

Tipo de artículo: Artículo original  
Temática: Ingeniería y gestión de software  
Recibido : 15/04/2016 | Aceptado: 05/05/2016

## *Calculation of priorities of test cases from the functional requirements*

# **Cálculo de prioridades de casos de prueba a partir de los requisitos funcionales**

Martha Dunia Delgado Dapena <sup>1\*</sup>, Sandra Verona Marcos <sup>1</sup>, Perla Beatriz Fernández Oliva <sup>1</sup>, Danay Larrosa Uribazó <sup>1</sup>

<sup>1</sup> Facultad de Ingeniería Informática, Instituto Superior Politécnico José Antonio Echeverría, Cuba. 114, No. 11901 entre 119 and 127, Marianao, Código Postal: 19390, La Habana, Cuba. {[marta](mailto:marta.ceis.cujae.edu.cu), [sverona](mailto:sverona.ceis.cujae.edu.cu), [perla.dlarrosau](mailto:perla.dlarrosau@ceis.cujae.edu.cu)}@ceis.cujae.edu.cu

\* Autor para correspondencia: [marta@ceis.cujae.edu.cu](mailto:marta@ceis.cujae.edu.cu)

---

### **Abstract**

*This paper presents a proposal for prioritizing test cases based on the functional specification of software requirements of the project. Function is proposed for calculating the priority of test cases for software projects. Defined function is based on seven indicators whose information can be obtained from the initial stages of the project. The same shall include the results of application of the priority function of two real projects.*

**Keywords:** *Software Quality, Functional Test, Test Case, Software Requirements, Calculation of Priorities*

### **Resumen**

Este trabajo presenta una propuesta de asignación de prioridades a casos de prueba funcionales partiendo de la especificación de requisitos del proyecto de software. Se propone una función para el cálculo de la prioridad de los casos de prueba de los proyectos de software. La función definida se basa en siete indicadores cuya información es posible obtener desde las etapas iniciales del proyecto. De igual forma se incluyen los resultados de aplicación de la función de prioridad en dos proyectos reales.

**Palabras clave:** Calidad de Software, Pruebas Funcionales, Casos de Prueba, Requisitos de Software, Cálculo de Prioridades

---

## Introduction

Several authors agree on the magnitude of testing in software quality (Frankl *et al.*, 1998; Ganesan *et al.*, 2000; Everett & McLeod, 2007; Williams, 2010; Myers *et al.*, 2011). Some studies refer to the importance of estimating the effort associated with testing to decide manual or automated execution in each case (Singh & Misra, 2008), as well as test automation in specific environments (Yuan & Xie, 2006; Xie & Memon, 2007; Bouquet *et al.*, 2008; Masood *et al.*, 2009; Galler *et al.*, 2010; Ko & Myers, 2010).

There are a number of proposals that approach the subject of reducing test cases (Heimdahl & George, 2004; Polo *et al.*, 2007). These use algorithms where it is vital to have the time needed for testing stage, which is difficult to estimate sometimes. New processes and methodologies have emerged for designing testing and process control (Gutiérrez *et al.*, 2007; Dias & Horta, 2008; Naslavsky *et al.*, 2008; Nguyen *et al.*, 2010; Myers *et al.*, 2011).

As a starting point of this process are defined Test Cases, and it is necessary to determine in which order they will be executed. In this context it is essential to address the topic of prioritization of test cases, with mechanisms to decide what should be executed first.

There is a set of proposals to prioritize test cases (Elbaum *et al.*, 2000; Kim & Porter, 2002; Elbaum *et al.*, 2004; Jeffrey & Gupta, 2006; Fraser & Wotawa, 2007; Polo *et al.*, 2007), these are related mainly to the analysis of source statements, loops and other elements of the application code, so it is necessary to have reached the stage of implementation for these methods. On the other hand, functional test cases are closely related with system requirements to be tested.

The proposal presented in this paper is framed precisely in black box testing and part of the definition of software project requirements. The main objective is to propose a function to evaluate the priority of a test case with respect to another, starting from the description of the requirements. The following sections detail the indicators defined in the evaluation function, the procedure for determining priorities and two case studies where the function is applied.

## Methods

A function is been defined to obtain a value associated with the priority of each test case. Seven indicators are considered,  $I_i$  for a sub-domain  $1 \leq i \leq 7$ , divided into two groups. The first group includes a set of indicators related to project requirements. Associated to the project requirements is the Test Case to be evaluated.

The indicators of the first group are: the risk of the requirement ( $I_1$ ), requirement stability ( $I_2$ ), priority requirement within the project ( $I_3$ ) and the relevance of the requirement for the client ( $I_4$ ). The second group includes a set of

indicators that define the characteristics of the test case, they are: the significance of the test procedure ( $I_5$ ), the relevance of the input values ( $I_6$ ) and the relevance of the scenarios that cover the test case ( $I_7$ ).

The evaluation function to obtain the priority of the test case value  $j$  is  $f \rightarrow (0, 1)$ . Its first term includes the indicators of the first group and the other terms include indicators of the second group.

$$f\left(\left(w_i, \delta_i(x)\right)_{i=1}^4\right)_{j=1}^n = w_1 f_r\left(\left(w_k, \theta_k(y)\right)_{k=1}^4\right) + \sum_{i=2}^4 \frac{w_i \cdot \delta_i(x)}{\text{Max}(\delta)} \quad (1)$$

Where,

$w_1$ , is a value between 0 and 1 corresponding to the weight of the indicators related to the associated requirement, meaning the indicators of the first group.

$f_r\left(\left(w_k, \theta_k(y)\right)_{k=1}^4\right)$ , is the function through which the indicators related to the requirement  $y$  associated with the test case  $x$  are evaluated.

$w_i$ , for  $2 \leq i \leq 4$ , signifies the weight associated with the indicators  $I_5, I_6$  e  $I_7$  respectively.

$\delta_i(x)$ , is the function to get the value of the indicator  $i$  for the test case  $X$ .

The indicators associated directly with the requirement are evaluated using  $f_r \rightarrow (0, 1)$ .

$$f_r\left(\left(w_k, \theta_k(y)\right)_{k=1}^4\right) = \sum_{k=1}^4 \frac{w_i \cdot \theta_i(y)}{\text{Max}(\theta)} \quad (2)$$

Where,

$w_i$ , for  $1 \leq i \leq 4$ , is a value between 0 and 1 corresponding to the weight associated with the indicators  $I_1, I_2, I_3$  e  $I_4$  respectively.

$\theta_i(y)$ , is the function to get the value of the indicator  $i$  for the requirement  $y$  associated to the test case  $X$ .

In order to determine the priority of a test case this function has to be inserted in a method for obtaining each of the values of the indicators, analyzing the information stored in the project.

## Results and Discussion

The validation of the function was implemented from two perspectives, the first in which a group of experts was surveyed and a second where the function was applied to prioritize the test cases into two concrete projects, as case studies.

For validation through expert judgment nine specialists were selected, due to their expertise in these areas of research. Two rounds of surveys were applied with the intention of identify indicators and their significance in the process of prioritizing test cases.

The first round of a survey consisted of selecting from 10 factors found in the literature, those which could have a major significance on the priority of the test cases therefore, should be considered in the function for calculating priority. From the results of this analysis, a group of indicators that had agreement among experts were selected. The experts refined and added other indicators that were not in the original proposal. The second round permitted for the validation of the result. Seven indicators were included in the proposed function as a result of the application of this method.

The application of the two case studies involved the project team who executed the procedure shown in Figure 1. Involved in this process are the test analyst and the tester. This latter will obtain as a result the set of test cases, ordered by priority. However to run these activities it was necessary to fulfill a prerequisites group in each of the pilot projects, they are:

1. The designer of test cases prepared and delivered a list of test cases and identified equivalence classes for the attributes of the entities of the system.
2. The project architect was responsible to guarantee that project requirements were properly prioritized. In the case of two pilot projects fuzzy values were used. To indicators  $I_1, I_2, I_3$  e  $I_4$  the values assigned were "High", "Medium" or "Low", which for priority calculation function represent 1, 0.5 y 0 respectively. On the other hand, to indicators  $I_5, I_6$  e  $I_7$  numerical values were assigned.

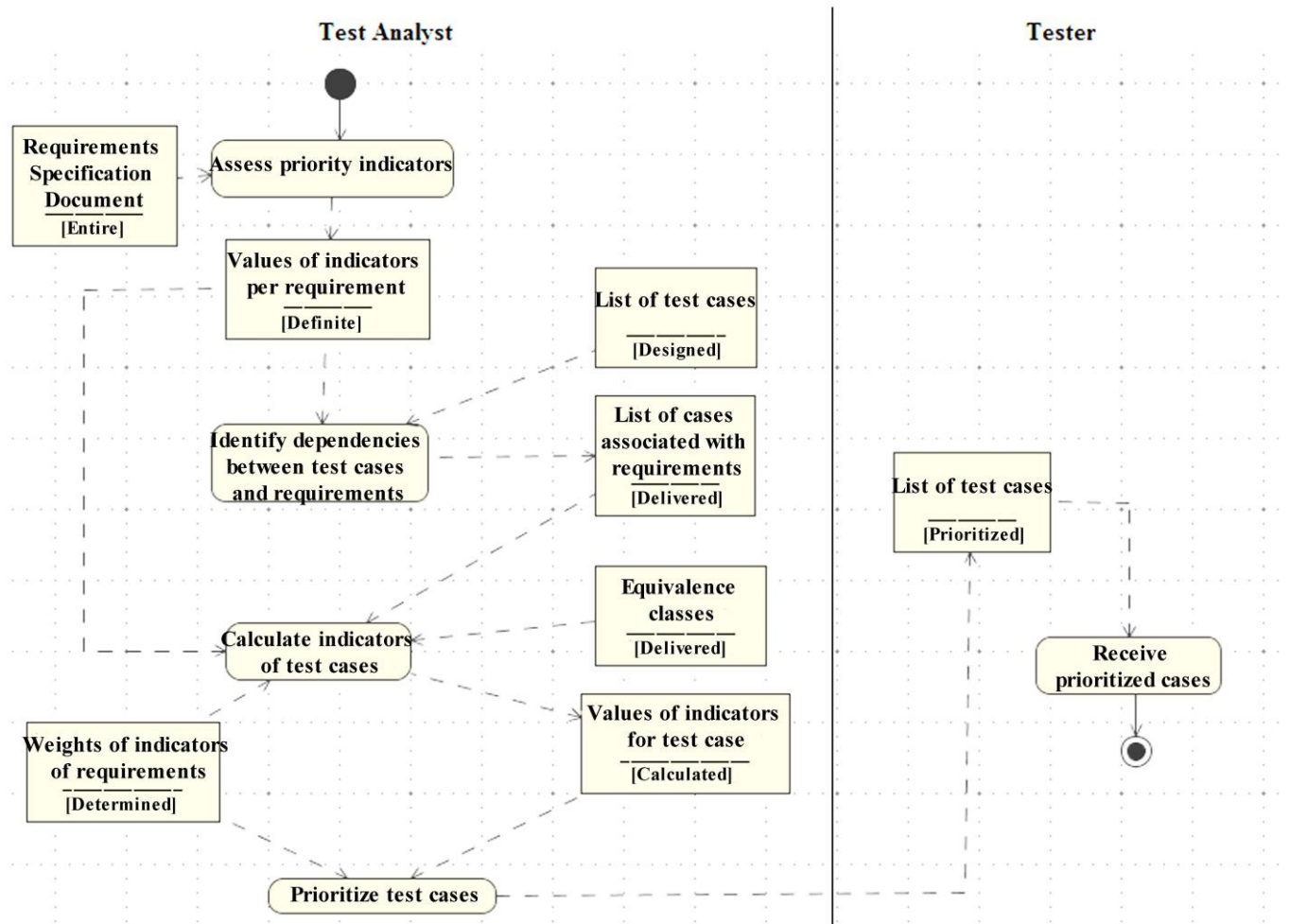


Figure 1. Activity diagram describing the priority assignment process

The two pilot projects have between 20 and 45 requirements. The first (P1) is a project that was in development. As a result of the execution and enforcement of the priority assignment function four blocks of ordered test cases were obtained.

The second project is a new version of an application that is currently delivered to the users. The new version consists of 16 modules or subsystems. The study was performed for one of the modules of the previous version in order to compare the results obtained by using the priority function and the prioritization process that was taken before, during its development.

Fifty-two test cases were obtained in the process of generating test cases. These test cases were sorted into five groups. From detailed analysis of the test cases in the groups, significant differences were detected with the priorities set out by the project team without using priority function. The results are shown in Figure 2.

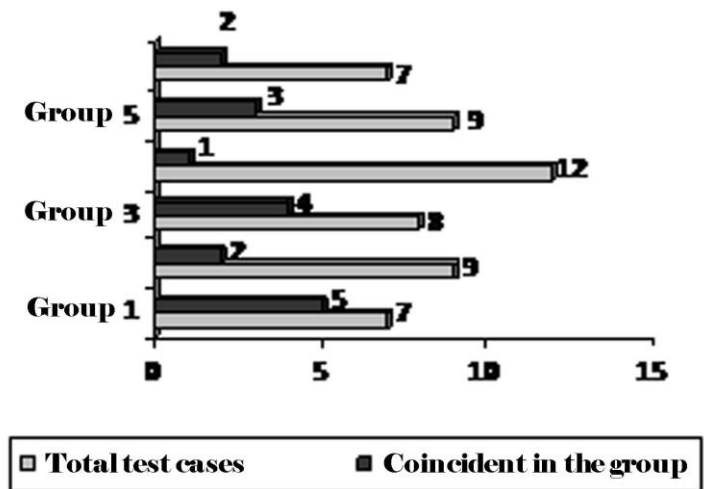


Figure 2. Location of test cases into priority groups as a result of applying the function  $f \rightarrow (0, 1)$

Based on these results a survey was applied to the team to determine whether or not the differences represented an improvement in the prioritization, and how these differences were assessed.

Respondents agree that the test cases that were prioritized by the function, in its initial estimation had been relegated because the following priority criteria were not taken into account: requirement stability, the relevance of the input values and the relevance of the scenarios that cover the test case. Figure 3 shows the distribution of the 28 test cases that are in a different group because originally they were not considered indicators or assigned a different weight.

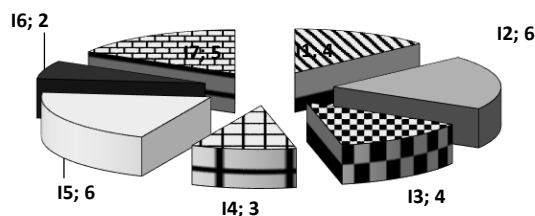


Figure 3. Influence of the indicator in the change of priority of test cases

## Conclusions

During the development of this research a function for calculating priority of functional test cases of software was defined. With the purpose of reach this achievement a set of indicators that can be considered from the early stages of the project have been identified.

The priority function is defined in such a way that allows for the addition of new indicators. To add a new indicator, a new term should be included and the weights for each member of the function, adjusted.

The pilot test on two real projects has provided an initial approach to the prioritization of test cases, which can be enhanced through their application in other projects.

## References

- BOUQUET, F.; C. GRANDPIERRE, *et al.* 2008 A Test Generation Solution to Automate Software Testing. En: International workshop on Automation of software test. AST '08 Proceedings of the 3rd international workshop on Automation of software test. Leipzig, Germany, ACM. pp. 45-48.
- DIAS, A. C. and G. HORTA. 2008. Supporting the Selection of Model-based Testing. Approaches for Software Projects. En: International Workshop on Automation of Software Test. Proceedings of the 3rd International Workshop on Automation of Software Test, AST 2008. Leipzig, Germany, ACM, 2008. pp. 21-24.
- ELBAUM, S.; A. G. MALISHEVSKY, *et al.* 2000 Prioritizing Test Cases for Regression Testing. En: International symposium on Software testing and analysis. ISSTA '00 Proceedings of the 2000 ACM SIGSOFT international symposium on Software testing and analysis Portland, Oregon, USA, ACM, 2000. pp. 102-112.
- Elbaum, S.; G. Rothermel, *et al.* 2004 Selecting a Cost-Effective Test Case Prioritization Technique. Software Quality Journal, 2004, 12(3): pp. 185-210.
- EVERETT, G. D. and R. MCLEOD. 2007 Software Testing: Testing across the entire software development life cycle. New Jersey, USA, Wiley-IEEE Computer Society Press, 2007. 280 p.
- Frankl, P. G.; R. G. Hamlet, *et al.* 1998 Evaluating Testing Methods by Delivered Reliability. IEEE Transactions on Software Engineering, 1998, 24(8): pp. 586-601.

FRASER, G. and F. WOTAWA. 2007 Test-Case Prioritization with Model-Checkers. En: Conference on IASTED International Multi-Conference: Software Engineering SE'07 Proceedings of the 25th conference on IASTED International Multi-Conference: Software Engineering Innsbruck, Austria, ACM, 2007. pp. 267-272.

GALLER, S. J.; A. MALLER, *et al.* 2010 Automatically Extracting Mock Object Behavior from Design by Contract Specification for Test Data Generation. En: Workshop on Automation of Software Test. AST '10 Proceedings of the 5th Workshop on Automation of Software Test. Cape Town, South Africa, ACM, 2010. pp. 43-50.

Ganesan, K.; T. Khoshgoftaar, *et al.* 2000, Case-Based Software Quality Prediction. International Journal of Software Engineering and Knowledge Engineering, 2000, 10(2): pp. 139-152.

GUTIÉRREZ, J. J.; M. J. ESCALONA, *et al.* 2007 Generación automática de objetivos de prueba a partir de casos de uso mediante partición de categorías y variables operacionales. En: Jornada de Ingeniería del Software y Bases de Datos. XII Jornadas de Ingeniería del Software y Bases de Datos. Zaragoza, España, 2007. pp. 105-114.

HEIMDAHL, M. P. E. and D. GEORGE. 2004 Test-Suite Reduction for Model Based Tests: Effects on Test Quality and Implications for Testing. En: IEEE international conference on Automated software engineering ASE '04 Proceedings of the 19th IEEE international conference on Automated software engineering Linz, Austria, ACM, 2004. pp. 176-185.

JEFFREY, D. and N. GUPTA. 2006 Test Case Prioritization Using Relevant Slices. En: International Computer Software and Applications Conference. 30th Annual International Computer Software and Applications Conference (COMPSAC'06) Chicago, USA, IEEE, 2006. pp. 411-420.

KIM, J.-M. and A. PORTER. 2002 A History-Based Test Prioritization Technique for Regression Testing in Resource Constrained Enviroments. En: International Conference on Software Engineering ICSE '02 Proceedings of the 24th International Conference on Software Engineering Florida, USA, ACM, 2002. pp. 119-129.



- Ko, A. J. and B. A. 2010 Myers Extracting and Answering Why and Why Not Questions about Java Program Output. *ACM Transactions on Software Engineering and Methodology*, 2010, 20(2): pp. 1-36.
- Masood, A.; R. Bhatti, *et al.* 2009 Scalable and Effective Test Generation for Role-Based Access Control Systems. *IEEE Transactions on Software Engineering*, 2009, 35(5): pp. 654-668.
- MYERS, G. J.; T. BADGETT, *et al.* 2011 The art of software testing. 3a. edición. New Jersey, JohnWiley & Sons, 2011. p.
- NASLAVSKY, L.; H. ZIV, *et al.* 2008 Using Model Transformation to Support Model-Based Test Coverage Measurement. En: International workshop on Automation of software test AST '08 Proceedings of the 3rd international workshop on Automation of software test Leipzig, Germany, ACM, 2008. pp. 1-6.
- NGUYEN, D. H.; P. STROOPER, *et al.* 2010 Model-Based Testing of Multiple GUI Variants Using the GUI Test Generator. En: Workshop on Automation of Software Test AST '10 Proceedings of the 5th Workshop on Automation of Software Test Cape Town, South Africa, ACM, 2010. pp. 24-30.
- POLO, M.; I. GARCÍA, *et al.* Priorización de casos de prueba mediante mutación. *Actas de Talleres de Ingeniería del Software y Bases de Datos*, 2007, 1(4): pp. 11-16.
- SINGH, D. AND A. K. MISRA 2008 Software Test Effort Estimation. *ACM SIGSOFT Software Engineering Notes*, 2008, 33(3): pp. 80-85.
- WILLIAMS, N. Abstract Path Testing with PathCrawler. En: Workshop on Automation of Software Test AST '10 Proceedings of the 5th Workshop on Automation of Software Test Cape Town, South Africa, ACM, 2010. pp. 35-42.
- Xie, Q. and A. M. Memon 2007 Designing and Comparing Automated Test Oracles for GUI-Based Software Applications. *ACM Transactions Software Engineering and Methodology*, 2007, 16(1): pp. 41-77.
- YUAN, H. and T. XIE. 2006 Substra: A Framework for Automatic Generation of Integration Tests. En: International workshop on Automation of software test AST '06 Proceedings of the 2006 international workshop on Automation of software test Shangai, China, ACM, 2006. pp. 64-70.