**:** 44.

ISO/IEC (2016). ESTÁNDAR INTERNACIONAL ISO/IEC 25023 Sistemas e ingeniería de software Calidad de sistemas y software Requisitos y Evaluación (SQuaRE) - Medición de la calidad del producto del sistema y software. Switzerland**:** 61.

ISTQB, I. S. T. Q. B. (2014). Effectiveness Survey**:** 39.

ISTQB®, I. S. T. Q. B. (2018). Worldwide Software Testing Practices Report, International Software Testing Qualifications Board ISTQB®**:** 23.

Kaur, A., P. Grover and A. Dixit (2019). Performance Efficiency Assessment for Software Systems. Software Engineering, Springer**:** 83-92.

Micro Focus, C. G., Sogeti (2020). World Quality Report pp. 68.

Mina, M. A. E. and D. d. P. G. Barzola (2017). "La industria del software en Ecuador: evolución y situación actual.," **Vol. 38 Nº 57**: 6.

Muñoz, J. C.-R. (2018). Asistente para la evaluación de la calidad del producto de software según la familia de Normas ISO/IEC 25000 utilizando el enfoque GQM.

Paz, J. A. M., M. Y. M. Gómez and S. C. Rosas (2017). "Análisis sistemático de información de la Norma

ISO 25010 como base para la implementación en un laboratorio de Testing de software en la Universidad Cooperativa de Colombia Sede Popayán." 4to Congreso Internacional AmITIC 2017, Popayán, Colombia.: 6.

Sánchez, I. J. M. (2015). Marco de trabajo para el uso de la tecnología en el proceso de monitoreo y control de proyectos de software Máster Máster, Universidad de las Ciencias Informáticas

Sucre, F. S. and D. P. Chirinos (2014). "Marco de trabajo para gestionar las competencias laborales." Revista Venezolana de Información, Tecnología y Conocimiento **vol. 11, núm. 3**: pp. 11-32.

Tinajero, M., V. Catota and E. J. P. U. R. d. C. A. y. E. Catota (2021). "LA TÉCNICA DE IADOV. NIVELES DE SATISFACCIÓN DEL CLIENTE EN RM LATACUNGA–MALTERÍA PLAZA AÑO 2019." **4**(1): 110-120.

Zaraket, W. M. y. F. A. (2016). Coverage-Based Software Testing: Beyond Basic Test Requirements. Advances in usabilidad. . XIV Workshop de Investigadores en Ciencias de la Computación.

Conflict of Interest

The authors of this article authorize the distribution and use of their article.

Authors' contributions

1. Conceptualization: Aymara Marin Diaz
2. Data curation: Yaimí Trujillo Casañola
3. Formal analysis: Maidelyn Piñero González
4. Acquisition of funds: -
5. Research: Maidelyn Piñero González
6. Methodology: Maidelyn Piñero González
7. Project Administration: Yaimí Trujillo Casañola
8. Resources: Raidel Páez Llopiz
9. Software: Maidelyn Piñero González y Raidel Páez Llopiz
10. Supervision: Yaimí Trujillo Casañola
11. Validation: Aymara Marin Díaz

12. Visualization: Maidelyn Piñero González y Raidel Páez Llopiz
13. Editing - original draft: Maidelyn Piñero González
14. Writing - proofreading and editing: Aymara Marin Díaz

Funding

No funding was necessary for the development of the research.