

Multicriteria Model for Selection of Automated System Tests

Márcia Sampaio^{1,2}, Paula Donegan², Ana Karoline Castro², Plácido Rogério Pinheiro², Adriano de Carvalho¹, and Arnaldo Dias Belchior²

1 Instituto Atlântico

Rua Chico Lemos 946, 60822-780 Fortaleza – CE, Brazil
{marcia,adriano}@atlantico.com.br

WWW home page: <http://www.atlantico.com.br>

2 Mestrado em Informática Aplicada, Universidade de Fortaleza

Av. Washington Soares 1321, 60811-341 Fortaleza – CE, Brazil
{pauladonegan,akcastro}@gmail.com,

{belchior,placido}@unifor.br

WWW home page: <http://www.unifor.br/mia>

Abstract. Software Test Automation is a renowned way to enhance the test process, but it is not always the best approach for the project, despite that a crucial matter when applying software tests is to decide the viability of automating them. This decision is not always easy to take, because it involves persons exposing their point of view, which can bring a conflict with others opinion. In this context, this work aims to implement a decision aid model based on multicriteria to help test analysts and stakeholders in the selection of use cases for automation, according to organizational and projects' realities.

1 Introduction

Automated Software Testing is an activity that seems to have obvious benefits: tests can be executed quicker, are consistent and can be repeated various times without adding costs, however is not a trivial activity and requires good planning. It is necessary to keep in mind that automation requires good planning for the entire testing process, so that problems can be avoided. For this reason, the decision about which use cases (UCs) to have their tests automated is of great importance for a project's success.

Please use the following format when citing this chapter:

Sampaio, M., Donegan, P., Castro, A., K., Rogério Pinheiro, P., de Carvalho, A., Dias Belchior, A., 2006, in International Federation for Information Processing, Volume 205, Research and Practical Issues of Enterprise Information Systems, eds. Tjoa, A.M., Xu, L., Chaudhry, S., (Boston:Springer), pp.777-782.

Tests execution may also be hybrid. The choice depends on a detailed analysis of the tests' complexity. This decision is complicated to be taken, because of the diversity of questions to be analysed and their impact on the project; however are more efficient considering time [1].

Therefore, this paper has the objective of showing a case study of the application of a multicriteria methodology to assist with the selection of use cases to be automated, with the intention of improving efforts spent. The methodologies of Multicriteria Decision Aiding (MCDA) have as strongest presupposition the fact of recognizing the importance of subjectivity from a decision-maker, person with power and responsibility of assuming consequences [4].

2 Multicriteria Model for Selection of Automated Tests

Every project is unique, with its own characteristics and specific requirements, involving a certain degree of uncertainty related to the decision of use cases to have their system tests automated. To know the project's life cycle, the people involved with tests, organizational influences, use cases' stability and types of tests necessary are all strategic factors to minimize those uncertainties.

Practical experience has shown that only an average of 60% of the project should be automated [2]. Therefore, it is extremely important to choose the correct use cases to automate and to facilitate the decision making of use cases to be automated; a multicriteria model was formulated, optimizing the process based on objective and subjective criteria to apply on diverse organizational projects. This model has a sequence of generic steps, distributed in phases of the multicriteria methodology that must be executed to make the decision.

2.1 Structure Phase

This phase aims the construction of a formal model, capable of being accepted by all the actors as a structure of representation and organization of the entire group of evaluation criteria, consisting of a specific system's analysis and making potential alternatives of decision explicit.

A criterion (c) is a tool to evaluate tests susceptible to automation in terms of a certain *point of view* (PV) or concern of the actors responsible for the analysis. In the phase where the project tests are planned, an identification of those responsible for extracting criteria that have influence in test automation is needed. The quantity of criteria (n) may vary in each project.

In accordance with [2, 3, and 6], the organization's main criteria are chosen and organized in a table, with reasons for each criterion and the question that should be answered to analyse the criterion.

The actors must be selected from stakeholders related directly or indirectly by the decision process, revealing their point of view (PV). Possible actors are project's

manager, coordinator, test analyst, requirements analyst, developer and tester. The quantity of actors selected may change in accordance to the project's characteristics.

Actors have to answer questions related to criteria for each UC, informing if they attend ($PV(c) = 1$) or not ($PV(c) = 0$) to every criterion. If actors are not in condition of answering to a question, the question is considered as non-applicable ($PV(c) = null$), therefore:

$$PV_x(a, c) \in \{null, 0, 1\}, \text{ where } x = UC$$

A questionnaire is elaborated to obtain weights of actors (*weight of actor – WA*), embracing actor's experience in activities related to tests; roles performed; participation in projects; training and participation in test conferences. Each item has a value. With the measurement of all items, the actor's weight is obtained, which is normalized by a discrete variable between *zero* and *six* (considering MACBETH classification). The value varies for actors and criteria, but is equal for UCs.

$$WA(a, c) \in \{0, 1, 2, 3, \dots, 6\}$$

Every *criterion (c)* receives a *priority* (classification), in accordance to the degree of relevance among the previously established criteria, in a decreasing order of importance. Every actor (*a*) must classify the criteria taking into consideration their relevance for the project's test process, and not for a specific UC.

$Priority(a, c_i) \in \{1, \dots, i, \dots, n\} \mid \forall c_i, c_j \mid i \neq j \Rightarrow Priority(a, c_i) \neq Priority(a, c_j)$ where $n = |criteria|$; $i, j > 0$; $Priority(a, c) =$ priority of criterion *c* for actor *a*

The following formula is obtained for every UC *x*, every actor *a* and criterion *c*:

$$WA(a, c) \in \{0, 1, 2, 3, \dots, 6\} \text{ and } PV_x(a, c) \in \{0, 1\}$$

$$Priority(a, c_i) \in \{1, \dots, i, \dots, n\} \mid \forall c_i, c_j \mid i \neq j \Rightarrow Priority(a, c_i) \neq Priority(a, c_j)$$

The three sets of values must be equalized (on the same base – base 1) after they are informed so that a correct evaluation is possible without favoring a value to the detriment of another. Therefore:

$$[WA(a, c)]_i \in \{0, 0.17, 0.33, 0.50, 0.67, 0.83, 1\} \text{ and } [PV_x(a, c)]_i \in \{null, 0, 1\}$$

$$[Priority(a, c)]_i \in \{1/n, 2/n, \dots, n/n\}$$

Afterwards, for each actor, the three variables per criterion are multiplied, obtaining a specific punctuation, the score *S*.

$$S_x(a, c) = \{ [WA(a, c)]_i \times [PV_x(a, c)]_i \times [Priority(a, c)]_i \}$$

A score *S* for every criterion of every UC is given to each actor, and ordered under the form $S_{x,j}(a, c)$ where $S_{x,j}$ represents the *j*th score of the criterion for UC *x* and actor *a* ordered with component $j = 1, \dots, m$, where *m* represents the number of actors. Then, the medium (*Me*) of the scores is calculated, as follows:

$$Me(x, c) = \begin{cases} [S_{x, j/2} + S_{x, (j/2+1)}] / 2, & \text{if } m \text{ is even,} \\ [S_{x, (j+1)/2}], & \text{otherwise} \end{cases}$$

The values of $S = null$ are not considered to calculate the medium. The medium's value for the UC represents the final score of each criterion per UC, considering all actors involved in the decision process. These mediums are the base of classification of UCs to automate.

2.2 Evaluation Phase

This phase constructs matrixes of judgments and acquires scales of cardinal value for every criterion. The tasks are implemented with the MACBETH methodology [5].

To use the model, equivalence was made with results and the MACBETH classification. The equalization of *point of view*, *weight of actor* and priority (shown in previous section) results in values of the general score between 0 and 1. Medium's values are divided into six intervals related to MACBETH's classification. With the obtained categories, the MACBETH's modeling is made showing the attraction difference among UCs.

2.3 Recommendation Phase

This phase consists on the analysis of results generated by MACBETH from scales of values generated in the matrixes of judgments, which is composed of various actions that must be analysed according to the decision-maker evaluation.

The results obtained during the evaluation phase generate reports with graphics, and analysis of UCs classified in a ranking so that they may be given priorities for automation. Given this classification, the stakeholders will establish, according to specific capacities of the project and obtained results, the UCs to be automated, respecting order obtained with evaluation.

3 An Application of the Multicriteria Model: A Case Study

In the Structure Phase (step "*Identify Criteria*"), the test analyst and the coordinator made an analysis of which criteria to analyse to select UCs for automation.

Afterwards the following actors were chosen: project's tester; and representative of the test group. Afterwards they answered questionnaires for all criteria and the decision-maker obtained the weights for each actor according to the questionnaires. Each actor classified the selected criteria depending on their own perception of their priorities when on step "*Attribute Priorities to Criteria*".

A partial evaluation of the score (S) is made for every criterion, multiplying *point of view* of each actor for each UC, *weight of actor* and criterion's *priority*. The values given by the actor were equalized to guarantee an equal evaluation. In step "*Calculate General Scores*" the medium is calculated to obtain the final score of the UC for each criterion, obtaining the partial evaluation of each actor.

In the Evaluation Phase, a matrix of judgment was made for every criterion using MACBETH, informing attraction differences between UCs. Then scales of cardinal value were obtained and a quantitative analysis of criteria values could be made. The one with highest weight would be "*Criticality*" and the lowest "*Availability of Time/Resources*".

In the Recommendation Phase, a classification for automation was obtained for every criterion. See Figure 1 to visualize the general classification obtained. The highest priority automation UC would be “*Adjust Schedule*” (UC14) and the lowest “*Download File*” (UC18). Given the obtained results, the actors arranged a meeting and selected the four most recommended (UC14, UC08, UC20 and UC19).

This analysis provided a proper planning of UCs to automate, in accordance to the project’s structure available. Besides, the resultant values, after applying the model, provides a visibility of the criteria’s level related to each UC, making it possible to prevent problems.

Actions scores							
Actions	Overall	CRITIC	STABIL	REPEAT	PERFOR	USABIL	TIME
[all high]	100.00	100.00	100.00	100.00	100.00	100.00	100.00
UC14	86.65	94.74	95.65	76.92	73.33	93.75	95.00
UC8	74.36	89.47	60.87	92.31	46.67	87.50	90.00
UC20	73.13	84.21	47.83	69.23	93.33	62.50	85.00
UC19	66.49	68.42	86.96	38.46	86.67	37.50	70.00
UC7	53.19	52.63	73.91	84.62	20.00	31.25	55.00
UC18	5.97	5.26	4.35	7.69	6.67	6.25	5.00
[all low]	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Scaling constants:		0.24	0.21	0.19	0.22	0.13	0.02

Fig. 1. Use Cases’ general score

4 Conclusions

A multicriteria model used for software system tests was presented to select UCs for automation. The application of this model has shown satisfactory, since the selection is not random anymore, depending on diverse subjective opinions and frequently costly. Furthermore, some problems, such as inadequate use of available resources and unnecessary automation, have been minimized, showing efficiency when using the model. Therefore, any software company working, or willing to work, with test automation, can use this model.

References

1. P. Donegan, L. Bandeira, A. Matos, P. Ferreira, C. Maia, and G. Pires, *Applyability of Automated Functional Tests* (In portuguese: Anais SBQS - Simpósio Brasileiro de Qualidade de Software, 2005).
2. D.J. Mosley, A.B. Posey, and A. Bruce, *Just Enough Software Test Automation* (Just Enough Series, Yourdon Press, 2003).
3. K. Cem, J. Bach, and B. Pettichord, *Lessons Learned in Software Testing: A Context-Driven Approach* (John Wiley & Sons, 2002).

4. G.G.S.C. Souza Um Modelo Multicritério para Produção de um Jornal, Masters Dissertation in Applied Informatics. (University of Fortaleza, Fortaleza, 2003).
5. C.A. Bana e Costa and J.C. Vansnick, MACBETH: A Theoretical Framework for Measuring Attractiveness, XIth International Conference on MCDA (Aug. 1994).
6. E. Dustin, *Effective Software Testing: 50 Specific Ways to Improve Your Testing* (Addison-Wesley, 2002).