A Lightweight Software Product Quality Evaluation Method

Giuseppe Lami and Giorgio Oronzo Spagnolo
System and Software Evaluation Lab, Information Science and Technologies Institute,
National Research Council, Pisa, Italy

Keywords: Software Product Quality, ISO/IEC 25000, Quality Evaluation.

Abstract: In this paper, we describe an evaluation method, called QuESPro (Quality Evaluation of Software Product), aimed at performing third party evaluation of the suitability for the intended use of software products, by targeting a trade-off between the mere informal expert judgment and the application of complex and expensive evaluation methods. The QuESPro is based on the framework provided by the ISO/IEC 25000 series standard and provides a step-wise process to determine a quantitative evaluation of the relevant quality characteristics of software products. With the aim of assessing the feasibility of the QuESPro method in terms of feasibility, identifying its strengths, and identifying improvement opportunities we applied it to an industrial case study. The results of such a case study are reported in this paper as well.

1 INTRODUCTION

Software is today pervasive and crucial for the business of companies and organizations as several vital functions are reliant on software solutions. For managers the information about the fitness of software products in use with respect the current and future business needs is pivotal for strategic decisions and investments. Often organizations do not have the capability to gather such an information as the software product they want to evaluate is developed by external software houses. For this reason, they refer to third party, independent, and qualified organizations to perform evaluations of software products aimed at understanding the degree of adherence to their demands and needs of in use software solutions. In addition, it is worth saying that the currently available methods to evaluate a software product in a systematic, quantitative, and sound way are generally complex, time consuming and expensive thus, primarily for small and medium enterprises, that represent a barrier to perform such activities. To face such a situation the System & Software Evaluation Lab (SSE) of the Information Science and Technologies Institute, as a third-party independent evaluation body with experience in assessing process and evaluating software products, defined a methodology to evaluate software products adopting a systematic and sound approach targeting cost-effectiveness of the evaluation. Such a methodology, that is based on the quality model provided by the ISO/IEC 25010 standard (ISO, 2011), has been identified with the acronym QuESPro (Quality Evaluation of Software Product), is described in this paper. There is a large literature describing quality evaluations based on or inspired by the quality model provided by ISO/IEC 25010 (Miguel, 2014), (Ouhbi, 2014). Some of them aim at defining specific procedures to perform software quality evaluations (Rodriguez, 2016), (Lee, 2014). Some other are focused on extending or customizing the ISO/IEC 25010 quality model to fit specific contexts (Falco, 2021), (Neri, 2018), (Ortega, 2003), (Estdale, 2018), (Nakai, 2016).

As the application of those methods is often complex and expensive, they are hard to be applied by small-medium enterprises. The QuESPro method described in this paper aims at responding to the demands of short-term and cost-effective quality evaluations of software products that are sounder and more systematic than the mere expert judgment, but not highly demanding in terms of costs and time

This paper is structured as follows: in section 2 the QuESPro methodology is described then, in section 3, the experience of the application of the QuESPro methodology in an industrial case study is described and the related results presented. Finally, in section 4, conclusions and lessons learned are provided.

Lami, G. and Spagnolo, G.
A Lightweight Software Product Quality Evaluation Method.
DOI: 10.5220/0011316400003266
In *Proceedings of the 17th International Conference on Software Technologies (ICSOFT 2022)*, pages 524-531
ISBN: 978-989-758-588-3; ISSN: 2184-2833
Copyright © 2022 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

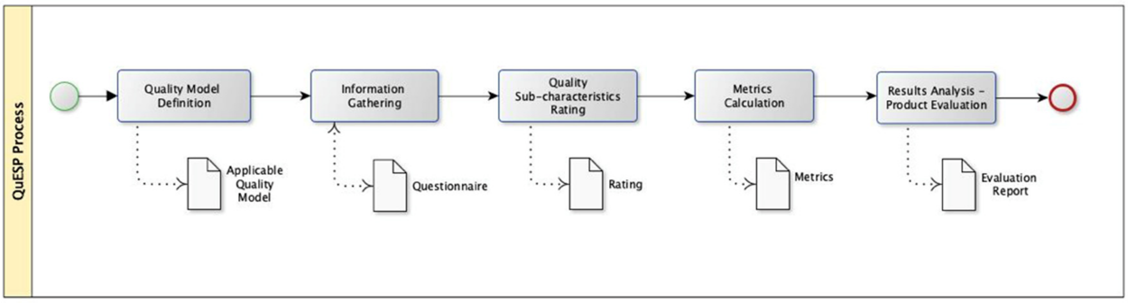


Figure 1: QuESPro process.

2 QUESPRO EVALUATION METHOD

In this section, we describe the QuESPro method to perform lightweight software product quality evaluation. The QuESPro method has been setup at the SSE to face the demands coming from industry for third-party evaluation of existing software products with a reasonable balance between strictness and cheapness. To be effective and repeatable, software product quality evaluation method shall rely on a well-defined evaluation process, which describes the set of activities and tasks that are carried out when an evaluation activity is conducted along with the related outcomes.

The QuESPro method is composed of the following phases.

1. Quality model definition,
2. Information gathering,
3. Quality sub-characteristics rating,
4. Calculation of metrics,
5. Evaluation results reporting and improvement areas identification.

Figure 1 describes, by means of a diagram in BPMN (OGM, 2013), the sequence of the phases of the QuESPro method, along with the indication of the outcomes. In the following sub sections, each phase of the evaluation process is described in more detail.

2.1 Phase 1: Quality Model Definition

A Quality Model is defined as a "set of characteristics, and of relationships between them, which provides a framework for specifying quality requirements and evaluating quality" (ISO, 2011).

The Quality Model is the cornerstone of a product quality evaluation method.

The quality of a software product is the degree to which that software product satisfies the stated and implied needs of its various stakeholders, and thus provides value. Those stakeholders' needs are precisely what is represented in the reference quality model, which categorizes the product quality into characteristics which, if necessary, are divided into sub-characteristics.

The starting point for the definition of the Quality Model used in the QuESPro method is the ISO/IEC 25010 (ISO, 2011) standard that provides a wide-spectrum, generally accepted, quality model for software products.

The ISO/IEC 25010 provides a product quality model composed of eight characteristics (which are further subdivided into sub-characteristics) that relate to static properties of software and dynamic properties of the computer system. The quality characteristics and the related sub-characteristics of the ISO/IEC 25010 Product quality model are provided in Table 1.

For space limits, the definition of the Quality (sub-)characteristics in Table 1 is not provided in this paper, to get them refer to (ISO/IEC, 2011). The Quality Model provided by the ISO/IEC 25010 standard is general as it contains a wide spectrum of quality characteristics and sub-characteristics.

The relevance of those (sub-) characteristics may vary according to the specific software product under evaluation and its context of use. For this reason, it should be possible to tailor the quality model it to identify the most relevant (sub-) characteristics and focus the evaluation only on those. The Quality Model tailoring allows to reduce the complexity of the evaluation process as well.

The QuESPro method addresses the Quality Model tailoring by means of the prioritization of the quality sub-characteristics of the ISO/IEC 25010 quality model, with the aim of giving higher priority to the sub-characteristics more relevant for the product under evaluation.

Relevance is a property related to a quality sub-characteristic and indicates the degree to which the

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

overall quality and the intended use of the software product under evaluation depends on the fulfilment of that quality sub-characteristic.

The prioritization is conducted by using a four-values scale representing the degree of relevance for the intended purpose:

3 – highly relevant
2 – moderately relevant
1 – slightly relevant
0 - not relevant

To determine the degree of relevance of the quality sub-characteristics, the principal aspects considered are the implemented functionalities, the product's context of use, and the stakeholders needs.

Thus, to determine the degree of relevance of each quality sub-characteristic, the following criteria have been identified:

- Impact on costs: the severity of problems (or the costs related to the occurrence of problems) due to possible lacks in terms of the sub-characteristic in the current/forecast context of use. For example: the sub-characteristic Modifiability is highly relevant in the case the product is sold to many different customers each of them needing a customized version of the product to fit with its specific demands.
- Impact on Stakeholders: The extent to which the current/forecast stakeholders needs are affected in the case of lacks in terms of the sub-characteristic. For example: the sub-characteristic Operability is highly relevant in the case a software product targeting users without experience in using that technology.
- Impact on functionality: The extent to which the current/forecast implemented functionalities are affected in the case of lacks in terms of the sub-characteristic. For example: the sub-characteristic User Interface Aesthetics is not relevant in the case of a software product implementing a kernel function of an OS.

To support and make more systematic the determination the degree of relevance, an ad hoc check list reflecting the above criteria has been developed. The determination of the degree of relevance shall be based on the understanding of the actual context of use, stakeholders needs, and product properties. This phase is intended to be performed with a strict interaction between evaluators and users and developers of the software product under evaluation.

The outcome of this phase is the applicable Quality Model, derived, starting from the ISO/IEC 25010, by removing those quality sub-characteristics with a degree of relevance lower than 2.

2.2 Phase 2: Information Gathering

The collection of the necessary information to perform the rating of the quality sub-characteristics is based on a specific Questionnaire and on interviews to key stakeholders.

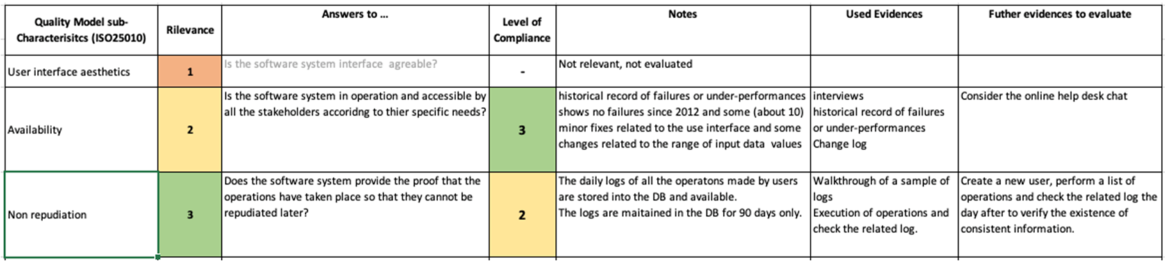
The Questionnaire is aimed at gathering basic data regarding the functional and non-functional characteristics of the software product under evaluation and the hardware environment used to execute it. The questionnaire is divided into several parts each of them composed of specific open questions targeting the architecture of the software, the involved software components and the related interfaces, the quality of the source code, available user and maintenance documentation, performance and security measures in place, and working load capability. At the discretion of the evaluators, the answers given may be required to be corroborated by the analysis of technical documentation (as, for instance, software architectural design, protocol specification, instrumental measures, …).

The interviews to key stakeholders are aimed at confirming and completing the information obtained by means of the questionnaire. The persons to involve in the interviews are developers and maintainers of the software product under evaluation (for the aspects related to the constructive characteristics) and users/supervisors (for the functional and performance aspects). It is recommended to conduct the interviews in combination with live-run show of the software under evaluation in order to confirm and enforce the answers with concrete evidences.

The outcomes of this phase are the questionnaire filled and interviews minutes containing the important information obtained.

2.3 Phase 3: Rating of Sub-Characteristics

On the basis of the evidences collected in the information gathering phase, the quality sub-characteristics are considered and evaluated in terms of the extent they are fulfilled by the product under evaluation. Such an evaluation results in a rating on a four-values scale called Level of Compliance. The ratings represent the degree of fulfilment of the quality sub-characteristic they refer to, and indirectly they indicate the related level of risk.



Quality Model sub-Characterisitcs (ISO25010)	Rilevance	Answers to ...	Level of Compliance	Notes	Used Evidences	Futher evidences to evaluate
User interface aesthetics	1	Is the software system interface agreable?	-	Not relevant, not evaluated		
Availability	2	Is the software system in operation and accessible by all the stakeholders accoridng to thier specific needs?	3	historical record of failures or under-performances shows no failures since 2012 and some (about 10) minor fixes related to the use interface and some changes related to the range of input data values	interviews historical record of failures or under-performances Change log	Consider the online help desk chat
Non repudiation	3	Does the software system provide the proof that the operations have taken place so that they cannot be repudiated later?	2	The daily logs of all the operatons made by users are stored into the DB and available. The logs are maitained in the DB for 90 days only.	Walkthrough of a sample of logs Execution of operations and check the related log.	Create a new user, perform a list of operations and check the related log the day after to verify the existence of consistent information.

Figure 2: Example excerpt from a rating report.

Level of Compliance rating values are:

3: Good [the quality sub-characteristic is substantially fulfilled. The risk of occurrence of problems related to the sub-characteristic is low]
2: Sufficient [the quality sub-characteristic is largely fulfilled. The risk related to the sub-characteristic is medium]
1: Insufficient [the quality sub-characteristic is partially fulfilled. The risk of occurrence of problems related to the sub-characteristic is high]
0: No fulfilled [the quality sub-characteristic is not fulfilled at all]

The outcome of this phase is a rating record, that contains the rating values accompanied with the indication of the sources of evidences used for the rating and some notes aimed at indicating possible critical issue and for justifying any possible low ratings (Figure 2. shows an example excerpt from a rating report). This is done with the purpose of allowing an ex-post analysis of the rating and thus enforcing its repeatability.

2.4 Phase 4: Calculation of Adequacy Metrics

The reference metric for the evaluation of the quality sub-characteristics is called: *Adequacy*.

The Adequacy metric is applicable to each quality sub-characteristic of the quality model. The calculation of the Adequacy metric relies on the value assigned to two indicators: Level of Compliance (LC) and the Degree of Relevance (DR), which are introduced and discussed in the above sub-sections. The Adequacy metric provides a measure of the extent to which the software product under evaluation exposes technical and functional characteristics that adequately respond to a quality sub-characteristic.

Adequacy: *f* (LC, DR)

The Adequacy metrics is based on a four-values ordinal scale. The values of such a scale are N (not adequate), M (partially adequate), L (largely adequate), H (fully adequate). The interpretation of the results of the Adequacy metric calculation self-explanatory. The Adequacy metric is defined to relate each other the relevance and the compliance of a quality sub-characteristic so that the highest values are obtained in the case of highly relevant and highly compliant quality sub-characteristic and the lowest ones are obtained in the case of highly relevant and low compliant. In Figure 3 the Adequacy metric calculation rule is described.

2.5 Phase 5: Reporting

The final step of the evaluation process consists of the release of an Evaluation Report. The Evaluation Report contains different views of the results obtained. A view is a way to represent the evaluation results with the suitable level of detail to target specific stakeholders.

In particular, the evaluation report shall contain as a minimum the following views:

- Managers view: it targets the decision makers of the organization, and it is focused on the risks for the organization related to the found weaknesses.
- Developers/maintainers view: it targets the technical staff. This view is focused on the provision of technical description of the causes of low ratings and related possible improvement actions.

Other possible views can be added to target, for instance, product users or potential customers.

The Evaluation Reports is a combination of metrics ratings and experts' judgment. In fact, it is expected to contain not only the rating associated to the relevant quality sub-characteristics, but also a part where the strengths and weaknesses found are described and possible improvement actions are identified.

The contents and structure of the Evaluation Report have been specified though the definition of a specific document template with the aim of assuring a complete and well-structured provision of evaluation results.

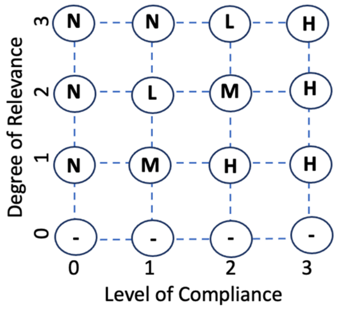


Figure 3: Adequacy metrics rating rules.

3 CASE STUDY

In this section we report the experience of the application of the QuESPro method in an industrial case study. The case study performed is complete enough to represent a sort of empirical validation of the method in terms of feasibility and fitness-for-use.

The rest of this section describes the organization, the software product involved in the case study and the outcomes of its evaluation.

3.1 Context Description

The case study for the application of the QuESPro method has been conducted in a retail corporation (the Sponsor of the evaluation) that operates in a specific geographical area and includes more than 150 points of sale (mainly grocery stores). Every point of sales shares a unique centralized logistic and administrative organization and relies on unique supply chain and services. The commercial strategy is based on the provision of unique assortments, products, and services for all points of sale.

3.2 The Software Product under Evaluation

The software product evaluated is dedicated to the accounting, purchasing and management of the active and passive cycle of the warehouse, with integrated logistics. The software product is developed to satisfy the need to optimally manage goods and therefore rotations and warehouse stocks for distribution companies.

The evaluated software product was first released more than ten years ago by an external software house that, so far, took care of its continuous update, customization, improvement and extension on the basis of the experience in the field and taking into account customers' demands both in terms of speed and simplicity and of control and security of the large volumes of information managed. Fast remote connection systems in IP technology (internet) are used by the points of sale to obtain information on the assortments and place orders for goods. The software product functionalities are shown in Figure 4.

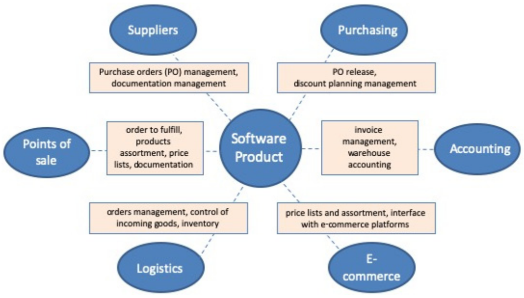

Suppliers
Purchasing
Purchase orders (PO) management, documentation management
PO release, discount planning management
Points of sale
order to fulfill, products assortment, price lists, documentation
Software Product
invoice management, warehouse accounting
Accounting
orders management, control of incoming goods, inventory
price lists and assortment, interface with e-commerce platforms
Logistics
E-commerce

Figure 4: Functionalities of the Software Product.

The architecture of the product is basically client/server, with the access to a Data Base managed by a QPS server. The client part allows the access to the Data Base by the stakeholders (point of sale, purchasing, accounting, …) through specialized forms. The product is developed in a Windows environment. The product's components are developed using different programming languages: Cobol, Visual Basic 6, C/C++ Windows. The data are stored on a DBMS using MariaDB.

A secure and confidential proprietary protocol, based on TCP/IP, is used for communication between the clients and the server.

The creation of new data views and queries is performed though a specific module (Form Creator), which uses an interpreted language (proprietary scripting) to create forms for the interactions with the Data Base. Figure 5 provides a synthetic representation of the product architecture.

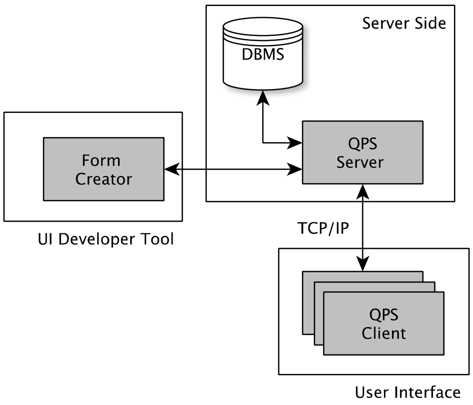


Figure 5: Architecture of the software product.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

3.3 Evaluation Purpose

The trigger of the case study was the need of the evaluation Sponsor to understand the extent to which the software product in use, described above, fulfils its current and future needs. As the network of groceries is going to grow in the next months, the Sponsor was interested in understanding whether the current one was still responding to the upcoming situation.

3.4 Application of QuESPro

In the following, the activities performed following the QuESPro process are briefly described along with the related outcomes.

Quality Model Definition Phase:This phase of the evaluation process has been conducted through an interview meeting with representatives of the sponsor organization that uses the software product, in order to understand the context of use, the current and future users' needs and the provided functionalities. The interview meeting took 1 day. As a result of the Quality Model definition phase, the sub-characteristics of the reference Quality Model have been prioritized by Relevance (as shown in Table 1 – third column).
According to the prioritization determined by means of the Relevance rating, only the quality sub-characteristics having the rating greater than 1 have been maintained in the quality model and have been evaluated afterwards. Consequently, the quality sub-characteristics Coexistence, Appropriateness recognizability, Learnability, Operability, User interface aesthetics, Accessibility, Reusability, Testability, and Replaceability have been excluded from the evaluation scope.

Data Gathering Phase: The data gathering phase has been conducted by releasing the questionnaire to both the Sponsor and the software house that developed the product. Moreover, two interview meetings involving the software development/maintenance leader and the representatives of the sponsor have been undertaken. The duration of each meeting has been 1 day. The evaluation team was composed of the authors of this paper. During the interview meetings with the development leaders the evaluators took the opportunity to observe the behaviour and the technical characteristics of the software product by means of real operational runs.

Rating of Sub-characteristics Phase: The rating of each sub-characteristic has been made by the evaluators on the basis of the data gathered (questionnaire, interviews, meeting notes). The determination of the rating is basically an expert judgment with the constraint that each downrating (lower than lower than 3) is required to be accompanied by an explicit argumentation and justification.

Table 1 contains, for each quality sub-characteristic, the assigned Level of Compliance rating along with the indication of the evidences used to determine the rating and possible clarification notes and the argumentations for the low ratings.

Metrics Calculation Phase: In Table 1 the rating determined by the calculation of the Adequacy metric is reported for each relevant quality sub-characteristics.

The sub-characteristics having an Adequacy rating equal to H or M are considered sufficiently achieved by the product evaluated. The others (Modularity, Analysability, Modifiability) are considered not sufficiently achieved.

Reporting Phase: The outcomes of the evaluation are described in the evaluation report issued to the evaluation sponsor. The results are presented by addressing each quality characteristic of the applied quality model. The only quality characteristic resulting weak is the Maintainability.

The weaknesses found are basically due to two main factors:

- Lack of documentation describing the software architecture. Information regarding the identification of the software elements and their interfaces and the specification of the features they implement is largely incomplete.
- Centralization of development and maintenance. The development and maintenance (corrective and evolutionary) are carried out by a single, very expert and skilled software engineer.

Although he has currently the full control and a deep knowledge of the product, there is a high risk for the continuity of system maintenance and future extensions.

The management view of the report identifies the risk for the organization. These risks are due to the availability of the unique developer/maintainer of the software product. In case of developer unavailability, the risks are:

- interruption of product functionality in case of failure;
- degradation of product performance in the event of changed operating conditions.

Table 1: Outcomes of the application of QuESPro.

ISO/IEC25010 quality characteristic	ISO/IEC 25010 quality sub-characteristic	Relevance rating	Level of compliance	Used Evidences	Adequacy Rating
Functional Suitability	Functional Completeness	3	3	Observation of runtime behaviour; Interviews	H
	Functional Correctness	3	3	Observation of runtime behaviour; Interviews	H
	Functional Appropriateness	3	3	Interviews	H
Performance Efficiency	Time-behavior	3	3	Observation of runtime behaviour;	H
	Resource utilization	2	3	Observation of runtime behaviour; Interviews	H
	Capacity	3	3	Interviews; Analysis of to working load data from previous years	H
Compatibility	Coexistence	1			
	Interoperability	3	3	Interviews; Questionnaire answers	M
Usability	Appropriateness recognizability	1			
	Learnability	1			
	Operability	1			
	User error protection	2	2	Observation of runtime behaviour;	M
	User interface aesthetics	1			
	Accessibility	1			
Reliability	Maturity	3	3	Interviews	H
	Availability	2	2	Interviews	M
	Fault tolerance	3	3	Questionnaire answers	H
	Recoverability	2	3	Interviews	H
Security	Confidentiality	2	2	Observation of runtime behaviour; Interviews	M
	Integrity	2	2	Interviews	M
	Non-repudiation	3	3	Run of specific test cases	H
	Accountability	3	3	Run of specific test cases	H
	Authenticity	2	2	Observation of runtime behaviour;	M
Maintainability	Modularity	3	2	Observation of runtime behaviour; Interviews	L
	Reusability	1			
	Analyzability	3	1	Source code walkthrough; Interviews	N
	Modifiability	3	2	Observation of runtime behaviour; Source code walkthrough; Interviews	L
	Testability	1			
Portability	Adaptability	2	2	Interviews	M
	Installability	2	3	Interviews	H
	Replaceability	1			

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Moreover, the risk of maintenance costs likely to move out of control due to the programmer's high bargaining power has been identified.

The Developers/maintainers view of the report contains the detailed description of the technical weaknesses found (e.g. the need of defining and applying a coding policy in order to make the code easier to be analyzed).

4 CONCLUSIONS

Since the early 80s, the software engineering community addressed the definition of schemes to characterize the quality of software and to evaluate it in a systematic and, possibly, quantitative manner. As a consequence, several techniques and methods have been defined for the evaluation of software products. Nevertheless, the application of them is often complex and expensive, thus not suitable for contexts where fast results are required, and limited resources are available. Often for such contexts the only way to evaluate software product quality is through informal expert judgement.

As a trade-off between expert judgment and complex evaluation techniques, we defined a lightweight method (called QuESPro) able to combine limited cost of application, and systematic and evidence-based evaluation of software products. In this paper, we described in detail the QuESPro method and we reported the outcomes of its application in an industrial case study. The QuESPro method is based on the quality model provided by the reference standard for software quality evaluation (the ISO/IEC 25010) and it is structured as a sequence of steps.

The outcomes of the case study showed that QuESPro not only the feasibility of the method but they highlighted several strengths:

- identification of precise improvement / risky areas;
- provision of quantitative measures suitable for possible benchmarking;
- possibility of tailoring/tuning according to the actual context of use and user needs,
- evaluation driven by a defined process,
- deployment of the method documented enough to be analysed and repeated,
- possibility to provide different detailed views of results for different roles of the organization.

The major improvement area identified is related to the lack of an automatic tool supporting and driving the application of the method. For this reason, we started the development of a specific tool to be used in the deployment of the QuESPro method.

REFERENCES

ISO/IEC 25010. (2011). Software Engineering: Software Product Quality Requirements and Evaluation (SQuaRE) Quality Model and guide. International Organization for Standardization, Geneva, Switzerland.

Miguel, J. P., Mauricio, D., & Rodríguez, G. (2014). A review of software quality models for the evaluation of software products. arXiv preprint arXiv:1412.2977.

Rodríguez, M., Oviedo, J. R., & Piattini, M. (2016). Evaluation of Software Product Functional Suitability: A Case Study. Software Quality Professional, 18(3)

Lee, M. C. (2014). Software quality factors and software quality metrics to enhance software quality assurance. British Journal of Applied Science & Technology, 4(21), 3069-3095.

Falco, M., & Robiolo, G. (2021). Product Quality Evaluation Method (PQEM): A Comprehensive Approach for the Software Product Life Cycle. In 7th International Conference on Software Engineering (SOFT).

Neri, H. R., & Travassos, G. H. (2018). Measuresoftgram: a future vision of software product quality. In Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (pp. 1-4).

Ortega, M., Pérez, M., & Rojas, T. (2003). Construction of a systemic quality model for evaluating a software product. Software Quality Journal, 11(3), 219-242.

Estdale, J., & Georgiadou, E. (2018). Applying the ISO/IEC 25010 quality models to software product. In European Conference on Software Process Improvement (pp. 492-503). Springer, Cham.

OMG (2013). Business Process Model and Notation. https://www.omg.org/spec/BPMN/2.0.2/PDF

Ouhbi, S., Idri, A., Aleman, J. L. F., & Toval, A. (2014). Evaluating software product quality: A systematic mapping study. In 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (pp. 141-151). IEEE.

Nakai, H., Tsuda, N., Honda, K., Washizaki, H., & Fukazawa, Y. (2016). A SQuaRE-based software quality evaluation framework and its case study. In 2016 IEEE Region 10 Conference (TENCON) (pp. 3704-3707). IEEE.