

Tipo de artículo: Artículo original  
Temática: Ingeniería y gestión de software  
Recibido: 02/02/2022 | Aceptado: 22/03/2022

## Testing strategy for organizations developing web portals

Estrategia de pruebas para organizaciones desarrolladoras de portales web

Alicia Chacón Rodríguez <sup>1\*</sup> <https://orcid.org/0000-0003-2631-6612>

Maidelyn Pinero Gonzalez <sup>2</sup> <https://orcid.org/0000-0002-6534-132X>

Aymara Marin Diaz <sup>3</sup> <https://orcid.org/0000-0001-5101-7804>

Yaimi Trujillo Casañola <sup>4</sup> <https://orcid.org/0000-0002-3138-011x>

<sup>1</sup> University of Informatics Sciences, Cuba. San Antonio Highway Km 2 1/2. [achaconr@uci.cu](mailto:achaconr@uci.cu)

<sup>2</sup> University of Informatics Sciences, Cuba. San Antonio Highway Km 2 1/2. [mpinero@uci.cu](mailto:mpinero@uci.cu)

<sup>3</sup> University of Informatics Sciences, Cuba. San Antonio Highway Km 2 1/2. [amarin@uci.cu](mailto:amarin@uci.cu)

<sup>4</sup> University of Informatics Sciences, Cuba. San Antonio Highway Km 2 1/2. [yaimi@uci.cu](mailto:yaimi@uci.cu)

\*Corresponding author. ( [achaconr@uci.cu](mailto:achaconr@uci.cu) )

---

### ABSTRACT

Web portals in the last decade have gained a high level of popularity among users who usually surf the Internet. This has led to customers becoming more rigorous every day and demanding innovative products of the highest quality and attractiveness. To do this, these products are tested, which are key activities in software development, since they help detect defects that would otherwise go unnoticed until the software is deployed. However, various studies prove that in most of these products only functional tests are carried out. This is a problem because the detection of the defect is far from the moment in which it is introduced, which affects

the costs of correction and lengthens the schedules of the project, without covering the structural and non-functional tests . This article describes a testing strategy that encompasses functional, non-functional, structural, and change-associated testing based on web portals. The strategy takes into account good practices documented in internationally recognized models, norms and standards. It incorporates the experience gained in testing these types of products. It integrates what to test from the tiered test type and how to test from describing objectives, typical test objects, test bases, approaches and responsibilities, typical defects and failure types, techniques, and strategies. The results of the expert assessment and a case study are shown for a better understanding of the proposal.

**Keywords:** quality; defects; strategy; portal; tests; website .

## RESUMEN

Los portales web en la última década han ganado un elevado nivel de popularidad entre los usuarios que usualmente navegan en Internet. Esto ha conllevado a que los clientes sean cada día más rigurosos y exijan productos innovadores de altísima calidad y atracción. Para ello, estos productos son sometidos a pruebas, las cuales constituyen actividades claves en el desarrollo de software, puesto que ayudan a detectar defectos que, de otro modo, pasarían desapercibidos hasta que el software sea desplegado. Sin embargo, diversos estudios comprueban que en la mayoría de estos productos solo se realizan pruebas funcionales. Esto supone un problema pues se aleja la detección del defecto del momento en que se introduce, lo que incide en los costos de corrección y alargan los cronogramas del proyecto, sin cubrir las pruebas estructurales y no funcionales. El presente artículo describe una estrategia de pruebas que abarca pruebas funcionales, no funcionales, estructurales y asociadas al cambio en función de portales web. La estrategia tiene en cuenta buenas prácticas documentadas en modelos, normas y estándares reconocidos internacionalmente. Incorpora la experiencia adquirida en la realización de pruebas a estos tipos de productos. Se integra el qué probar a partir del tipo de prueba por niveles y el cómo probar a partir de describir los objetivos, objetos de pruebas típicos, bases de pruebas, enfoques y responsabilidades, defectos y tipos de fallas típicos, técnicas y estrategias. Se muestran los resultados de la valoración de expertos y un estudio de casos para mejor comprensión de la propuesta.

**Palabras clave:** calidad; defectos; estrategia; portal; pruebas; web.

## Introduction

Currently, new technologies are always present and within the reach of most people and companies, so the need to have a website increases day by day. Not only for those who want to make money online, but for all those companies or people who offer a service or have a business. (Erazo, 2012, Magaly, Rolando et al., 2020) Companies increasingly choose to use web portals, whose fundamental characteristic is to serve as a gateway to offer the user, in an easy and integrated way, access to a series of resources and services related to the same theme. Includes: links, search engines, forums, documents, applications, electronic purchase, etc. (Coloma, Pino et al., 2018, Magaly, Rolando et al., 2020). Mainly, an Internet portal is aimed at solving specific information needs on a particular selected topic, organizing and distributing content to meet the needs of its users/consumers (Dominguez, 2006, Fajardo, Escalona et al., 2018).

There are different types of web portals among them (Erazo, 2012, Pinho, Franco et al., 2018)

1. General Portals (Mega Portals or Horizontal Portals)
2. Specialized Portals.
3. Corporate Portals
4. Vertical Portals (Vortals)

It is important to clarify that, despite having different classifications, they have elements in common, they all offer resources, information services divided and organized depending on the type of public they are aimed at; allow personalization of information, which is displayed in an easy-to-use interface with a pleasant design (Santana Bonilla, Eirín Nemiña et al., 2017). Therefore, the proposed strategy can be applied regardless of these classifications.

Therefore, as these software systems are becoming more and more important in society, the demand for their quality increases. According to the Institute of Electrical and Electronics Engineers (IEEE), **software quality** is the degree to which a system, component, or process meets specified requirements and the needs or

expectations of the customer or user (IEEE 1990, A. I. Vlasov, 2022). Which depends largely on the tests that have been carried out during its development. Testing is therefore one of the most important activities in software development. Without them, the software would contain many defects that would cause failures that could only be detected when the product is installed and used in a real production environment, thus causing the impact of these failures to be serious and the cost to solve the defects that cause them is large (Boehm and Basili, 2007, Diaz, Casañola et al., 2020, Srivastava, Kumar et al., 2021).

However, despite the importance of testing, in many organizations they are not given enough attention or are simply omitted from software product development planning. Many times, the pressure to have a finished product on a set date causes activities, mistakenly considered as expendable, to be eliminated or reduced in planning. In this way, the aim is to reduce costs and be able to have a product that can be delivered on time, although the drawbacks are usually much greater than the advantages of simply delivering “something”. Thus, eliminating testing activities, instead of accelerating software development, means that these products can never be considered truly finished, since it is possible that defects continually appear that must be resolved once the software has been deployed. In this context, a defect can be defined as an unexpected behavior of the system, which does not correspond to its requirements and specifications (Fernández, 2015, Chen, 2021). The tests are very expensive so they are left for the later stages of the project and do not cover all types of recommended tests. On the other hand, tests are the most important resource for evaluating the quality of software (Vásquez Romero, 2018, Vera, Valdivia et al., 2020); however, quality and testing are not the same thing. Quality is built into software throughout the engineering process, if it's not there when testing begins, it won't be there when it's finished. For this reason, they should focus on prevention and control activities from the beginning of software development. As stated in the principles of evidence. (David Flores Mendoza, 2019)

Like all software, web portals are based on principles such as: graphic design, navigability, web positioning, user load, performance efficiency, security, among others (Toledo, 2019). However, with the high demand for these products and the limited time for their production, developers choose to use methodologies with an agile approach, where functional and non-functional requirements such as reliability, functionality, security, scalability, usability, among others, they cannot be omitted since this lack would directly lead to the failure

of the project (Pasini, Ramón et al., 2006, Santana Méndez and Trujillo Casañola, 2010, Guamán Barbecho, 2019, Zhao Huang, 2021).

In the bibliographic review carried out, the authors were able to verify that there are problems in the detection of defects in the early stages of development of web portals (Aldana La Ó and Ramos Medina, 2009, Ruiz Tenorio, 2010, Vásquez Romero, 2018). evidences the scant attention paid to aspects as relevant as making an interactive design focused on maintaining the receiver's attention, which adapts to the specific characteristics of each user, and the efficiency of quickly displaying requested information, have been part of the background, ignoring Thus, quality characteristics set out in the ISO 25010 standard, causing deliveries of products with numerous defects in both functional and non-functional tests, but also a proposal is not made of what to test and how to test in the different stages of the development cycle to guarantee more attractive and efficient products to the user(ISO/IEC 2011, Sánchez, 2021).

In order to verify the above, an analysis of the defects detected in testing stages, at the national level, was carried out on this type of product. An interview was conducted with the National Center for Software Quality (CALISOFT), where the questions were aimed at knowing what were the most common defects in the portals and the quality characteristics of the product, according to the ISO / IEC 25010: 2016 Standard, which most affected were based on the number of associated defects in their tests. It was stated that the most affected characteristics in the evaluations are: functional adequacy, performance efficiency and usability(ISO/IEC, 2016, González, Diaz et al., 2021).

On the other hand, the data of the tests carried out at the University of Informatics Sciences in the last five years were analyzed. According to these data, the most affected characteristics are: functional suitability and usability. In the tests to evaluate the functional adequacy characteristic, the validation type defects were the most manifested. In the usability evaluations, the defects of visibility, aesthetics and design and those associated with consistency and standards are highlighted.

Analyzing these findings, it can be established that many of these defects are introduced in the early stages of the software. (Zamuriano, 2004, Ruiz Tenorio, 2010). The following table shows some of the most common defects in websites as the cause that originates them.

**Table 1** - Some of the most common defects in websites.

Disciplines software development	test level	Types of defects	Cause of defects
<b>Requirements</b>	Component, System	<ol style="list-style-type: none"> <li>1. Validation defects can be introduced</li> <li>2. Defects related to Response Time</li> </ol>	<ol style="list-style-type: none"> <li>1. Due to an incorrect survey or absence of the requirements (functional or non-functional)</li> <li>2. Deliberate deviation from requirements (due to time pressure, advances without authorization)</li> <li>3. Documentation errors</li> </ol>
<b>Analysis and design</b>	System	<ol style="list-style-type: none"> <li>1. Correspondence between artifacts</li> <li>2. Aesthetics and design</li> <li>3. Visibility</li> </ol>	<ol style="list-style-type: none"> <li>4. Inadequate communication between team members</li> <li>5. Design Errors</li> <li>1. Insufficient evidence</li> <li>2. Documentation errors</li> </ol>
<b>Implementation</b>	System	<ol style="list-style-type: none"> <li>1. Not checking the web in different browsers and resolutions</li> <li>2. Navigability.</li> </ol>	<ol style="list-style-type: none"> <li>1. Non-compliance with standards related to a web portal facing the Internet</li> </ol>

Font: Own elaboration.

Then, it can be established that, if tests are designed throughout the life cycle of the portals, with specific activities to deal with the introduction of these types of defects, the final quality of the product developed within the national territory will be improved, as well as the delivery times to customers and thus the satisfaction of end users.

Therefore, it is established as a **problem** to be solved for this research: What and how to test throughout the life cycle of portals, taking into account national experiences and good practices recommended in the most used models, norms and standards? The **objective** of the research is to develop a testing strategy to evaluate

web portals, which allows detecting defects closer to the moment they are introduced and reducing delivery times to the client and the satisfaction of end users.

## **Methods or Computational Methodology**

For the development of this research, the methods mentioned below were used. In addition, a brief explanation of the purposes for which they were used is provided.

### **Theoretical methods:**

1. Dialectical method for the critical study of previous works and to use them as a source of reference and comparison of results.
2. The analytical-synthetic method was used for the study of the bibliography about the most used quality models internationally.
3. The deductive hypothetical for the identification of the problematic situation and the solutions.

### **Empirical methods:**

1. Interview to obtain information in order to argue the problematic situation and the validation of the results.
2. The survey to obtain the experiences of the organizations.
3. The participant observation to obtain the necessary information for the approach of the problem, as well as to carry out the confrontation of the obtained results.
4. Statistical methods to assess the effect of the proposal.
5. The experimental method to check the usefulness of the data obtained from the implementation of the general testing strategy.

From the point of view of an investigation, a strategy must consider the following stages: Diagnosis, Approach of the general objective, Strategic Planning, Instrumentation and Evaluation. For this research, an alignment

of these stages with the characteristics of a testing strategy was carried out. The general objective of the proposed strategy is identified, which would be contained in the organization's quality policy based on the results of the diagnosis. Strategic planning is carried out by defining the objectives by test levels and later the instrumentation is described where it is explained what to test from the type of test by levels and how to test from describing the objectives, typical test objects, test bases, approaches and responsibilities, defects and types of typical failures and techniques and strategies (Díaz, Casañola et al., 2019, Díaz, Casañola et al., 2020). Initially, the development organization must take into account in its test policy the quality activities related to the tests. Feedback from these activities should be used to detect changes in risks so that planning can be adjusted. These activities may include scope, objectives, risks, planning for analysis, design, implementation, execution, and testing activities (2018, Board-ISTQB® 2018, Pinho, Franco et al., 2018). For the drafting of the policy, the use of brainstorming is proposed where members of the organization's management who can make decisions in the processes, those responsible for directing the quality processes and some actors of these processes participate, thus improving quality. of these products, decrease the delivery time to customers and increase the satisfaction of end users.

After establishing the policy, the test strategy to be executed is designed. The proposal is based on the one designed by the authors (Díaz, Casañola et al., 2020), see Table 2. Taking into account that this strategy is for all software products, it is complemented by proposing activities based on the aforementioned needs of the portals.

**Table 2** - Instrumentation of the test strategy.

	<b>Component</b>	<b>Integration</b>	<b>System</b>	<b>Acceptance</b>
<b>Objectives</b>	<ol style="list-style-type: none"> <li>1. Reduce risk</li> <li>2. Verify the functional and non-functional requirements of the component</li> <li>3. Find component defects</li> </ol>	<ol style="list-style-type: none"> <li>1. Reduce risk</li> <li>2. Verify functional and non-functional behaviors of interfaces</li> <li>3. Create confidence in the quality of interfaces</li> </ol>	<ol style="list-style-type: none"> <li>1. Reduce risk</li> <li>2. Verify that the functional and non-functional behaviors of the system</li> <li>3. Create confidence in the quality of the system</li> </ol>	<ol style="list-style-type: none"> <li>1. Validation that the system is complete and will function as intended</li> <li>2. Verify that the functional and non-functional behavior of the system</li> </ol>



	4. Prevent defects from escaping to other levels of testing	4. Find defects in the interaction between components 5. Prevent defects from escaping to other levels of testing 6. Verify integration between systems	4. Fault 5. Prevent defects from escaping production	3. Satisfy user, operation and legal or regulatory requirements
<b>Typical test objects</b>	1. Components/ classes/ units/ modules 2. Code	1. Subsystems 2. Databases Infrastructure 3. Interfaces	1. Applications 2. System configuration	1. Applications 2. Systems 3. System and data configuration
<b>test base</b>	1. Component Requirements 2. Detailed design 3. Code	1. Software and systems design 2. sequence diagrams 3. Specifications of interfaces and communication protocols 4. Use cases 5. Architecture	1. Specification of software and system requirements. 2. Risk analysis reports 6. Use cases, user stories 3. System behavior models 4. System and user manuals	7. business processes 8. user requirements 9. Regulations, legal contracts 10. Software Requirements Specification 11. Use cases, user stories 12. System, user and installation manuals.
<b>Approaches and responsibilities</b>	It's from the developer, it focuses on test automation, usually white box.  These tests must be carried out by the project team.	Component integration testing is the responsibility of the developer. Integration testing between systems is the responsibility of the tester.  These tests must be carried out by the project team.	Testers whose fundamental task is the detection of defects from the design of the test.  An independent testing team is proposed to the development team.	They are the responsibility of customers, users and system operators.  For this level, if there is clarity about what has been agreed between the parties, it is proposed that only the development

				team and the client intervene.
<b>Typical defects and failures</b>	1. Validation	1. Response time	1. Validation 2. Response time 3. Correspondence between artifacts 4. Aesthetics and design 5. Visibility	1. Response time 2. Not checking the web in different browsers and resolutions 3. Navigability.
<b>test types</b>	1. Functional testing 2. Structural 3. Regression and confirmation tests	1. non functional 2. Structural 3. Regression and confirmation tests	1. Functional testing 2. Non functional 3. Regression and confirmation tests.	1. Functional testing 2. Non functional 3. Regression and confirmation tests.
<b>Organization (Strategies and tests)</b>	1. Analytics (requirements) 2. methodical 3. It is recommended at this level to carry out a complete review of both the functional and non-functional requirements. 4. For structural tests, the use of the method to verify the correct use of coding standards. 5. Carry out confirmation tests for all the defects detected in the previous tests and confirmation tests based on the risk analysis. 6. At this level, the automation of all the proposed tests is proposed.	1. Analytics (requirements and risks) 2. methodical 3. based on models 4. For this level, a review of the functional and non-functional requirements with an analytical approach is recommended.  In the structural tests, compliance with standards and optimization of functions must be verified, as well as the correct integration between the components through architecture models. Confirmation tests must be carried out on all detected	1. Analytics 2. methodical 3. based on models 4. Reactive or based on experience  It is proposed to start with a complete review of the functional and non-functional requirements. Taking into account the available testing time, it can be done through an analysis of requirements or risks. Functional testing should be started in order to ensure some stability in the product before starting non-functional testing. Models made for the correct understanding of the business of the product can	1. Analytics 2. methodical 3. advisory At this level, unlike the rest, it is recommended to start with a method of functional and non-functional testing to validate that what has been agreed with the client is correctly built. Then you can continue with an analysis of the risks that the client considers. Regression and confirmation tests are proposed to be carried out, but it is necessary to be clear about the scope of the agreed contract, since there may be customer

		defects and confirmation based on a risk analysis.	be used through operational flows for functional tests.  Regression and confirmation tests should be used to verify that all detected defects were correctly corrected and that no new ones were introduced in said process.	requests that are not non-conformities but requests for change.  The consultative approach is proposed since at this level it is considered that the client should be the one to guide the tests.
--	--	--	--	---

Font: Own elaboration.

There is a group of specific characteristics in the development of national web portals, they have the characteristics that are content-oriented and focused on the user interface, favoring visual creativity and the incorporation of multimedia. Many of the sites are directed by objectives of content, promotion of products or services (Calderón Montero and Hernández León, 2007, Alvarez, Aldana et al., 2018). In this type of products, the majority of dissatisfactions either by the user/consumer or in the tests that are carried out are those related to validation, aesthetics and design and performance efficiency for them, the researchers propose as a result of the investigation activities more specific linked to detecting these most common defects in this type of products, such as review of requirements to verify that they contain in their specifications everything required by the client, enhance communication (essential requirement) in the development team, implementation of good practices of quality activities to mitigate future risks and errors, so that defects closer to the stage where they are introduced are detected, which will reduce correction costs from the very beginning of the product.

## Results and Discussion

The proposal will be applied to the University of Informatics Sciences, in which software products are developed, including web portals for different businesses (educational, health, sports, industries...) and for the

development of these products, methodologies based on in user stories. The quality policy that was identified from the focus group was:

Quality activities are carried out to prevent, control and improve the quality of products. These activities represent revisions of the specifications, the design, the coding with the aim of preventing possible failures that may exist in the web portals. As a strategy, it was selected to combine the analytical strategies: based on requirements , based on experience and methodical.

**Table 3 - Description of activities.**

<b>Feature: Functional Suitability</b>				
<b>Test levels</b>	<b>Most common defects</b>	<b>Quality activities for its prevention</b>	<b>Roles and responsibilities</b>	<b>Observations</b>
<b>Component</b>	Validation	<b>Exercise 1</b> : Review of requirements	external quality reviewer project analyst	In the formal technical reviews (RTF) it must be verified that all the requirements must be related to functionality, there must be a good definition (data type, limit values, input/output conditions, response to all kinds of situations), everything necessary for its understanding by the developer
<b>Integration</b>		<b>Activity 2</b> : Strengthen Analyst - Developer link .	Project analyst and developer	Good communication between both pillars of the team should be encouraged, since if there is poor communication between them, the product's objectives will not be achieved, influencing errors such as validation errors.
<b>Integration</b>		<b>Exercise 3</b> : Perform inspections on every code update	Project analyst and developer	Every time an embedded element on the screen such as buttons and forms that includes the insertion of data is updated or modified, the developer must communicate so that it can be tested if everything complies with the above stated in the requirements, and does not influence its correct operation. . (interface tests)
<b>Feature: Performance Efficiency</b>				
<i>Risks</i>			<i>Requirements</i>	

<ol style="list-style-type: none"> <li>Lack of resources on the server side</li> <li>Network with inadequate bandwidth</li> <li>Poor or weak operating system capabilities</li> <li>Poorly designed functionality and other hardware or software conflicts that can lead to degraded software performance</li> </ol>		<ol style="list-style-type: none"> <li>The infrastructure elements must be reviewed to ensure that they do not contain any known vulnerabilities, as well as the administrative tools used for the maintenance of the different components.</li> <li>Requests to the application must be met in less than 5 seconds.</li> <li>Requests to the DB must be attended to in less than 3 seconds</li> <li>Define performance efficiency requirements from the beginning of the software.</li> </ol>		
<b>Integration System or Acceptance</b>	Response time	<b>Activity 4:</b> Identify risks associated with software resources.  <b>Activity 5 :</b> Define efficiency requirements.	Analyst	From the beginning of the software, possible deficiencies must be defined in order to influence the performance of the system, as well as establish requirements that take into account the characteristics associated with temporal behavior, the use of resources and capacity.
<b>Feature: Software Usability</b>				
<i>Risks</i>		<i>Requirements</i>		
<ol style="list-style-type: none"> <li>Insecure communication with the client (dissatisfaction)</li> <li>Inadequate documentation</li> <li>Confusing information architecture (if the user does not find what they are looking for, they leave)</li> <li>Vulnerable software products</li> <li>High quality costs due to rework because the further the detection of defects is from the stage where they are introduced, the more expensive their elimination.</li> </ol>		<ol style="list-style-type: none"> <li>Define and correctly write non-functional requirements (usability)                             <ul style="list-style-type: none"> <li>Indicate to the user in which part of its structure it is located, that is, if it shows breadcrumbs</li> <li>Provide all the information involved in performing a certain action to the user (forms, descriptions of required fields, confirmation or error messages)</li> </ul> </li> <li>Carry out inspections of the documentation, there must be similarity with the product that is being developed.</li> </ol>		
<b>Integration System</b>	Aesthetics and design  Visibility  Navigability.	<b>Activity 6 :</b> Define a design according to the needs of the user  <b>Activity 7 :</b> Establish project quality standards linked to usability.  <b>Activity 8 :</b> Define if the product will be visible internationally.	Analyst, project leader, user, developer.	For this characteristic, it is essential to link the user from the beginning of the development cycle, since the perfection of the product starts from him.  The project manager must ensure that basic internal and unit tests are carried out with the user during development, as a prerequisite for Quality Control tests (verify that the screens and messages are clear to the user)
<b>Note :</b> A good survey of both functional and non-functional requirements must always be taken into account, since the development of the product is part of them. In addition, when creating user stories, take into account the proposal of a user interface prototype, as well as make clear the messages and specifications that must be shown to the user.				

Font: Own elaboration.

It must be taken into account that these proposed activities are associated to discover and mitigate the most common defects related to the development of web portals. For the evaluation of the given proposal, it was necessary to carry out a survey that would allow obtaining expert criteria. For the selection of these, the curricular analysis technique was applied. 15 experts participated with more than ten years in the software industry and from different software development organizations nationwide.

### Expert selection process

An assessment of the possible experts was carried out, the following were considered as the initial selection criteria: knowledge related to quality and quality management in software projects, work experience in the software industry of 8 years or more, practical experience as the main factor. Taking these criteria into account, a questionnaire for curricular knowledge was carried out. As a result, 15 experts were selected at the national level, from the institutions: DESOFT, XETID, ICU and CUJAE. After selecting the experts, taking into account the knowledge about the quality characteristics of the product and the practical experience they have about the value of complying with them, the Delphi method was applied to take advantage of the common elements in the group of experts. Anonymity was preserved through the use of communication flows that allow the participation of experts even if they are geographically dispersed (Trujillo Casañola, Febles Estrada et al., 2015, Díaz and Castañola, 2019)

Satisfactory results were obtained from the analysis of the answers given, since all the categories were evaluated as Very High or High, validating the contribution of good practices associated with minimizing defects in the development of these products, as well as activities necessary to incorporate from the beginning of the development of the software, as fundamental for the satisfaction of the users and acceptance of the product. A mode of High or Very High was obtained and the experts did not cast votes on the scale of Low or None. From the votes cast by the experts, the following results are obtained (Table 3):

**Table 3** - Percentage of the experts' criteria.

Criterion	Percent	Fashion
-----------	---------	---------

Relevance	97.4	5
Relevance	94	5
Coherence	86	4
Comprehension	90	5
Accuracy	64.7	4

Font: Own elaboration.

Based on these results, it can be ensured that the experts agree that the activities and good practices proposed have a positive impact on the prevention and control associated with these types of errors and that they must be incorporated from the beginning of the software development, being necessary to carry out a correct management of product requirements so that they do not negatively affect the operation of the system and its acceptance.

## Conclusions

1. Both functional and non-functional requirements must be correctly defined from the beginning of software development to obtain a design and architecture that correspond to the behavior of the product during its development to minimize the defects or failures that these may cause.
2. Carrying out software tests is important to reduce the risk of failure in operation, however, many times they are carried out after the product is finished and only functional tests are carried out.
3. The results reaffirm the need to carry out early tests in the development of the product, complemented with testing strategies and activities to obtain positive criteria at the end of the product.

## References

(2018) "Certified Tester Foundation Level Syllabus." International Software Testing Qualifications Board (ISTQB).

- A. I. Vlasov, B. V. A., and L. V. Juravleva (2022). "Quality estimation method in advanced software systems." AIP Conference Proceedings 2467.
- Aldana La Ó, N. s. and J. L. Ramos Medina (2009) "Propuesta de procedimiento general para pruebas de liberación de los productos de la Facultad 1."
- 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.
- Board-ISTQB®, I. S. T. Q. (2018). *Probador Certificado del ISTQB®*.
- Boehm, B. W. and V. R. Basili (2007). Software Defect Reduction Top 10 List. *Software Engineering: Barry W. Boehm's Lifetime Contributions to Software, Development, Management, and Research*. R. W. Selby. California, Estados Unidos: 34 (31): 135-137.
- Calderón Montero, M. and R. A. Hernández León (2007). Propuesta de procedimiento para la captura de requisitos en los proyectos de portales de la UCI.
- Chen, Z. (2021). "An Empirical Study on Deployment Faults of Deep Learning Based Mobile Applications." *IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*: pp. 674-685.
- Coloma, M. A. V., Á. M. B. Pino and D. E. L. Mora "MÉTRICA DE CALIDAD EN LOS PORTALES WEB."
- Coloma, M. A. V., Á. M. B. Pino and D. E. L. Mora (2018). "MÉTRICA DE CALIDAD EN LOS PORTALES WEB."
- David Flores Mendoza, I. M., Paola Tovar (2019). Creación de una metodología óptima y eficiente para la implementación de pruebas de software, Instituto Politécnico Nacional de México.
- Díaz, A. M., Y. T. Casañola and D. B. Hidalgo (2019). "Apuntes para gestionar actividades de calidad en proyectos de desarrollo de software para disminuir los costos de corrección de defectos." *Revista chilena de ingeniería vol. 27 N° 2*: pp. 319-327.
- Díaz, 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.
- Díaz, A. M. and Y. T. Castañola (2019). Aseguramiento de la Calidad del Proceso y el Producto. *Maestría de Calidad en Software*: 67.



Dominguez, M. A. C. S., A.M. Jorge (2006). A Web-Based System to Monitor the Quality of Meta-Data in Web Portals. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IATW'06).

Erazo, C. F. R. (2012) "DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL PORTAL WEB DE LA EMPRESA DE AUTO VENTAS "AUTO FÁCIL", APLICANDO LA HERRAMIENTA DE DESARROLLO WEB OPEN SOURCE DRUPAL." Repositorio Digital Universidad Técnica del Norte.

Fajardo, I. V., Y. R. Escalona and R. Morales (2018). "Portal web para GEDEME." Revista Publicando 5(14 (3)): 497-501.

Fernández, M. Á. F. (2015). Aplicación de técnicas de pruebas automáticas basadas en propiedades a los diferentes niveles de prueba del software.

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.

Guamán Barbecho, E. P. (2019). "Análisis y evaluación de la calidad del sitio web de la Universidad del Azuay."

IEEE (1990). "IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990)." IEEE Computer Society, Los Alamitos. CA.

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.

Magaly, L. M. N., M. R. J. Rolando, M. R. R. Fernando and P. S. C. Marcela (2020). Mensajería cliente-servidor aplicando sockets en las herramientas GEANY IDE 1.31, PHYTON 3.7 y POSTGRESQL 9.5 en el sistema operativo CENTOS 7. Conference Proceedings.

Pasini, A. C., H. D. Ramón and P. J. Thomas (2006). Testing en aplicaciones web. XII Congreso Argentino de Ciencias de la Computación.

- Pinho, C., M. Franco and L. Mendes (2018). "Web portals as tools to support information management in higher education institutions: A systematic literature review." *International Journal of Information Management* 41: 80-92.
- Ruiz Tenorio, R. (2010). "Las pruebas de software y su importancia en las organizaciones."
- Sánchez, K. M. M. (2021). *Análisis y evaluación del sistema Eproductions ERP para el cumplimiento de los requisitos de calidad de la Norma ISO/IEC 25010, en la Empresa Viaxperta S.A. Licenciatura en Sistemas de Información, Universidad de Guayaquil*
- Santana Bonilla, P. J., R. Eirín Nemiña and D. Marín Suelves (2017). "Análisis y evaluación de portales institucionales en España. Los casos de Canarias, Galicia y Valencia."
- Santana Méndez, W. and Y. Trujillo Casañola (2010). *Proceso de desarrollo para portales Web.*
- Srivastava, N., U. Kumar, P. J. J. o. I. E. Singh and E. Engineering (2021). "Software and performance testing tools." 2(01): 1-12.
- Toledo, M. A. (2019). "HERRAMIENTAS PARA TESTING DE APLICACIONES WEB." *Conexiones* 1(5): 140-144.
- Trujillo Casañola, Y., A. Febles Estrada and G. León Rodríguez (2015). "Modelo para valorar las organizaciones desarrolladoras de software al iniciar la mejora de procesos."
- Vásquez Romero, D. P. (2018). "Estrategia de pruebas de software en un proyecto full stack para minimizar el impacto sobre los clientes y usuarios en una empresa de telecomunicaciones."
- Vera, Y. P., J. J. G. Valdivia, S. M. Z. Quentasi, D. M. C. Yana and R. E. C. Apaza (2020). "Design Thinking en la Planificación de Pruebas de Software." *Innovación y Software* 1(2): 40-51.
- Zamuriano, R. F. (2004). *Las inspecciones de software y las listas de comprobación, Tesis de Maestría, Instituto Superior Politécnico, José Antonio Eceverría ....*
- Zhao Huang, J. M. (2021). "Gender differences in user perception of usability and performance of online travel agency websites." *ScienceDirect* 66.

### **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: Alicia Chacón Rodríguez
6. Methodology: Alicia Chacón Rodríguez y Maidelyn Piñero González
7. Project Administration: Yaimí Trujillo Casañola
8. Resources: Alicia Chacón Rodríguez
9. Software: Alicia Chacón Rodríguez y Maidelyn Piñero González
10. Supervision: Aymara Marin Díaz y Yaimí Trujillo Casañola
11. Validation: Aymara Marin Díaz y Yaimí Trujillo Casañola
12. Visualization: Alicia Chacón Rodríguez y Maidelyn Piñero González
13. Editing - original draft: Alicia Chacón Rodríguez
14. Writing - proofreading and editing: Maidelyn Piñero González y Aymara Marin Díaz

### **Funding**

No funding was necessary for the development of the research.