Evaluation of Software Product Functional Suitability: A Case Study

MOISÉS RODRÍGUEZ, JESÚS RAMÓN OVIEDO, AND MARIO PIATTINI

The fact that software has entered every sphere of people's lives has sparked greater interest in quality assurance. In recent years, that interest has focused on the software product itself, which in turn has led to the emergence of international quality standards such as ISO/IEC 25000. This is a standard used for establishing the evaluation process and quality models for software products. It does not, however, provide specific metrics or thresholds that would enable the evaluation of the quality characteristics that have been determined. To address this issue, this article presents a quality environment made up of a quality model, an evaluation process, and a set of tools for evaluating one of the characteristics proposed by the ISO/IEC 25000 standard: functional suitability. In addition, the results of a case study are explained in detail. This study was carried out to evaluate the first software product that has achieved certification in functional suitability with a level 5 out of 5, where the practical application of the environment created was tested.

Key words

functional suitability, ISO/IEC 25000, software product quality evaluation

INTRODUCTION

Software quality is a hot topic at the moment, due mainly to the fact that software has spread its influence into a large part of industry and existing disciplines. This, in turn, has brought about the creation of larger and more complex projects (Wong et al. 2004). There are thus greater demands placed on the software that is being produced to ensure that it meets the necessary requirements. The use of software in spheres that are very sensitive and where mistakes are especially serious, such as healthcare, banking, and security, has meant that there are more users willing to accept that costs may rise, if that means there is a consequently higher level of software quality.

All of these considerations have, in recent years, led to an increase in the number of software life-cycle process certifications based on quality models (for example, CMM, CMMI, ISO 15504, ISO 12207, ISO 9001). These highlight the significance of quality assurance activities. It was certainly true that in the initial stages the software quality models focused on the basic concept that "a quality process produces a quality product" (Kitchenham and Pfleeger 1996; Maibaum and Wassyng 2008; Soley and Curtis 2013), and certification related to the quality of the software process appeared, such as ISO/IEC 15504 (ISO/IEC 2004) or CMMI (SEI 2010). Over the last few years, however, there has been a change of philosophy, with a new focus on software product quality. There is greater and greater acceptance of the idea that "software quality evaluations should be based on direct evidence about the product, not only on evidence about the process" (Maibaum and Wassyng 2008), since a high-quality process does not necessarily ensure a good-quality product. The last decade has witnessed the emergence of a large number of proposals for the assurance of software product quality. This is what led to the new ISO/IEC 25000 family of standards (ISO/IEC 2014), which aims to establish a standardized working framework for evaluating software products. The key parts of the ISO/IEC 25000 family are: ISO/IEC 25010 (ISO/IEC 2011a), which defines the quality model for software product, and ISO/IEC 25040 (ISO/IEC 2011b), which sets out the process for evaluating software product quality. The quality model proposed in the ISO/IEC 25010 standard is made up of eight quality characteristics, among which is functional suitability.

Functional suitability checks if the product or system provides functions that meet all the stated or implicit needs when used under specific conditions. According to the systematic review carried out in Rodríguez and Piattini (2012), functional suitability is one of the most relevant characteristics, and it is among those that generate the greatest interest. This is because having an evaluation available that indicates the level of fulfillment of the product's functional requirements helps ensure that the software product is suitable for the functions it must perform. The objective of this article, taking the aforementioned points into account, is to present a complete proposal for the evaluation of functional suitability and to demonstrate its application by means of a practical case study. The remainder of this article is organized as follows: first, there is an overview of the existing metrics for measuring functional suitability. The next section sets out the environment that the authors have built to evaluate functional suitability. Then, the authors present the case study of the evaluation of the functional suitability of a software product. Finally, conclusions are provided, along with an outline of work to be undertaken in the future.

ANALYSIS OF EXISTING FUNCTIONAL SUITABILITY METRICS

Research conducted on the evaluation of functional suitability has been appearing with ever-increasing frequency over the last few decades. A set of measures, whose objective is to carry out the evaluation of this characteristic, has come into being as a direct result of that work (Blanco et al. 2012). Thus, for the goal of this article to be reached, it was first necessary to study the metrics that had been defined; the aim was to find out if suitable measures for carrying out the evaluation of functional suitability in a pragmatic way already existed.

In performing this study of functional suitability metrics, the authors took as their starting point the results obtained in a systematic review of measures for evaluating functional suitability (Blanco, Reales, and Rodriguez 2012). In that review, the measures were matched with the subcharacteristics of functional suitability in ISO/IEC 25010. This made it easier both to analyze the metrics and to choose the ones to use in this research.

- Most metrics were proposed only theoretically. It is thus difficult to use these measures to carry out an evaluation of the functional suitability of a software product in real settings.
- Some of the measures can only be evaluated once the software product has been placed into the production environment. These metrics would therefore not be useful if one wishes to carry out an evaluation and certification of functional suitability before the delivery of the product to the client.
- There are measures that are complicated to obtain, due to the fact that they can be calculated only from the work products; in most cases, these are not produced during software product development.
- Metrics for measuring the same quality attributes, but that do so with different granularity (functionalities, tasks, requirements), have been found. There is obviously no point in measuring the same quality attribute once more, only with different granularity.

The authors should also mention that for a software product's functional suitability to be measured, there should be an environment that is made up of:

- A quality model, which sets out the subcharacteristics, the quality properties, and the metrics that are used to determine the value of the functional suitability of a software product
- An evaluation process, which describes the set of activities and tasks that are carried out when an evaluation process is conducted

- A technological environment, comprising a set of software tools that support the measurement, the application of evaluation criteria, and the visualization of the results

In 2012, AQCLab was granted an accreditation certificate for its software product maintainability evaluations by Entidad Nacional de Acreditación (ENAC), a Spanish accreditation body (Verdugo, Rodriguez, and Piattini 2014). Recently, AQCLab, to which the authors of this article belong, has adapted its evaluation environment for maintainability (Rodríguez and Piattini 2014) to evaluate a new quality characteristic: functional suitability. For that, AQCLab has developed:

- A quality model aligned with the ISO/IEC 25010 standard, which encompasses a set of measures that have none of the drawbacks outlined in earlier paragraphs and which tackle the evaluation of the functional suitability of the software product
- An evaluation process that adapts the activities defined in the ISO/IEC 25040 standards to perform an evaluation of the functional suitability of a software product (Rodriguez, Piattini, and Fernández 2015)
- A technological environment with software tools that make it possible to gather together the metrics on functional suitability and to calculate the indicators and subcharacteristics, as well as the value of the functional suitability

In 2016, the ENAC also granted AQCLab the accreditation certificate for the software product functional suitability evaluations.

AN ENVIRONMENT FOR THE EVALUATION OF FUNCTIONAL STABILITY

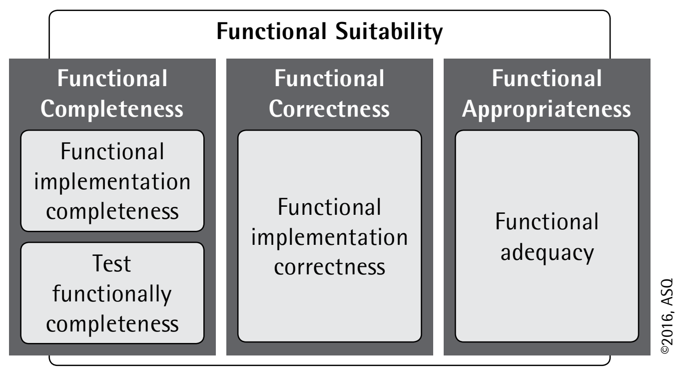
The following sections contain a description of each of the components that make up the environment for evaluation of functional stability built by the AQCLab.

Model of Functional Suitability

The model of functional suitability has been defined by taking the ISO/IEC 25010 standard as its basis. In this standard, functional suitability has changed; instead of having five subcharacteristics, as in ISO/IEC 9126 (ISO/IEC 2001), it now has three, since four subcharacteristics have been eliminated, and two new subcharacteristics have been added.

Functional suitability is defined in ISO/IEC 25010 as the degree to which a product or system provides functions that meet the stated or implicit requirements when used under specific conditions. So, the evaluation of functional suitability is understood as the degree to which a product or system conforms to the functional requirements (hereinafter referred as requirements) described in the product requirements specification, because it is impossible to know the implicit requirements of the different use context. ISO/IEC 25010 states that the characteristic of functional suitability can be evaluated on the basis of the three subcharacteristics set out in Figure 1.

FIGURE 1 Subcharacteristics of functional suitability



Evaluation of the subcharacteristics of functional suitability

The quality model defined for functional suitability begins precisely from the starting point given by the three quality subcharacteristics defined in the ISO/IEC 25010 standard for this characteristic. These are:

- **Functional completeness.** This is defined in ISO/IEC 25010 as the degree to which a set of implemented functions covers the specified tasks and meets the users' objectives. Functional completeness is understood as the ability of the system to provide the specified requirements in the product requirements specification.
- **Functional correctness.** This is defined in ISO/IEC 25010 as the degree to which a

product or system offers correct results, with the required degree of precision. Functional correctness is understood as the results generated by the requirements as expected.

- **Functional appropriateness.** This is defined in the ISO/IEC 25010 as the degree to which the functions facilitate the accomplishment of the tasks and objectives that have been set. Functional appropriateness is understood as the ability of the system to carry out the requirements that are needed for the different usage objectives that have been specified.

The functional suitability value is obtained from these subcharacteristics through a profiling function. The profiling function can get a result from a collection of values, avoiding problems caused by other functions such as mean or median. Moreover, the profiling function can change the scale of input and output values (that is, input values subcharacteristics [0-100] and output value functional suitability [1-5]). To understand the profiling function it is necessary to know the concepts: level, range, and profile. The set of input values is divided into intervals called *levels*. The established levels are shown in Table 1.

The output of the profiling function (functional suitability value) is obtained from the ranges. The *range* is a vector; it indicates the maximum number or percentage of items (subcharacteristics) that must be in each level to get the output value (functional suitability value). For example, range 3, shown in Table 2, indicates that to obtain a functional suitability value of 3 the following conditions must be met:

- There cannot be subcharacteristics in level 1, that is, subcharacteristics with a result lower than 25, as shown in Table 1.
- The maximum number of subcharacteristics in level 2 is one.
- The maximum number of subcharacteristics in level 3 is two.
- The maximum number of subcharacteristics in level 4 is three.

The optimal level is not used in the profiling function because the maximum number is always the maximum number of items or 100 percent (for this reason, in the profiling function appear only levels 1 to n-1, for this reason in Table 2 shows level 1 to 4). In this case, the maximum number is three, which is the number of subcharacteristics. Table 2 shows the ranges to obtain the different functional suitability values. These ranges were established from a study and validated to ensure that the functional suitability values generated are correct (Kitchenham and Pfleeger 2008).

The profile is the vector with the values obtained for each level. For example, suppose one evaluates a system and gets the following values for functional suitability subcharacteristics: 23.5, 55.8, and 87.9. Subcharacteristics are classified into levels depending on their value, as shown in Table 1. Therefore, a subcharacteristic belongs to level 1 (value 23.5), another subcharacteristic belongs to level 2 (55.8), and another subcharacteristic belongs to level 4. Thus, the system *profile* is (1, 1, 0, 1, 0): one subcharacteristic is in level 1, level 2, and level 4 and zero subcharacteristics are in level 3 and level 5.

After defining the basic concepts of the profiling function, the authors explain how it is applied. Once the system profile has been obtained (1, 1, 0, 1, 0), this profile is compared with the ranges defined in the profiling function, as shown in Table 2. The comparison must start with the higher range, in this case range 5, and it must continue with the lower ranges until the profile fulfills the conditions specified by the range selected. For

TABLE 1 Subcharacteristics: Quality levels

Level	Quality values	
1	0–25	
2	25–50	
3	50–75	
4	75–95	
5	95–100	

TABLE 2 Definition of the ranges for the evaluation

		Levels				Functional suitability values
		1	2	3	4	
Range	0	-	-	-	-	0
	1	3	3	3	3	1
	2	2	3	3	3	2
	3	0	1	2	3	3
	4	0	0	0	3	4
	5	0	0	0	0	5

example, the previous profile (1, 1, 0, 1, 0) is compared with the ranges shown in Table 2. The authors observed that the profile does not fulfill the conditions of range 5 because the profile has one subcharacteristic in level 1 and range 5 does not allow any subcharacteristics in this level. In ranges 4 and 3 the same thing happens. However, the profile fulfills the conditions of range 2 because the profile has only one subcharacteristic in level 1 and range 2 allows a maximum of two subcharacteristics and the rest of conditions are fulfilled. Thus, the functional suitability value for the *profile* (1, 1, 0, 1, 0) is 2, the value indicated for range 2, as shown in Table 2.

Functional suitability is evaluated from subcharacteristics defined in ISO/IEC 25010 and the profiling function described previously. But ISO/IEC 25010 does not define metrics to get the results of the subcharacteristics. To make this system operative, the authors defined a set of quality properties and measures defined according to the ISO/IEC 25023 (ISO/IEC 2016). The study of metrics that the authors described previously was used as the basis for choosing these properties and measures.

The evaluation is carried out using a bottom-up process, so that the lowest-level metrics are calculated first. As one can see on the lefthand side of Figure 2, each of these metrics has its own range: number of requirements, number of requirements tested, and so on. From the values obtained in the metrics, one goes up to the next level in the hierarchy, so as to obtain a quality value to be assigned to the property that is applied. In this case, the value of the properties has been normalized to between 0 and 100 (0 is the lowest quality value that a property may have, and 100 is its highest possible value).

After obtaining the values of all the properties, one can work out the value of each of the subcharacteristics of quality. Each of these subcharacteristics will be influenced by one or more of the properties for which the value has been calculated previously. As one can see on the lefthand side of Figure 2, the value of the subcharacteristics has once again been normalized to a range of between 1 and 100.

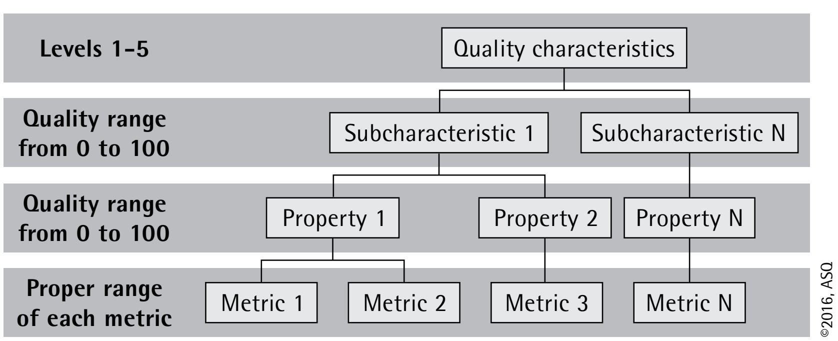
When the values of all the subcharacteristics have been calculated, one can work out the quality level of the functional suitability, following the profiling functions of Table 2. The authors will now present the particular quality property used to evaluate each subcharacteristic in the model.

Properties of functional completeness

The properties used to evaluate functional completeness are:

- **Functional implementation completeness.** The completeness of the functional implementation takes into account the specific tasks that the software product can carry out (Yahaya and Deraman 2010). To obtain the value of this property, the following are found using the metrics: the number of requirements that are implemented and the number of specified requirements that have been included in the product specification and indicated by the user or project owner; with these, one obtains the list of requirements that have been implemented in the product.
- **Functional test completeness.** This property studies whether the requirements that have been described and obtained from the requirements specification have undergone a process of testing. In this way the functional completeness is evaluated by comparing the requirements that have undergone a test process with those that have been implemented. If a requirement has not been subject to a testing process, one cannot know whether it has been completely implemented (Sacha 2006). To obtain the value of this property, the following are found by means of the metrics:

FIGURE 2 Hierarchy and elements that have an influence on the evaluation of the software product



the number of requirements implemented, along with the number of requirements that have been tested, and the percentage of statement coverage to execute functional test case systems. To measure statement coverage it is necessary to use tools like Emma, Opencover, Simplecov, and so on. The list of tested requirements is thereby obtained, giving assurance that they have been implemented completely.

Property of functional correctness

The property used to evaluate functional correctness is:

- **Functional implementation correctness.** This property evaluates the level of correctness in the requirements implementation. In other words, it evaluates if the requirements implemented are doing what was expected of them. In order to discover the correctness of a system, one must check whether the functional requirements detailed in the requirements specification are being fulfilled (Mehmood and Cherfi 2009). To obtain the value of this property, the following are found by means of the metrics: the number of requirements implemented, along with the number of requirements that have been tested, as well as whether the results have been satisfactory, and the percentage of statement coverage to execute functional test case systems. One can thereby obtain a list of requirements that present the behavior desired by the user, and that was established during the analysis of the product by the users or clients.

Property of functional appropriateness

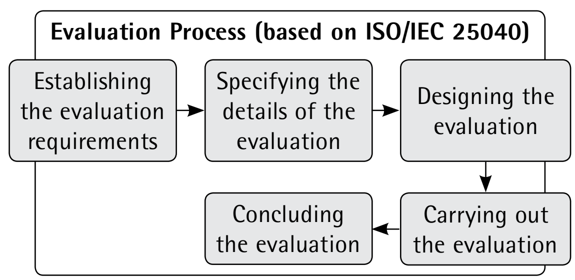
The property that is used to evaluate functional appropriateness is:

- **Functional adequacy.** This property refers to whether the software product fulfills the requirements of each type of user. The requirements for each type of user are called *usage objectives*. In other words, it is about whether the product software provides all the usage objectives as expected. In order to obtain the value of this property, first the requirements of each usage objective need to be identified, as well as the proportion of the requirements where the result is correct for each usage objective and the degree to which the usage objectives are appropriate. In this way, one can discover whether the software product is appropriate for all types of users because it provides the requirements expected.

The Evaluation Process

The evaluation process for functional suitability is an adaptation of the process used by AQCLab for evaluating maintainability, based on the ISO/IEC 25040 standard (see Figure 3). It is made up of five activities, in which the following tasks are performed:

FIGURE 3 Evaluation process AQCLab



- **Activity 1:** Establishing the evaluation requirements. In this activity the purpose of the evaluation is established. The quality requirements of the product are obtained, and the parts of the product to be evaluated are identified.
- **Activity 2:** Specifying the details of the evaluation. An evaluation model is chosen, the decision criteria for the metrics are determined, and the decision criteria for the evaluation are established.
- **Activity 3:** Designing the evaluation. The evaluation activities are planned.
- **Activity 4:** Carrying out the evaluation. The measurements are made, the decision criteria are applied to the quality measures, and the design criteria are then applied to the evaluation. This is where the greatest effort has been made, adapting the evaluation

process that AQCLab already had at its disposal for maintainability. This activity is much more manual in the case of functional suitability, and therefore more tedious. This is because the evaluator has to identify the software product requirements that have been implemented, run the functional test case to verify if the results of the software product are as expected and measure statements coverage, and find out which requirements are not relevant to the user.

- **Activity 5:** Concluding the evaluation. The results of the evaluation are checked, an evaluation report is produced, and the quality of the evaluation is checked. The data are then disposed of or treated to ensure that AQCLab is not storing any confidential data once the evaluation has been finished.

Technological Environment

The last element that must be tackled in the evaluation of the functional suitability characteristic is the technological environment. AQCLab already had a technological environment at its disposal for evaluating maintainability (see Figure 4), and this was adapted for the construction of the one needed for evaluating functional suitability. The technological environment of AQCLab is made up of three levels:

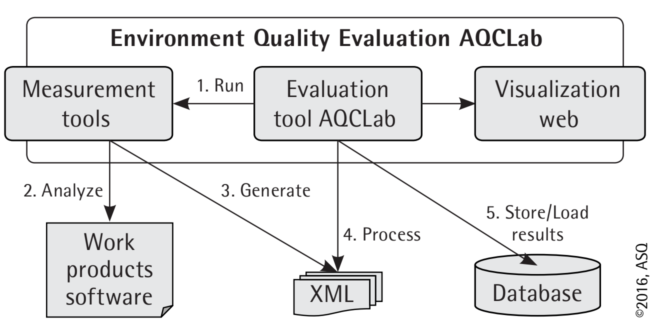
- **Measurement tools:** This is the first level, and its mission is to generate the results of the measures (in an XML file). The advantage of this level is that it is easily extendible by adding new tools that make it possible to calculate metrics for other quality characteristics.
- **Evaluation tool AQCLab.** This is the second level; its goal is to obtain the values for the properties, subcharacteristics, and quality characteristics, based on the results of the measures from the level below.
- **Visualization web.** This is the third level, and its role is to present the information obtained from the software product evaluation in an understandable form. Visualization is of great use in the evaluation of software product quality (Moraga, Calero, and Bertoa 2010). Figure 5 demonstrates how the AQCLab tool displays the evaluation results. This tool shows the results quality characteristics, subcharacteristics, properties, and metrics of the quality evaluation made by AQCLab. Moreover, this tool generates reports with the results of the evaluations.

A new measurement tool had to be developed so that the environment of AQCLab could be adapted (first level). This tool, called the *functional suitability tool*, has the aim of generating an XML file, which collects the results of the metrics employed in the evaluation of the functional suitability of the software product. To generate these results, the tool allows the evaluator to register all the information of the requirements that the software product should implement to meet the requirements set out at the beginning of the life cycle.

Once the requirements have been registered, the application lets the evaluator indicate the state the requirement is in, that is, whether it is specified, implemented, or tested. The tool also makes it possible to associate the requirements with its usage objective.

The authors should also point out that to visualize the evaluations of functional suitability, an adaptation of the visualization environment of AQCLab was carried out. This has permitted the visualization of both of the characteristics that its environment currently evaluates: maintainability and functional suitability.

FIGURE 4 Environment quality evaluation AQCLab

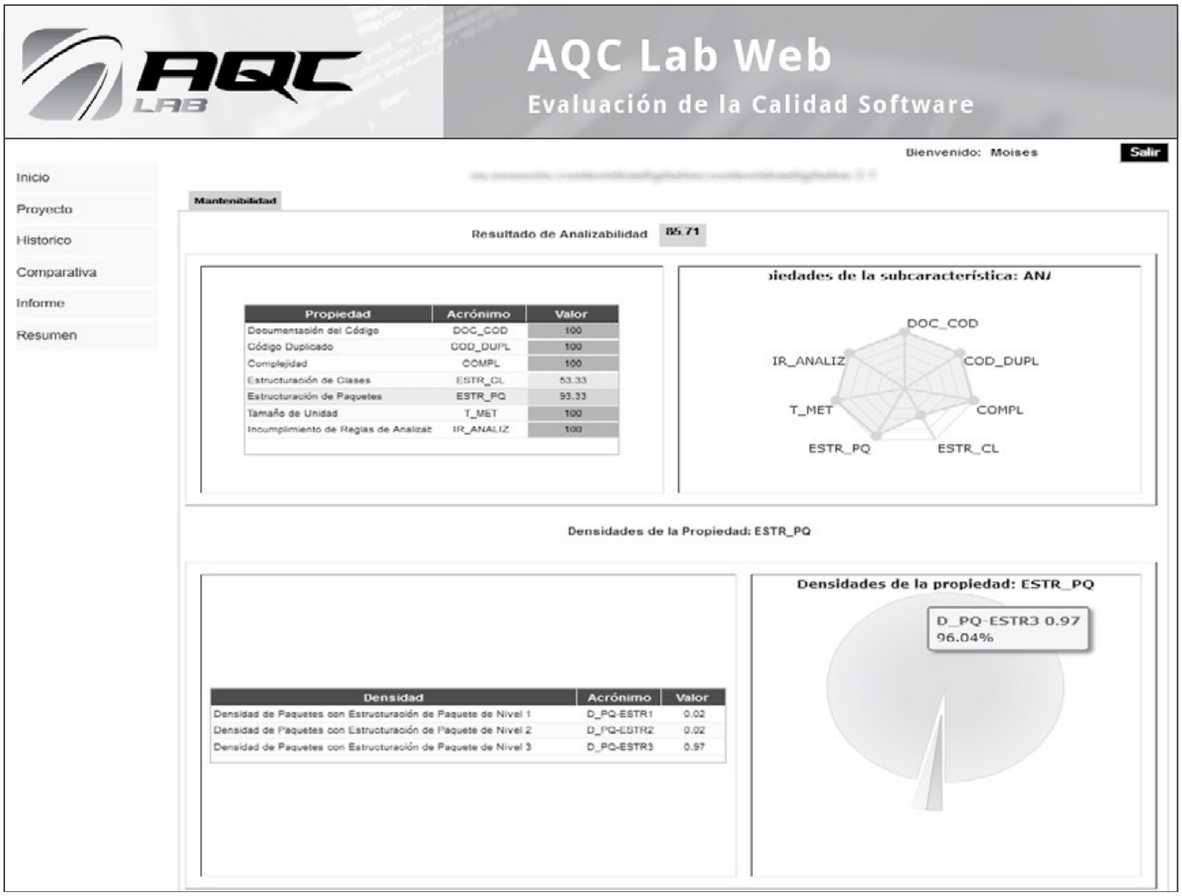


A CASE STUDY

Once developed, the evaluation environment functional suitability carried out a case study to evaluate the functional suitability of a software product. The case study follows the guidance described by Runeson (Runeson et al. 2012).

The aim of the case study is to verify evaluation environment functional suitability, which can be used to evaluate the functional suitability of a software product. Furthermore, it is desired

FIGURE 5 Visualization environment of AQCLab



to detect the difficulties that may appear in evaluating the functional suitability of real software product. Therefore, the software product presented in the following section was selected for the case study.

Presentation of the Company and the Evaluated Product

Bitware S.L. is a firm involved in business consulting, IT projects, processes, and general solutions within a technological framework. It has a staff of about 30 people, the majority of whom work in activities related to the analysis, design, and development of information systems.

Bitware has a system of project management and software development that is certified by AENOR against ISO/IEC 15504 – 12207 model, at maturity level 3. It is the first company in Spain to have achieved certification in software product maintainability at level 5 against ISO/IEC 25000 AENOR Software Maintainability Conform Certification (Rodríguez et al. 2015). The project evaluated in this case study is BitRRHH 1.0, a web application for managing the human resources of an organization.

After presenting the product employed in a case study, the following case study activities are data preparation and collection. To obtain these data it is necessary to evaluate BitRRHH 1.0 using environment evaluation functional suitability. The following section describes the tasks that were performed during the evaluation.

Evaluation Process for Functional Suitability

In the case study, two evaluations of functional suitability were performed at BitRRHH 1.0. In each aseessment, evaluation environment functional suitability was employed. The work products that were used to carry out assesments are shown in Table 3.

TABLE 3 Work product for assessment

Work product for assessment
BitRRHH_RequirementsEspecification.pdf
BitRRHH_TestPlan.pdf
BitRRHH_UserManual.pdf
BitRRHH_SourceCode.zip
BitRRHH Application

First evaluation

In the first evaluation, the following tasks were carried out:

- **Activity 1. Establishing the evaluation requirements.** In this activity a number of meetings were held with Bitware; the model was explained in these sessions, as was the evaluation process. Bitware presented its requirements and needs, and an evaluation plan was started, based on this information.
- **Activity 2. Specifying the evaluation.** In this activity, the evaluation module was determined. In this particular evaluation, however, there was no need to specify the evaluation module, as the objective was to apply the functional suitability model defined earlier in this article.
- **Activity 3. Designing the evaluation.** The scope of the evaluation was presented in detail in this activity. In this case the scope was the whole product—the activities needed to carry that out were planned.
- **Activity 4. Performing the evaluation.** In this activity, the measurement of the metrics for functional suitability was carried out. The authors ran BitRRHH 1.0 and analyzed its documentation. They discovered that there were specified requirements that had not been implemented in BitRRHH 1.0; only 84 requirements were implemented. Moreover, they found that some requirements have no associated functional test cases in the test plan and some tests do not generate the results indicated in the test plan. Only 68 requirements generated the expected result; the rest generated unexpected results. Also, they were measuring statement coverage with Opencover Tool while they were running the functional test case and obtained only 22.45 percent statements. Finally, the authors checked whether the usage objectives defined for BiRRHH 1.0 were achieved and found that two usage objectives were fully satisfied and the rest (other two) were partially satisfied. The results obtained in measuring the product are shown in Table 4. Table 4 only shows the results of the base metrics. These results are used for obtaining the values of the properties described previously.

TABLE 4 Results of the measurement of functional suitability in the first evaluation

Metrics	Value
Number of requirements specified	86
Number of requirements implemented	84
Number of requirements tested	68
Number of requirements tested successfully	45
Usage objective satisfied	2
Statement coverage of functional test cases	20.45%

- **Activity 5. Finishing the evaluation.** To conclude the evaluation, a detailed report was drawn up, giving the level of functional suitability reached and the results obtained in each of the subcharacteristics of which the model is made up. In addition, an exhaustive report was produced, giving details of those requirements that were not completed and correct, and usage objectives that were not met by the product. The overall level of functional suitability for the product was 2 (insufficient quality).

After completing this first evaluation, the results of the evaluation and the observations of the evaluator were analyzed. It was concluded that the evaluation environment allows for the evaluation of the functional suitability of a software product from technical product documentation and running of the software product. Furthermore, the result of functional suitability is justified, namely, the functional suitability value of 2 on the 1-to-5 scale is justified by the low number of functional test cases, which means they could not ensure the product from working properly. After analyzing the results with

Bitware and verifying that the product did not reach the desired level of functional suitability, it was agreed to do a re-evaluation when the development team resolved the issues indicated by AQCLab. So, after two weeks of work by Bitware, a second evaluation was performed.

Second evaluation

After the changes carried out by Bitware on the product, AQCLab performed a new evaluation. This evaluation left out the first two activities, since that would simply have repeated work already done in the first evaluation, contributing nothing new. The second evaluation, therefore, began with activity 3; the following tasks were performed:

- **Activity 3. Designing the evaluation:** In this activity, the action plan for carrying out the second software product evaluation was produced.
- **Activity 4. Performing the evaluation:** In this activity, the new version of the software product was measured, returning to perform tasks carried out in activity 3 of the first evaluation. The values from the metrics provided the data for a fresh evaluation of the functional suitability. The results obtained from measuring the product are shown in Table 5.

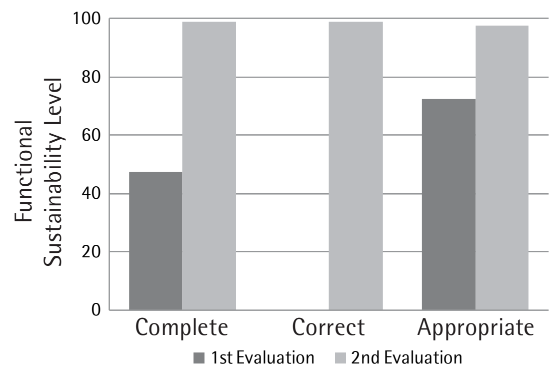
TABLE 5 Results of the measurement of functional suitability in the second evaluation

Metrics	Value
Number of requirements specified	93
Number of requirements implemented	92
Number of requirements tested	92
Number of requirements tested successfully	91
Usage objective satisfied	4
Statement coverage of functional test cases	96.64%

- **Activity 5. Finishing the evaluation:** To conclude the second evaluation, a new report was produced containing the results obtained. This time, after the corrections made by Bitware, the level reached was 5 (excellent quality). This meant that there was no need to produce any report to identify possible improvements in the software product.

As seen from the results of the metrics obtained in the second evaluation (see Table 5), Bitware solved all the problems with its product, improving the result of the metrics and consequently the results of the evaluation of the functionality suitability (see Figure 6). After analyzing the results with Bitware and checking that the results of the evaluation were desired, the next phase of the case study—analysis and interpretation of the results—was conducted.

FIGURE 6 Results of the first and second evaluation



Analysis and Interpretation of the Results

After performing two evaluations of BitRRHH 1.0 and analyzing the annotations made by the evaluator one can conclude that the evaluation environment allows carrying out evaluations of functional suitabilty for a software product, according the model defined by AQCLab based on ISO/IEC 25000. In addition, the results of functional suitability evaluations showed a consistent level of compliance with the requirements of the product, since the relationship between compliance requirements and functional suitability level was ratified by the Bitware team. Furthermore, in March 2015, AENOR certified functional suitability of BitRRHH 1.0, as shown in Figure 7, thus validating that the results obtained in the evaluation of functional adequacy are consistent with the number of product requirements implemented and the expected behavior.

CONCLUSIONS

Software product quality is one of the main concerns of software development companies. This article presented an environment for the evaluation of the functional

FIGURE 7 Software product quality certificate

Certificate of Conformity
Quality of Software Product

Software Product ISO 25000 FUNCTIONAL SUITABILITY AENOR conform

PS-2015/0001

AENOR, Spanish Association for Standardization and Certification, certifies that the Software Product

BITRRHH, versión 1.0

of the company

BITWARE, S.L.

Wich is/are carried out in: BUGANVILLA, 5 1ºD 28036 MADRID

In accordance with.: ISO/IEC 25000 Standard Software engineering – Software product Quality Requirements and Evaluation (Square)

For the characteristics: FUNCTIONAL SUITABILITY

Certification system: This certificate is valid considering the assessment report ID: BITRRHH-1.0 with date 2014-09-23 of the AQC Lab laboratory accredited by ENAC and the audit report of software product certification of AENOR PS-01/15 with date 2015-02-23

Issued on: 2015-02-27
Validity date: 2021-02-27

Avelino BRITO MARQUINA
Director General de AENOR

AENOR Asociación Española de Normalización y Certificación | Génova, 6. 28004 Madrid. España Tel. 902 102 201 – www.aenor.es

suitability of the software product. In addition, and to demonstrate the practical application of the evaluation of functional suitability, the authors presented the results of a case study where it was shown that the evaluation environment of functional suitability can be used to evaluate real software products.

After the case study was completed, it was realized that the evaluation environment for functional suitability is independent of the technology and the programming language used in the development of the software product. In addition, the case study identified work products required to address an evaluation of functional suitability: requirements specification, functional test cases, source code, and software product run. Thus, work products to be used in evaluating functional suitability can be standardized and therefore can reduce subjectivity in the evaluation.

Another challenge is how to perform the evaluation. For functional suitability, unlike in the case of evaluations of maintainability, the majority of evaluation tasks are carried out manually. In addition, due to the close relationship that exists between the business logic of software products and some of the measures of functional suitability, it is advisable to produce user manuals for the software product to aid the evaluator in the task of getting to know the functionalities offered by the product, as well as how they are actually executed.

The authors should add that all the work presented in this article has allowed AQCLab to increase the size of its laboratory (evaluation process, quality model, and measurement tools), and thus be able to tackle the evaluation of functional suitability.

Finally, the Asociación Española de Normalización y Certificación (the Spanish Association for Standardization and Certification) (AENOR) has begun to certify software products for the characteristic of functional suitability, as is already done with the characteristic of maintainability, thanks to reports from AQCLab generated after applying the evaluation environment presented in this article.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments. This work was funded by the SDGear project (Ministerio de Industria, Energía y Turismo dentro del Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica 2013-2016 and Fondo Europeo de Desarrollo Regional FEDER, TSI-100104-2014-4), by the SEQUOIA project (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2015-63502-C3-1-R), and by the GLOBALIA projects (Consejería de Educación, Ciencia y Cultura [Junta de Comunidades de Castilla La Mancha]) and Fondo Europeo de Desarrollo Regional FEDER.

REFERENCES

Blanco, A., P. Reales, and M. Rodríguez. 2012. Metrics to evaluate functional quality: A systematic review. In *Proceedings of the 7th Iberian Conference on Information Systems and Technologies* (CISTI 2012), Madrid, Spain.

ISO/IEC. 2001. ISO/IEC 9126:2001, Software product evaluation–Quality characteristics and guidelines for their use. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. 2004. ISO/IEC 15504:2003, Information technology software process assessment. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. 2011a. ISO/IEC 25010:2011, Software engineering – Software product quality requirements and evaluation (SQuaRE) – System and software quality model. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. 2011b. ISO/IEC 25040:2011, Systems and software engineering – Systems and software quality requirements and evaluation (SQuaRE) – Evaluation process. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. 2014. ISO/IEC 25000:2014, Systems and software engineering – System and software quality requirements and evaluation (SQuaRE) – Guide to SQUARE. Geneva, Switzerland: International Organization for Standardization.

ISO/IEC. 2016. ISO/IEC FDIS 25023:2016, Systems and software engineering – Systems and software quality requirements and evaluation (SQuaRE) – Measurement of system and software product quality. Geneva, Switzerland: International Organization for Standardization.

Kitchenham, B., and S. L. Pfleeger. 1996. Software quality: The elusive target. *IEEE Software* 20, no. 1:12-21.

Kitchenham, B., and S. L. Pfleeger. 2008. Personal opinion survey. In *Guide to Advanced Empirical Software Engineering*, eds F. Shull, J. Singer, D. I. K. Sjoberg. New York, NY: Springer.

Maibaum, T., and A. Wassyng. 2008. A product-focused approach to software certification. *Computer* 41, no. 2:91-93.

Mehmood, K., and S. Cherfi. 2009. Evaluating the functionality of conceptual models. *LNCS Advances in Conceptual Modeling-Challenging Perspectives* 5833/2009: 222-231.

Moraga, M. A., C. Calero, and M. F. Bertoa. 2010. Improving interpretation of component based systems quality through visualization techniques. *IET Software* 4, no. 1:79-90.

Rodríguez, M., and M. Piattini. 2012. Systematic review of software product certification: In *Proceedings of the 7th Iberian Conference on Information Systems and Technologies* (CISTI 2012), Madrid, Spain, 631-636.

Rodríguez, M., and M. Piattini. 2014. Software product quality evaluation using ISO/IEC 25000. *ERCIM NEWS* (October):39-40.

Rodríguez, M., M. Piattini, and C. Fernández. 2015. A hard look at software quality. *Quality Progress* (September): 30-37.

Runeson, P., M. Host, A. Rainer, and B. Regnell. 2012. *Case study research in software engineering guidelines and examples.* New York, NY: Wiley.

Sacha, K. 2006. Evaluation of expected software quality: A customer's viewpoint. *LNCS – Fundamental Approaches to Software Engineering*, 170-183.

SEI. 2010. CMMI® for Development, version 1.3 (CMMI-DEV, V1.3). Improving processes for developing better products and services, Technical Report. Pittsburgh, PA: Software Engineering Institute.

Soley, R. M., and B. Curtis 2013. The consortium for IT software quality (CISQ). In *Proceedings of the 5th International Conference Software Quality Days* (SWQD 2013). Springer. Vienna, Austria, 3-9.

Verdugo, J., M. Rodríguez, and M. Piattini. 2014. Using agile methods to implement a laboratory for software product quality evaluation. XP2014. In *Proceedings of the 15th International Conference on Agile Software Development*, Rome, Italy, 143-156.

Wong, B., B. Boehm, J. Verner, and S. Chulani. 2004. Second workshop on software quality. In *Proceedings of the 26th International Conference on Software Engineering* (ICSE' 04), St. Louis, MO.

Yahaya, J. H., and A. Deraman. 2010. Measuring unmeasurable attributes of software quality using pragmatic quality factor. In *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology* (ICCSIT), Sintok Malaysia, 197-202.

BIOGRAPHIES

Moisés Rodríguez is founder and CIO of AQCLab, the first accredited laboratory for software product quality assessment using ISO/IEC 25000. His research interests include software product and process quality, software metrics, software testing, and data quality.

Rodríguez has a doctorate in computer science from the University of Castilla-La Mancha and he is a certified information system auditor (CISA) by ISACA, chief auditor in ISO/IEC 15504 by AENOR, and TMAP Next certified. He is a member of the Alarcos Research Group at UCLM. He can be reached by email at mrodriguez@aqclab.es.

Jesús R. Oviedo is software quality consultant at AQCLab, the first accredited laboratory for software product quality assessment using ISO/IEC 25000. His research interests include software product and process quality, software metrics, software testing, and data quality. Oviedo has a bachelor's degree in computer science from the University of Castilla-La Mancha and he is a certified information system auditor (CISA) by ISACA. He can be reached by email at: joviedo@aqclab.es.

Mario Piattini is a professor at the University of Castilla-La Mancha. His research interests include software quality, metrics, and maintenance. Piattini has a doctorate in computer science from the Technical University of Madrid and leads the Alarcos Research Group at UCLM. He is founder and scientific director of AQCLab. He can be reached by email at: mario.piattini@uclm.es.