Applying a multicriteria model for selection of test use cases: a use of experience


Graduate in Applied Computer Science (MIA), University of Fortaleza (UNIFOR), Av. Washington Soares, 1321 – Fortaleza, Ce, Brazil
E-mail: andrearsp@gmail.com
E-mail: marciagg@gmail.com
E-mail: adriano@unifor.br
E-mail: maikol@unifor.br
E-mail: placido@unifor.br
*Corresponding authors

Abstract: Nowadays, software organisations need to develop excellent and reliable products. This scenario has helped to increase the relevance of quality assurance activities, especially the testing discipline. However, sometimes the time and resources are limited and not all tests can be executed, demanding organisations to decide what use cases should be tested, to guarantee the predictability of the project’s time and budget. The multiple criteria methodologies support decisions, considering many factors, not only professional experience. This paper presents a multiple criteria model to support the decision of selecting the use cases that should be tested.

Keywords: tests; decision and analysis resolution; multiple criteria decision analysis.


Biographical notes: Andreia Rodrigues received her degree in Computer Sciences from University of Fortaleza, CE, Brazil in 2009. She is currently pursuing an MSc in Applied Computer Sciences at the University of Fortaleza. She has experience in computer sciences, especially in software engineering.

Plácido Pinheiro received his degree in Mathematics from Federal University of Ceará, Brazil in 1979. He is an Electrician Engineer from the University of Fortaleza, CE, Brazil (1983), with a Master degree in Mathematics at the Federal University of Ceará (1990) and PhD in Systems Engineering and Computation at the Federal University of Rio de Janeiro, Brazil (1995). He is currently a Professor at the University of Fortaleza and a Researcher of the National Counsel of Technological and Scientific Development (CNPq). He
Applying a multicriteria model for selection of test use cases

has experience in the modelling of industrial processes using mathematical programming and multicriteria. His training allows publishing in the area of mathematics, with emphasis on discrete mathematics and combinatorics, working primarily in mathematical programming and multicriteria.

Maikol Rodrigues received his degree in Computer Science from the State University of Ceará, Brazil in 1999. He received his Master degree in Computer Science from the University of Campinas (Campinas, State of São Paulo, Brazil). He is currently a Professor at the University of Fortaleza and Government Employee from DATAPREV. He has experience in the modelling of industrial processes using mathematical programming.

Adriano Bessa Albuquerque received his degree in Computer Science from Federal University of Ceará, Brazil in 1990, with a Master degree in Computer Science from the University of Fortaleza, CE, Brazil (2001) and PhD in Systems Engineering and Computation at the Federal University of Rio de Janeiro, Brazil (2008). He is currently a Professor at the University of Fortaleza. He has experience in software quality, mainly in software process improvement. His training allows publishing in the area of software quality, with emphasis on software process improvement, software metrics, software experimentation and knowledge management supporting software process.

Márcia Golçalves received her degree in Computer Sciences from the University of Fortaleza. She has a PMP – Project Management Professional (2005) certification. She is an Internal Auditor of quality in ISO 9002:2000 norm and a CMMI Appraiser in SCAMPI method. She has participated in the team responsible for the implantation of SW-CMMI level 3 and ISO 9001:2000, defining processes and software development methodologies based on rational unified process (RUP) and PMBoK. She acts as a member of organisational groups, such as processes engineering, project management, quality assurance, measurement and analysis, training and configuration. She coordinates the project office at the company where she works.

1 Introduction

Software testing is one of the disciplines that have the capability of providing assistance to improve the quality of an organisation’s products, because its goal is to evaluate how the product meets the clients’ specified requirements through a controlled execution of the software.

As reported by Presmann (2005) the cost to prevent an error on the maintenance phase is 60–100 times greater than the cost to correct it during the development phase. Despite the relevance of the quality assurance, its cost is high on the software development (Barreto, 2006). According to Wagner and Seifert (2005) 50% of the costs of the software development are related to software testing. In some cases, when there is not enough time and resources to guarantee complete test coverage, software organisations should reduce their scope.

This work has as main objective the implementation of an approach based on a methodology that provides a structured support to multicriteria decisions. This methodology assists the process of deciding which use cases should be selected to be tested by removing subjective decisions in a structured way. This decision is complicated
to be taken, because of the diversity of questions to be analysed and their impact on the project.

The multicriteria methodology helps to generate knowledge about the decision context and thus increases confidence of those making decisions on the outcome results (Evangelou et al., 2005). Other applications which have multicriteria involving software engineering were defined in Mendes et al. (2008, 2009) and Oliveira (2007). This research was carried out to define and execute an approach to support software organisations at selecting use cases that should be tested. The approach was based on the model defined in Gonçalves et al. (2006).

2 Software test

Testing consists of verifying, dynamically, the software behaviour to determine if it adheres to the specifications and executes correctly in the projected environment. As the main objective is to find software failures, this activity has a destructive nature (Dias, 2007).

Software testing focuses on the product’s quality and cannot be considered elementary, because many factors can compromise the success of this activity’s execution:

1. time limitations
2. resource limitations
3. lack of skilled professionals
4. insufficient knowledge of test procedures and techniques and adequate test planning;
5. subjectivity of requirement and test specifications
6. increase of the systems’ complexity.

Moreover, difficulties related to the test activity are also due to the great variety of combinations of input and output data and the large quantity of paths that make it infeasible to execute all possible test cases (Bandeira, 2005).

Test case is the definition of a specific condition to be tested. Its structure is based on input values, restrictions to its execution and expected results or behaviours (Craig and Jaskiel, 2002). Test cases are designed based on use cases, which represent interactions, that users external systems and hardware have with the software aiming to achieve an objective. The amount of test cases executed is one of the main factors which may influence the cost of testing (Myers, 1979). Therefore it is fundamental to define a test scope considering acceptance criteria and business risks.

3 Multiple criteria decision analysis (MCDA)

Decision-making should be exploited when deciding to execute or not some activities or to perform them applying some methods (Gonçalves et al., 2006). The multiple criteria decision analysis proposes to reduce the subjectivity on the decision-making. Nevertheless, the subjectivity will be always present, because the mathematically
analysed items are always results of human beings’ opinions. These multiple criteria models allow the decision-maker to analyse possible consequences of each action to obtain the best understanding of the relationships between actions and their goals (Flament, 1999). Objective criteria should be considered, as well as subjective criteria, even being generally disperse and diffuse in a decision context, but they are extremely important to assess actions.

3.1 MACBETH approach

The MACBETH methodology contemplates the understanding and learning of the problem content and is divided into three phases: structure, evaluation and recommendation, as we can see in Figure 1 (Bana e Costa et al., 2001).

The structure phase focuses on constructing a formal model, capable of being accepted by actors as a structure to represent and organise an entire group of evaluation criteria. It consists of analysing a specific system and making potential alternatives of decision explicit. The evaluation phase produces matrixes of judgments and provides scales of cardinal value for every criterion. The tasks are implemented with the MACBETH methodology. In the recommendation phase, the results generated by MACBETH are analysed using scales of values generated by the matrixes of judgments, which are composed of various actions that must be examined according to the decision-maker evaluation (Gonçalves et al., 2006). Structuring activities include:

1. definition of a family of fundamental points of view (FPV)
2. construction of descriptors
3. estimation of the impacts profiles of each action (Bana e Costa et al., 2001).

Figure 1 Phases of the multiple criteria decision aid (see online version for colours)

The construction of descriptors comprehends three stages:

1. description of each descriptor for each fundamental point of view
2. access impacts for each fundamental point of view
3. analysis of impacts according to the fundamental points of view (Pinheiro et al., 2008b).
The descriptors are desired to:
1. turn operational the analysis of impacts of the options in a FPV
2. describe the impacts with respect to FPVs
3. improve the structure of the evaluation model
4. verify the ordinal independence of the corresponding FPVs.

The FPV becomes operational if there’s a set of impact levels associated with it, defined by $N_j$, which should be sorted in descending order by the decision makers. Thus, they constitute a range of local preference, limited by the higher level $N_j^*$ that has more attractiveness and the lower level $N_j$, of less attractiveness, should meet the following pre-ordering condition:

$$N_j^* > \ldots > N_{k+1,j} > N_{k,j} > N_{k-1,j} > \ldots > N_{s,j}.$$ 

The main difference of MACBETH (Meskens, 1999) to other multiple criteria methods is that it requires only qualitative judgments of the difference between the elements’ attractiveness aiming to assign values to the options for each criterion and to weigh up the criteria. MACBETH applies the concept of attractiveness to measure the potential actions’ values. Therefore, when the decision-maker is demanded to judge the value of a potential action on a specific situation, he should think about his attraction to that action (Cunha, 2008). HIVIEW (Pinheiro et al., 2008a) is a tool to evaluate models defined using multiple criteria methodologies with an aggregation function, like MACBETH. By using HIVIEW, the decision-maker defines, analyses, evaluates and justifies his preferences, considering existent alternatives. This software facilitates a lot the analysis of complex problems, supporting the elaboration of the problem’s structure, specifying the criteria used to choose alternatives and to assign weights to the criteria.

The alternatives are evaluated by comparing these criteria and a preference value is assigned to each alternative’s criteria. Additionally, it’s possible to change judgments and to compare the obtained answers graphically, providing information to the decision-maker for re-evaluation. If necessary the decision can be rectified.

4 Model to select what use cases should be tested and the experience of use

The proposed model is based on the model presented on Gonçalves et al. (2006) and is composed of generic steps, grouped by the phases of the multiple criteria decision aid (MCDA). These steps are described in Table 1.

4.1 Experience of use

The organisation where the experience of use was performed is a government institution with 270 professionals allocated on the Information Technology Area, working on projects that support the organisation’s businesses. In 2004, a test team was organised and being external to the projects and responsible for executing systemic tests. Nowadays, the company has high demands of time and resources which make it difficult...
Applying a multicriteria model for selection of test use cases

251
to satisfy the desired testing coverage in all projects. Therefore, many projects reduce their testing scope to assure the delivery schedule. Priorities have to be applied to use cases and accordingly selected to decrease the testing scope. This is quite relative and varies according to the actors involved and to the criteria they judge relevant. The pilot project selected to apply the proposed approach was a project with a schedule restriction, because the organisation agreed on a date with the workers’ union, resulting on a fine if it was delayed.

Table 1  Model’s steps

<table>
<thead>
<tr>
<th>Phase</th>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structure</td>
<td>1</td>
<td>Identify criteria to be used on the use cases evaluation aiming to define their level of priority.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Identify roles that will expose their point of view, considering also their roles on the decision-making process.</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Each actor should assign a weight to all criteria, considering the full test process and not only a specific use case.</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>After the execution of the steps listed above, it is necessary to standardise the three sets of values, putting them on the same base (base 1). The goal is to perform a partial evaluation correctly, without any bias. Then, for each actor the three variables should be multiplied, considering each criterion, thus obtaining a specific score.</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Calculate, for each actor, a score to each criteria of each use case.</td>
</tr>
<tr>
<td>Evaluation</td>
<td>6</td>
<td>Construct the matrix of judgments and obtain the cardinal value scales for each defined criteria.</td>
</tr>
<tr>
<td>Recommendation</td>
<td>7</td>
<td>Prioritise use cases which will be tested, given an analysis of the results obtained from the previous phases.</td>
</tr>
</tbody>
</table>

The project’s life cycle was iterative/incremental and the model was applied on the project’s first iteration. The following use cases were part of the test cycle on which the model was applied:

- UC01_Execute_Sign_In_and_Sign_Out
- UC04_Search_Problems
- UC05_Demand_Benefit_Permission
- UC07_Search_Demands_Benefit_Permission;
- UC08_Approve_Demands_Benefit_Permission
Bellow, we explain how the steps were performed.

4.1.1 Structure

In step ‘identify criteria’, we held a meeting with all actors involved at selecting the criteria, given the criteria listed on Table 2. The selected criteria are those highlighted.

Table 2 List of criteria

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Reason</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability (ISO 9126)</td>
<td>The use case requires a strong interaction with the user and therefore must be usable.</td>
<td>Does the use case require a strong interaction with the user, being essential a high level of usability?</td>
</tr>
<tr>
<td>Security (ISO 9126)</td>
<td>The use case requires a specific control, reducing the access to information.</td>
<td>Does the use case require a specific control to reduce the access to informations?</td>
</tr>
<tr>
<td>Repeatability (Gonçalves et al., 2006)</td>
<td>The use case is being implemented on the first iterations and so will be tested many times.</td>
<td>Will the use case’s test be repeated many times?</td>
</tr>
<tr>
<td>Complexity (Gonçalves et al., 2006)</td>
<td>The use case has a large quantity of associated business rules or depends on complex calculations.</td>
<td>Does the use case have a lot of associated business rules or depend of complex calculations to function adequately?</td>
</tr>
<tr>
<td>User requirements (Dustin, 2002)</td>
<td>The use case needs to be implemented because it will satisfy any contractual or legal demand or the organisation may have financial loss.</td>
<td>Does the use case have to be implemented because it will satisfy any contractual or legal demand or it will prevent the organisation against large financial loss?</td>
</tr>
<tr>
<td>Operational characteristics (Dustin, 2002)</td>
<td>The use case is related to the functions which are frequently used.</td>
<td>Is the use case related to frequently used functions?</td>
</tr>
<tr>
<td>Functionality (ISO 9126)</td>
<td>The specific functions and properties of the product should satisfy the user.</td>
<td>Do the specific functions and properties of the product satisfy the user?</td>
</tr>
<tr>
<td>Reliability (ISO 9126)</td>
<td>The use case requires maturity on fault tolerance aspects.</td>
<td>Does the use case require maturity on fault tolerance aspects?</td>
</tr>
<tr>
<td>Availability of time/resource (Gonçalves et al., 2006)</td>
<td>If a use case requires specific resources (people with technological expertise, software and hardware resources, available budget) so that the test can be performed.</td>
<td>Are there enough and qualified resources to test the use case?</td>
</tr>
</tbody>
</table>
Applying a multicriteria model for selection of test use cases

A criterion (c) is a tool to evaluate tests that are susceptible to automation in terms of a certain point of view (PV) or concern of the actors responsible for the analysis. The quantity of criteria (n) may vary for each project.

In step ‘identify actors and their weights’, the following actors were selected:

- project manager
- tests coordinator
- project’s system analyst
- project’s test analyst.

The actors answered the questions related to each criterion and the questionnaire applied to obtain the actors’ weight, which was defined considering the role performed by the actor on the project (e) his knowledge of the business for which the software was developed.

A questionnaire was elaborated to obtain the weight of each actor (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 had a value and with the measurement of all items, the actor’s weight was obtained. In step ‘assign prioritisation to the criteria’, each actor (a) classified the criteria according to their relevance for the project’s test process. The results are presented on Tables 3 and 4.

### Table 3  Priorisation of the actors per criterion

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Test coordinator</th>
<th>Test analyst</th>
<th>Project manager</th>
<th>System analyst</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Repeatability</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Complexity</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Operational characteristics</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>User requirements</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Security</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

### Table 4  Weights of the actors per criterion

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Test coordinator</th>
<th>Test analyst</th>
<th>Project manager</th>
<th>System analyst</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usability</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Repeatability</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Complexity</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Operational characteristics</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>User requirements</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Security</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>
In step ‘execute a partial evaluation’, we multiplied, for each use case \((x)\), the values of the actors’ point of view \((PV_x)\), the actors’ weights \((WA)\) and the criterion’s level of priorities, as can be seen on the formula below. Besides, an example is presented aiming to facilitate the comprehension.

\[
E_x(a,c) = \left[ PV_x(a,c) \right] \left[ WA(a,c) \right] \left[ \text{Level of prioritisation } (a,c) \right]
\]

\[E_{UC01}(\text{test coordinator, operational characteristics}) = 0.5561\]

The example shows the score obtained for the UC01_Execute_Sign_In_and_Sign_Out use case considering the ‘test coordinator’ actor related to the ‘operational characteristics’ criterion. It is important to highlight that the values applied on the prioritisation and weights were equalised, aiming to guarantee an equal evaluation, as we describe below. The prioritisation value presented in Table 1 for the test coordinator with ‘operational characteristics’, is 5. So, to calculate the partial evaluation, we have:

\[
\left[ PA(a,c) \right] \in \{0, 0.17, 0.33, 0.50, 0.67, 0.83, 1\}
\]

Prioritisation (test coordinator, operational characteristics) = 0.83

Similarly, the actor’s weight has an equivalence scale. In the example, the weight of the test coordinator actor for the ‘operational characteristics’ criterion, presented in Table 1, corresponds to 4. In the Table 5 shows the equivalence of the actor’s weight with the scale already equalised.

<table>
<thead>
<tr>
<th>Value on Table 3</th>
<th>Scale PA ((a, c))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0.17</td>
</tr>
<tr>
<td>2</td>
<td>0.33</td>
</tr>
<tr>
<td>3</td>
<td>0.50</td>
</tr>
<tr>
<td>4</td>
<td>0.67</td>
</tr>
<tr>
<td>5</td>
<td>0.83</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

It is important to emphasise that the obtained values of the actors’ weights and level of priorities were equalised (on the same base – base 1) after they were informed, so that a correct evaluation is possible without benefiting a value at the cost of another.

Finally, in step ‘calculate the general scores of the criteria’, we calculated the median to obtain the final score of the use cases for each criterion. The value of the median, calculated for each use case, will be used as a basis to prioritise. The following equation illustrates the median calculation.

\[
Md(x,c) = \begin{cases} 
\frac{E_{\lfloor \frac{j}{2} \rfloor} + E_{\lfloor \frac{j}{2} + 1 \rfloor}}{2} & \text{if } j \text{ is even} \\
E_{\lfloor \frac{j}{2} + 1 \rfloor} & \text{otherwise.}
\end{cases}
\]
Applying a multicriteria model for selection of test use cases

where

\( x \) represents the user case

\( j \) number of actors

\( c \) criterion.

Therefore, to examine ‘operational characteristics’ for the

UC01_Execute_Sign_In_and_Sign_Out use case, we consider the scores of all the actors, according to the example below:

\[
\begin{align*}
E_{UC01} \text{ (test coordinator, operational characteristics)} &= 0.5561 \\
E_{UC01} \text{ (test analyst, operational characteristics)} &= 0 \\
E_{UC01} \text{ (project manager, operational characteristics)} &= 0.085 \\
E_{UC01} \text{ (system analyst, operational characteristics)} &= 0.1411
\end{align*}
\]

The model indicates that the scores must be ordered in the following way: \( E_{i,j} (a, c) \), where \( E_{i,j} \) represents the \( j \)th score of the use case’s criterion of actor ‘\( a \)’ ordered with \( j = 1, ..., m \) and \( m \) is the quantity of actors. After ordering the criteria’s scores, we have:

\[
\begin{align*}
E_{UC01,1} &= 0 \\
E_{UC01,2} &= 0.085 \\
E_{UC01,3} &= 0.1411 \\
E_{UC01,4} &= 0.5586
\end{align*}
\]

So, the calculation of the median will be:

\[
Md(x,c) = \frac{E_{1,j} + E_{\lceil \frac{j}{2} \rceil}}{2}
\]

\[
Md \text{ (UC01, operational characteristics)} = \frac{0.085 + 0.1411}{2} = 0.11305
\]

The median’s value obtained for the use case will be the base to prioritise the use cases as can be seen in the next section.

4.1.2 Evaluation

The evaluation phase was supported by the HIVIEW software. In step ‘apply scores on MACBETH’, we elaborated the matrixes of judgments in MACBETH for each criterion, having to calculate their subtraction of attractiveness (Table 6). For this, the values of the median of the criterion are subtracted between the use cases (\( x_1 \) to \( x_k \)), considering the module of the obtained value.
Table 6  Judgment matrix for criterion (c)

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>...</th>
<th>$X_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$\text{Me}(x_1, c) - \text{Me}(x_2, c)$</td>
<td>...</td>
<td>$\text{Me}(x_1, c) - \text{Me}(x_k, c)$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>...</td>
<td>$\text{Me}(x_2, c) - \text{Me}(x_k, c)$</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_i$</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

It is important to emphasise that when we calculate the difference of attractiveness between two use cases, we should consider the module of the obtained value. Then, the resulting value must be verified, using the Table 7.

Table 7  Dimensions of the difference of attractiveness

<table>
<thead>
<tr>
<th>Difference of attractiveness</th>
<th>MACBETH category</th>
<th>Results obtained with the model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Very weak</td>
<td>1</td>
<td>0.14</td>
</tr>
<tr>
<td>Weak</td>
<td>2</td>
<td>0.28</td>
</tr>
<tr>
<td>Moderate</td>
<td>3</td>
<td>0.42</td>
</tr>
<tr>
<td>Strong</td>
<td>4</td>
<td>0.57</td>
</tr>
<tr>
<td>Very Strong</td>
<td>5</td>
<td>0.71</td>
</tr>
<tr>
<td>Extreme</td>
<td>6</td>
<td>0.85</td>
</tr>
</tbody>
</table>

The judgment matrix for the ‘operational characteristics’ criterion is presented in Figure 2 as an example. This matrix has the following actions, which will be ordered according to the preference of the decision maker: UC01, UC04, UC07, UC05, UC10 and UC12. After judging all action pairs, scales of cardinal values are obtained. These scales allow analysing the criterion’s values quantitatively. Figure 2 illustrates the matrix of judgments for the ‘operational characteristics’ criterion.

Figure 2  Matrix of judgements to the criterion ‘operational characteristics’ (see online version for colours)
In the Figure 3 depicts the level of importance for each analysed criterion. According to the figure, the criterion with the highest weight was ‘operational characteristics’ and the one with the lowest weight was ‘security’.

**Figure 3** Criteria’s level of prioritisation aid (see online version for colours)

**Figure 4** Criteria operational characteristics level of prioritisation aid (see online version for colours)
4.1.3 Recommendation

The MACBETH provides several types of results, among those, the prioritisation of the use cases by criteria, as it is presented on Figure 4.

The results obtained during the evaluation phase generated reports with graphics and the use cases analysed are ranked. In this step, we identified the use cases with the highest level of priorities. To represent the general classification of all analysed criteria we used the graphic showed in Figure 5.

Figure 5 Selection of use cases (see online version for colours)

![Selection of use cases](image)

Figure 6 Percentage of detected errors to use cases (see online version for colours)

![Percentage of detected errors](image)
4.2 Validating the results of the experience of use

The model was applied at the beginning of the test cycle and we decided to test all use cases, aiming to analyse its adequacy and efficacy and to compare the obtained results. At the end of the test cycle, we calculated the percentage of errors, considering the quantity of use cases tested and the quantity of detected errors for each use case as displayed in Figure 6.

We could observe, considering the level of priorities assigned to the use cases obtained by the model execution, that the model assigned the highest level of priority to those use cases with a higher concentration of errors. However, the level of priority of UC07 was higher than those priorities of UC08, UC10 and UC12, but even so in UC07 were detected fewer errors.

5 Conclusions and further works

This work helped us to conclude that the restriction of time and resources is a real problem on many organisations and that an approach to support the decision to select use cases to be tested is very relevant. Besides, with the experience, we could see that the execution of the proposed model was satisfactory, allowing us to take the decision not using just our subjective experiences.

The application of the MACBETH approach was adequate to model the problem, but, others multicriteria decision aiding (MCDA) methodologies can be used or, if necessary, we can define a combination of them to define the model.

As further works, it will be important to apply the model in other software projects and to define customised questionnaires for other projects’ characteristics.

The proposed approach can be used in other contexts, such as: the selection of processes improvements to be deployed, as it often is not possible to introduce all the improvements at once, given the difficulty of assessing and measuring the effectiveness of implemented improvements. The selection of systemic tests that can be automated is another scenario where we can apply the model. Criteria such as cost, availability of resources and repeatability of tests should be considered.

Acknowledgements

Andreia Rodrigues and Plácido Rogério Pinheiro are thankful to the National Counsel of Technological and Scientific Development (CNPq) for the support received on this project.

References


**Websites**

Hiview Software, Catalyze Ltd., http://www.catalyze.co.uk

M-MACBETH, http://www.m-macbeth.com