Acceptance Testing Process for Products Developed at the University of Informatics Sciences

Ismaila López Sotolongo, Eng.[1]

[1] University of Informatics Sciences, Cuba, isotolongo@uci.cu

Abstract– The conformity of clients with products developed by the university is the main goal of the Quality of the University of Informatics Sciences. In order to validate software systems tests are performed according to the needs and requirements of each clients. Acceptance tests are vital for determination by the client acceptance or rejection of the developed system. From an analysis center in college this research is developed, which proposes a process defined stages and their products, activities that must be performed, and the responsibilities of each stakeholder. Through these actions is to standardize and organize the process of acceptance for university projects.

Keywords— process, acceptance tests, client acceptance.

Acceptance Testing Process for products developed at the University of Informatics Sciences

Ismaila López Sotolongo, Engineer in Informatics Sciences
University of Informatics Sciences, Cuba, isotolongo@uci.cu

Abstract– The conformity of clients with products developed by the university is the main goal of the Quality of the University of Informatics Sciences. In order to validate software systems tests are performed according to the needs and requirements of each clients. Acceptance tests are vital for determination by the client acceptance or rejection of the developed system. From an analysis center in college this research is developed, which proposes a process defined stages and their products, activities that must be performed, and the responsibilities of each stakeholder. Through these actions is to standardize and organize the process of acceptance for university projects.
Keywords—process, acceptance tests, client acceptance.

I. INTRODUCTION

Nowadays software systems are essential in all organizations in the world, mainly in service businesses such as financial, the air traffic control and telecommunications systems. This makes it clear that the successful implementation of public mission depends on the software that supports it. Therefore these systems become more complex, time-to-market and should be reduced because of this development becomes more difficult.
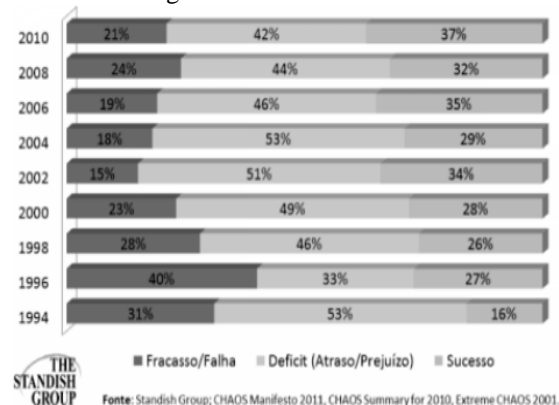
Although the software industry in the world has developed considerably in recent years, rapid development, the results achieved are not meeting expectations initially glimpsed basically because productivity is reached in general is low, the amount of resources to consume (in time mainly) is high and the work done rarely has the required quality. The projects are completed at a later date as planned and no problems are detected early in the development unruly and chaotic environment.

According Pointe Technology Group. Inc. [1] the five most common problems in the software development process are:
• Poor requirements: if requirements are unclear, incomplete, too general, or not provable, then there are big problems.
• Schedules unrealistic: if the work to be performed is much, too little time, problems are inevitable.
• Inadequate evidence: no one will know if the program is good until the customer complains of an accident or the system itself.
• New functions, features: new functionality or features after development is underway are required, this is extremely common.
• Lack of communication: if developers do not know what is needed, or the customer has wrong expectations, problems are guaranteed.

The Chaos Manifesto 2011 Standish Group [2], an analysis of the results achieved by software development projects where evidence is made that between 2008 and 2010 34.5% of the projects was successful, while 43.0% had problems with delays and an average of 22.5% were unsuccessful. Observe Figure # 1:

Fig. # 1 Chaos Manifiesto 2011. Standish Group

The research Standish Group Chaos Manifesto 2014 shows that 31.1% of the projects will be canceled before being completed. Other results indicate that 52.7% of projects will cost 189% of its original estimate. The main causes of failures identified in the software industry have been excesses in the cost and time of projects and the large percentage of features and functionality originally specified that are not available in the final product. [3]

Therefore we can deduce that the growing demand for software development does not correspond to the quality of execution of processes that allow collection. Product development of high quality software is therefore a matter of prime importance.

In recent years, software companies have opted for process improvement as a way to ensure the quality of their products. Software testing are activities of the process of control of the quality that provide information about the quality of the product to the interested party.

There are different types of tests that can be implemented at any time during the process of development, there are also defined four levels of evidence to clearly understand the different points or stages where certain types of testing can be performed, these are: unit testing , integration testing, system testing and acceptance testing.

Acceptance tests are designed to validate that the system meets the basic requirements expected performance and allow the user to determine the acceptance of the system. The University of Informatics Sciences (UCI), located in Havana, Cuba, has defined a process to guide the implementation of the acceptance tests of their projects.

1

II. Development

The development of software systems involves a series of production activities in which the chances of human error appears are enormous. Mistakes can begin to be from the beginning of the process, in which the objectives can be specified incorrectly or imperfectly and in subsequent steps of design and development. Because human inability to work and communicate perfectly, software development must be accompanied by a quality assurance activity.

International Organization for Standardization (ISO) 8402 [4] defines quality in terms of the ability to satisfy stated and implied needs you. 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

According to Pressman [5] is the correlation of the software produced explicitly stated requirements, standards development and prefixed with the implicit requirements are not formally established, the user wants.

A defining software quality should differentiate between product quality and quality of software development process. However, the goals established for product quality will determine the goals set for the quality of the development process, since the quality of the product will be a function of the quality of the development process. Without a good development process is almost impossible to get a good product.

Some authors such as Bauer [6], Boehm [7], Humphrey [8], Sommerville [9] and Pressman [5] have conducted research that reflect the need for a led development processes for greater efficiency in the performance of organizations; and ensure that the processes implemented in the organization are developed with the required quality.

The results of other studies argue that it is not only important to establish a perspective directed to processes for the fulfillment of the objectives of the organization from a quality approach, but the definition of initiatives that contribute to the continuous improvement of these processes is essential, to achieve a strategic maturity in the organization and therefore provide quality services in a competitive market. [10] [5] [11] [12]

An increase in these investigations that are intended to strengthen the software industry focusing on improving the quality of software processes (MPS) is highlighted. MPS initiatives focus on improving the performance, usefulness and effectiveness of processes in a disciplined manner. [13]

Some research [14] [15] reaffirm the importance of MPS overlooking the introduction of good practices in the implementation process models to increase the maturity and capacity thereof.

In recent years producing software organizations have chosen to enhance their development process by using quality models. Models of software lifecycle serve as high-level definitions of the phases that occur during a process. Not intended to provide detailed information, but provide the main activities and their interdependencies. [16]

A quality model is the set of characteristics and relationships between them which provide the basis for specifying quality requirements and evaluating quality. [4]

Process models are important because they provide order in which the project must perform their duties. [17]

Among the most used models are currently ISO / IEC 9126, created by the International Organization for Standardization in conjunction with the International Electo-Technical Commission (IEC) and Capability Maturity Model Integration (CMMI).

The CMMI model has an approach to process improvement, is composed of 5 levels, where each level includes a set of process areas (PAs). An AP is a group of interrelated that collectively help to achieve a set of objectives practices. Level 3 model has defined the AP Validation, which manages the processes related to the validation of computer systems. Validation tests whether the expected results actually obtained, ie this process ensures that the manufactured software behaves as expected and according to customer expectations. The purpose of validation is to demonstrate that a product or product component fulfills its intended use when placed in its intended environment. [18]

The IEEE 610.12-1990 standard [19] defines validation as the process of evaluating a system or component during or at the end of the development process in an effort to determine whether it satisfies specified requirements.

The validation process includes among its activities testing to representative users can successfully achieve their intended tasks using the software product. [12]

The application of systems tests can be run at any point in the process of software development.

One of the most neglected in this great process of development aspects is the test, despite being an obligatory reference when talking about improving quality. These, though not directly provide quality system, if they give a clear view of the weaknesses and associated risks, which can work to perfect, being classified as a crucial technique in quality and customer satisfaction. [20]

The IEEE defines testing as an activity in which a system or component is executed under specific conditions are observed or store the results and an evaluation of some aspect of the system or component is performed. [19]

Software tests are performed in order to detect errors, corrected, or improved quality reliability. There are different types of tests depending on the software unit to which it is applied and the objective to be achieved, for example, unit testing, integration, system and acceptance. [21]

From the point of view of the customer and users, the quality of a software product is mainly perceived by the flaws in the product and the severity that these have for the customer's business. To be competitive, software development companies need to ensure the quality of their products before installation in the customer environment. [22]

Acceptance tests are validation activities carried out by the client just before the system is delivered and are directed to judge whether the product is acceptable. Usually, the

acceptance tests are conducted throughout the system, or largely the same. [23]

The IEEE 1012-1986 [24] defines acceptance tests as formal tests are conducted to determine if a system fulfills the acceptance criteria and allows the user to determine whether to accept the system.

Another definition of acceptance tests given by the IEEE standard states that are formal evidence regarding user needs, requirements and business processes aimed at determining whether or not the system satisfies the acceptance criteria and to enable the user, customer or other entity authorized to determine whether to accept the system. [19]

The UCI has a training model, which is based on the interrelationship of training, research and production as well as provide support for the Cuban computer industry. Since the beginning of productive projects in college quality of the products has been a challenge, so the line of customers with products that are developed is the main goal of the Quality.

In 2008 three centers in the UCI were assessed at Level 2 of CMMI for Development v1.2. The university aims in 2015 a new program of improvements to opt for a satisfactory assessment of CMMI v1.3 for development in their maturity level 2 and make a diagnosis to determine the level of implementation of process areas of level 3 of this model in production development activity.

One of the process areas of CMMI Level 3 is Validation. The purpose of this research is to define a process that implements the generic and specific practices that sets the CMMI model, with a view to certifying university centers at Level 3 of the model. The acceptance of software products is a key to run within the computer system validation thread. The scope of this article within the research involves a study of the way in which university centers conducting the acceptance of the products they develop.

Currently not defined a process for acceptance tests products in the university. The acceptance tests right now performed by the project itself; the Quality does optionally, only if the development team upon request. It has been identified that there is loss of knowledge and information of a process to another acceptance tests.

Knowledge of the process of the acceptance tests is left alone in the specialists and testers players in the process. Furthermore, due to the characteristics of staff replacement is inevitable at some point and knowledge transfer mechanism is complicated; because the experience and information so that you can retrieve and reuse is not stored, also causing the same mistakes. Similarly it can be seen that the selection of test equipment is performed each time running acceptance tests to a product. This causes them to invest time in training some specialists who do not have the necessary skills to execute some specific tests.

From the identified activities and difficulties detected it is elaborated a process that standardizes the acceptance of products in the UCI.

III. RESULTS

For this research surveys to Quality Advisors of the 14 centers of the University. From the information obtained various problems were detected in 90% of the centers occurs that:

• The project leader is responsible for the acceptance tests.
• There is no standard for developing plans and test reports.
• The staff of the acceptance tests does not verify the accuracy and completeness of development documents (requirements, use cases, user stories, test cases).
• The defects discovered in the PA are not recorded and evaluated.
• The planning and assessment processes are performed acceptance tests consideration of the person in charge at the time.
• No defects defined expectations based on previous experiences.
• None automated tool is used to manage information process acceptance tests.
• The acceptance tests are made only to the final product, not the product components or any artifact development activities.
• In many cases the requirements and restrictions for the environment of the acceptance tests are identified when tests are planned.

The 100% of respondents said that the defects to the development team, 30% said that if necessary to senior management reporting communicate. In this it is necessary to highlight the influence that most of the time the testers are the members of the development team. Regarding the existence of processes to train staff of the tests, only 40% of respondents claimed to possess.

The difficulties identified above impacting negatively on production of the UCI. The final products reach the user with features that do not meet your expectations, causing exist change requests and delaying the delivery. Moreover not being evaluated defects discovered cannot be used to improve development processes and acceptance tests.

In the absence of a defined and institutionalized process becomes difficult to manage the accumulated experience and reuse; it also has a repository where the historical results of the acceptance tests are stored for easy management of that information. Also no continuity of work in the short term within the testing process itself, because the availability of the selected personnel to perform the PA is scarce and are different from one another test process is guaranteed. All exposed elements affect the compliance of clients with software developed at the university.

The process will consist of three stages which will make a series of activities by participants in the process. At each stage they are going to get work products resulting from the activities.

Stage 1 Preparation of Acceptance Tests
At this stage the types of tests set by the user in the solicitude in accordance with the characteristics possessed by the system on this basis the test plan is developed are analyzed.

Activities:
• Develop Test Plan.
• Prepare the test scenario.

- Communicate the plan to test involved.

Products:
- Input: Solicitude Testing
- Output: Test Plan

Responsibilities of participants:
- Tester Leader: Develops and communicates the test plan.
- Client: Set the stage of testing. Receive the test plan.
- Development Team: Prepare the test scenario. Receive the test plan.
- Test Equipment: Contribute to the development of the test plan. Receive the test plan.

Stage 2 Acceptance Tests Implementation

At this stage the final acceptance tests of the system is performed to ensure that all components meet the specified acceptance criteria.

Conducting tests recorded by a report containing the deviation from the requirements and problems that remain unresolved. Customer analysis is performed to inform the defects and thus keep them abreast of system status.

Activities:
- Run the tests.
- Collect the findings identified during testing.
- Analyze the defects detected.

Products
- Input: Lifecycle artifacts.
- Output: Result Acceptance Tests

Responsibilities of participants

- Tester Leader: Lead the testing process. Prepare the final report on the results, communicating the results to the development team and the client.
- Client: Test the system.
- Development Team: During testing their presence is not required but must be contactable during the entire time you are testing the system. You must take part in communicating the results to the client.
- Test Equipment: Test the system and gather the findings are.

Stage 3 Evaluation of the results of the acceptance tests

At this stage the test results are evaluated by analyzing the defects and checking that it has complied with the test plan. This evaluation consists of:
- Compare the results with those expected.
- Identify the source of every problem to refer who appropriate and determine what actions must be performed to resolve satisfactorily.
- Indicate what evidence you must perform

After performing the actions and found that their behavior is appropriate, the overall result of the evaluation of the acceptance tests including the approval of the system by the end user is documented.

Activities:
- Evaluate test results.
- Document the overall result of acceptance.

Products
- Input: Result Acceptance Tests.
- Output: Evaluation of the results of the acceptance tests.

Responsibilities of participants

- Tester Leader: Evaluate the outcome of the tests. Documenting the overall result of the evaluation.
- Client: Test and approve the system.
- Development Team: Evaluate the outcome of the tests. Perform actions to address the problems identified.
- Test Equipment: Test the system and gather the findings are.

For the proper execution of acceptance tests is essential that the client appoint personnel involved in the process of acceptance testing, it is even recommended that end users participate in this process are independent personnel who supported the process development. When the acceptance tests are carried out in facilities or rooms provided by the development firm called Alpha testing them, when executed from the client infrastructure are called Beta testing.

The process of acceptance tests defined in the investigation implementation conforms to both Alpha and Beta testing, because the fundamental difference of these tests lies in the environment in which they run. Generally only the structure of the products resulting from the process activities can be subject to change depending on the type of test (Alpha or Beta) running.

IV. CONCLUSIONS

From research could be obtained characterizing the state of the projects of the university when the activities of acceptance of its products. It was identified that currently does not have a standardized process to facilitate improved validation activities.

Through the proposed process is accomplished standardize the process of acceptance testing for the Quality of college. The same possible to organize and standardize the activities and artifacts to perform during the acceptance tests also be established roles involved, each with their responsibilities. A repository will be designed to store and facilitate the management of information generated during acceptance tests.

REFERENCES

[1] Pointe Technology Group, «Software Testing and Quality Assurance White Papers Disponible,» [En línea]. Available:

www.pointetech.com. [Último acceso: junio 2011].

[2] STANDISH-GROUP, « Chaos Manifiest,» 2011.

[3] STANDISH-GROUP, « STANDISH-GROUP CHAOS REPORT,» 2014.

[4] ISO, «Quality management and quality assurance -- Vocabulary». Patente 8402, 1994.

[5] R. S. PRESSMAN, Software Engineering. A Practitioner´s Approach. Translation of: Ingeniería de Software. Un Enfoque Práctico, Edtion ed. New York:McGraw-Hil1 Companies, 2010.

[6] F. L. BAUER, Software Engineering, Information Processing. In Proceedings of the Software Engineering, Amsterdam: North Holland Publishing, 1972.

[7] B. W. BOEHM, Software Engineering. Computers, IEEE Transactions on, 12/1976.

[8] W. S. HUMPHREY, A Discipline for Software Engineering, Boston, MA, USA: Edtion ed. Inc. Boston: Addison-Wesley Longman Publishing Co., Inc. Boston, 1995.

[9] I. SOMMERVILLE, Software Engineering. Edtion, ed.: Addison-Wesley, 2007.

[10] G. MATURRO, Modelo para la gestión del conocimientoy la experiencia integrada a las prácticas y procesos de desarrollo software, Universidad Politécnica de Madrid, 2010.

[11] L. M. A. P. POUYA, Managing knowledge in asoftware organization. Journal of Knowledge Management, 2003.

[12] ISO/IEC, Information Technology / Software Life Cycle Processes, 2004.

[13] N. ASHRAFI, The impact of software process improvement on quality: in theory and practice. Information and Management, 2003.

[14] S. ZAHRAN, Software Process Improvement: Practical Guidelines for Business Success., Edtion ed.: ADDISON WESLEY Publishing Company Incorporated, 1998.

[15] F. J. Pino, F. Garcıa y M. Piattini, Software process improvement in small and medium software enterprises: a systematic review., Software Quality Journal, 2008.

[16] IEEE, «Guide to the Software Engineering Body of Knowledge,» IEEE Computer Society. , 2004. [En línea]. Available: http://www.swebok.org.

[17] S. L. Pfleeger, Software Engineering,2nd Edition, Prentice Hill, 2001.

[18] SEI, CMMI para desarrollo, Versión 1.3, Software Engineering Institute http://www.sei.cmu.edu, 2010.

[19] IEEE, «Computer Dictionary. Software Engineering Terms». Nueva York Patente Standard 610, 1990.

[20] E. G. Valdivia, Estándares de calidad para pruebas de software, Lima, Perú: Sistemas e Informática, Universidad Nacional Mayor de San Marcos, 2005.

[21] J. Z. Hernández, Análisis de los procesos de verificación y validación en las organizaciones software, Madrid, 2011.

[22] B. P. Lamancha, Proceso de Testing Funcional Independiente, Uruguay: Instituto de Computación, Universidad de la República, 2006.

[23] R. Filippo, M. Torchiano, M. Di Penta, M. Ceccatod y P. Tonella, «Using acceptance tests as a support for clarifying requirements: A series of experiments,» Information and Software Technology, febrero 2008. [En línea]. Available: www.sciencedirect.com. [Último acceso: enero 2015].

[24] IEEE, «IEEE standard for software verification and validation plans.». Patente Std 1012-1986, 1986.