Identifying Quality Characteristic Interactions during Software Development

Gabriel Alberto García-Mireles[1], Mª Ángeles Moraga de la Rubia[2], Félix García[2] and Mario Piattini[2]
[1]*Departamento de Matemáticas, Universidad de Sonora, Blvrd. Rosales y Rodríguez s/n, Hermosillo, Sonora, Mexico*
[2]*Instituto de Tecnologías y Sistemas de Información, Universidad de Castilla-La Mancha, Ciudad Real, Spain*

Keywords: Software Quality Characteristics, Interaction between Quality Characteristics, Process for Monitoring Quality Characteristics Interactions, SQIMF Framework.

Abstract: Quality demands on current and future software systems need to address diverse quality characteristics considered important for the diversity of stakeholders. Dealing with software quality may require achieving a balance between relevant quality characteristics. Current literature shows that software organizations barely handle them. However, the lack of management of these interactions may be a causal factor in failed projects. In this paper, we present a process for monitoring interactions between quality requirements based on an interaction model of quality characteristics. This process is part of the SQIMF framework and supports the identification and characterization of the interactions between quality requirements. An exploratory case study was conducted in order to understand the factors that could influence the interactions that occur.

1 INTRODUCTION

In order to attain the expected quality in current software products and systems, software organizations should consider both the means used to provide a solution and the stakeholders' quality needs (Chung et al., 2000). The resulting quality requirements may constraints the design and implementation alternatives (Ameller et al., 2013). Dealing with quality requirements is a challenge, signifying that appropriate methods with which to support quality requirements throughout all the stages of the software development life cycle are needed, including approaches to deal with interactions between them (Ameller et al., 2013; Chen et al., 2013).

A particular issue concerning quality requirements that has arisen is how to deal with interactions between them and how to carry out the tradeoffs needed to establish appropriate target values (Loucopoulos et al., 2013). Before a trade-off study can be performed, interactions between the quality requirements under study should be characterized. Interactions have been described as situations in which the satisfaction of one requirement may affect the satisfaction of another (Robinson et al., 2003). For example, negative interactions between usability and security requirements could arise in software when implemented security components do not take into consideration either task scenarios or user skills in an operational environment. Interactions can be recognized by comparing requirements descriptions or analyzing their respective implementation (Robinson et al., 2003). In the quality requirements area, overlooking interactions is considered to be a causal factor in the failure of some projects (Boehm and In, 1996; Mairiza et al., 2010; Thakurta, 2013; Theofanos and Pfleeger, 2011). The lack of management of interactions between quality requirements can therefore increase software development costs, and could decrease stakeholder satisfaction (Chen et al., 2013; Dahlstedt and Persson, 2005).

Empirical studies on dealing with quality requirements interactions have shown that this issue is barely addressed in current industrial practices (Berntsson Svensson et al., 2012; Phillips et al., 2012). Although several methods with which to resolve conflicting interactions have been proposed (Barney et al., 2012), they are focused on supporting specific goals within a specific process such as those described in ISO/IEC 12207

García-Mireles G., Moraga de la Rubia M., García F. and Piattini M..
Identifying Quality Characteristic Interactions during Software Development.
DOI: 10.5220/0005351800780089
In *Proceedings of the 10th International Conference on Evaluation of Novel Approaches to Software Engineering* (ENASE-2015), pages 78-89
ISBN: 978-989-758-100-7

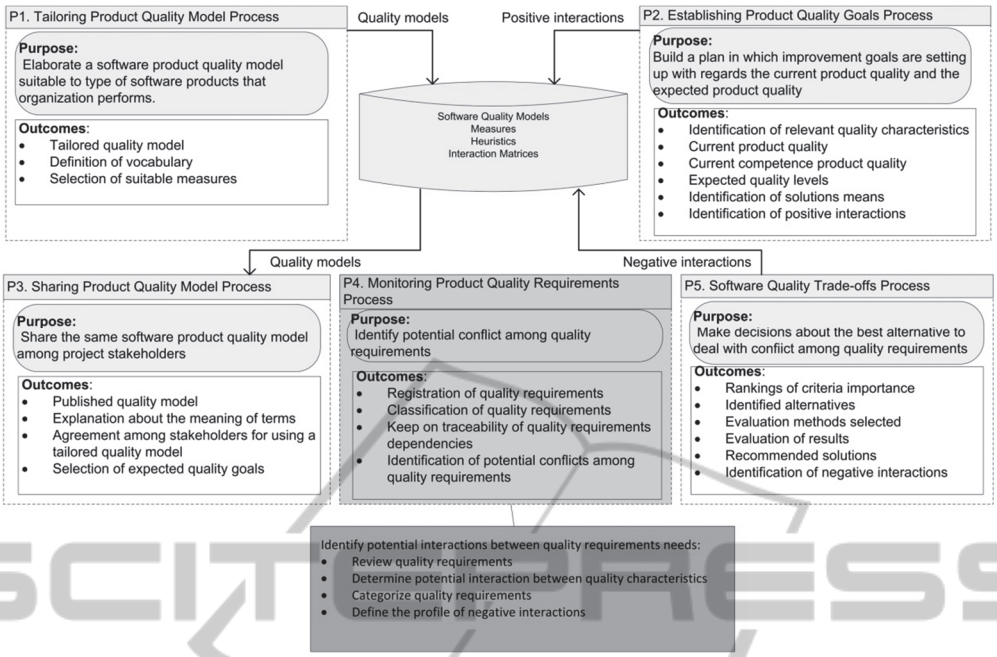

Figure 1: SQIMF framework.

(García-Mireles et al., 2014). Software organizations need appropriate process support if they are to identify and resolve conflicting interactions between quality requirements during all the stages of the software development life cycle.

One holistic approach that is considered useful as regards improving software product quality is based on the software process (Aaen et al., 2001). In order to establish a link between quality requirements and process reference models, a product quality model, such as ISO/IEC 9126 (ISO, 2001) or ISO/IEC 25010 (ISO, 2010), is necessary to identify practices with which to enhance software product quality. Current software process models are described at a high level of abstraction in order to support different kinds of software processes and they refer to quality attributes or quality characteristics (relevant quality terms used in this paper are depicted in Table 1). However, traditional process reference models, such as ISO/IEC 12207 (ISO, 2008) and CMMI (CMMI, 2010), lack the appropriate practices needed to address product quality characteristics (García-Mireles et al., 2012). Product quality evaluation in SPI initiatives barely addresses product quality as described in product quality models (Unterkalmsteiner et al., 2012).

Table 1: Definition of some quality-related terms.

Term	Definition
Quality requirement	A requirement that a quality attribute which is present in software (ISO, 2010)
Quality characteristic	Category of software quality attributes that have a bearing on software quality (ISO, 2010)
Quality model	Defined set of characteristics, and the relationships between them, that provides a framework in which to specify quality requirements and evaluate quality (ISO, 2010)
Target quality goals	A description of relevant quality characteristics and their respective expected values that an organization is attempting to attain in a software product
Interaction model	A matrix-based description of interactions between quality characteristics that shows the influences of one quality characteristic on the others
Attribute	Inherent property or characteristic of an entity that can be distinguished quantitatively or qualitatively by humans or by automated means (ISO, 25010)

In an effort to support software organizations that wish to deal with interactions between software quality characteristics, in a previous paper we proposed a process framework, called the Software Quality Interaction Management Framework (SQIMF), which can be used to manage interactions between quality characteristics (García-Mireles et al., 2013c). This framework describes only the

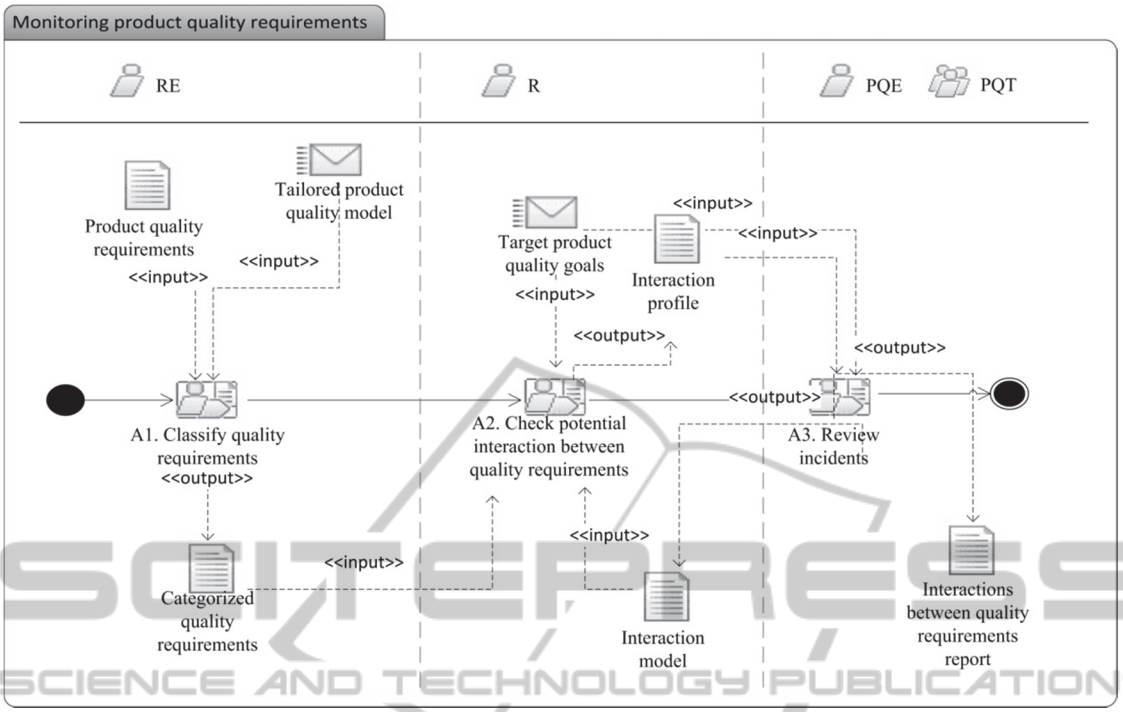

Figure 2: Activity diagram used to monitor product quality requirements process.

purpose and main outcomes of five suggested processes (Figure 1). Our goal in this paper is therefore to present the process employed to monitor interactions between quality requirements and the approach used to identify interactions between quality characteristics (P4 process in Figure 1). An exploratory case study was conducted in order to validate the activity needed to identify interactions between quality characteristics.

This paper is organized as follows: Section 2 describes some of the approaches with which to manage interactions between quality attributes/requirements/characteristics and presents an overview of the SQIMF framework. Section 3 provides a description of the process used to monitor interactions. Section 4 shows the case study design and the main outcomes obtained. Section 5 establishes the main findings of the exploratory case study and discusses threats to validity. Finally, Section 6 sets out our conclusions and discusses future work.

2 RELATED WORK

The software requirements field has actively studied interactions between requirements, which occur when one requirement places constraints on the design or coding options (Dahlstedt and Persson, 2005). The methods used to deal with interactions between requirements can be categorized as prioritization and negotiation approaches (Lehtola et al., 2004). Specific methods with which to deal with product quality requirements have also been suggested.

Several methods have been proposed to deal with product quality tradeoffs (Barney et al., 2012), which were reviewed in order to identify the target processes involved and the aims of each method. As a result, the methods reviewed were categorized as negotiation or prioritization approaches and the requirements needed to implement them were identified (García-Mireles et al., 2014).

Some methods are based on modelling approaches, such as the Non-Functional Requirements Framework (Chung et al., 2000) whose authors propose the Softgoal Interaction Graph in order to understand the effect of both quality goals and implementation means in product quality. Others propose quality tools and a software life cycle to deal with stakeholder quality needs in order to achieve a balance between quality characteristics (Boehm and In, 1996). Models based on ontologies are also proposed to identify potential interactions between quality characteristics (Al Balushi et al., 2013; Mairiza and Zowghi, 2010).

Addressing product quality characteristics in a

software project should consider appropriate practices that must be introduced into software process (Allen et al., 2006). Current process models lack systematic approaches with which to integrate methods that support product quality (Chiam et al., 2013). Software process literature pays little attention to product quality characteristics (Unterkalmsteiner et al., 2012). Indeed, Chiam et al. (2013) argued that there are no systematic approaches that can be used to represent and integrate methods that support product quality attributes within the current software process models. As the result of considering the aforementioned facts, in a previous work we presented the SQIMF framework whose objective is to manage interactions between quality characteristics (García-Mireles et al., 2013c).

The main goal of SQIMF is to manage interactions between quality characteristics which occur during software development and to provide support as regards enhancing a product quality characteristic based on process models or standards (García-Mireles et al., 2013c). The framework relies on a repository of tailored product quality models focused on usability, maintainability, and security (Figure 1). It also contains interaction models which describe the type of relationships that exist between quality characteristics. The initial content of these models (matrices) is based on a review of interactions between quality characteristics (García-Mireles et al., 2013a).

In order to provide the enhancement of a product quality characteristic with support, the SQIMF framework includes a set of methods whose objective is to carry out mappings between process models oriented toward a specific quality characteristic and generic process models. An example of the mapping of usability practices with ISO/IEC 12207 is presented in (García-Mireles et al., 2013b). The resulting process models can be used in software process improvement initiatives to enhance software product quality.

The process framework is composed of several processes, which can be implemented at both organizational and project level. At project level the processes address how to tailor product quality models and how to select appropriate quality goals and values. At project level, the processes describe how to keep the development team informed about product quality concepts, monitoring product quality requirements and resolving negative interactions through a trade-off study.

The processes in the SQIMF framework are presented at a high level of abstraction in García-Mireles et al. (2013c). This paper therefore describes the main components of the process used to identify and characterize interactions between quality requirements, denominated as monitoring product quality requirements.

3 PROCESS FOR MONITORING INTERACTION BETWEEN QUALITY REQUIREMENTS

The process is described with the SPEM 2.0 notation (OMG, 2008) and is also modelled in the EPF composer version 1.5 (https://eclipse.org/epf/) in order to ensure that the documentation is generated in a standard format. The purpose of the monitoring quality requirements process is to identify potential conflicts between quality requirements. The process relies on the quality interaction model which provides information about the potential interactions between quality characteristics. Conflicting interactions may appear when quality requirements interact with each other, particularly when they cannot achieve the expected quality values. Whatever the cause of issues between quality requirements, the interaction profile should be described for further analysis. As outcomes of the process, the quality requirements are redefined if this is necessary, the interaction model is updated with new interactions profiles and the report concerning the issues that have arisen is created. The report can support the update of product quality goals and further analysis to improve product quality goals. The activity diagram of this process is shown in Figure 2.

3.1 Process Objectives

The following objectives can be attained using the monitoring quality requirements process:

- A review of the consistency of quality requirements with target quality values.
- A verification of the potential interactions between quality requirements by means of the quality characteristics.
- An update of the appropriate interaction model using the interactions discovered.

3.2 Inputs and Outputs

The work products required in this process are: product quality requirements and related product components, a tailored product quality model, target

product quality goals, and an interaction model. The output artefacts are: a prioritized list of quality requirements, an interaction profile and a summative report containing the interactions found.

3.3 Roles

The roles participating in this process are presented in Table 2.

Table 2: Roles participating in the process.

Role name	Description
Product Quality Team (PQT)	A group of participants who have a diversity of quality interests in a particular software product. They can describe quality goals and apply appropriate methods to introduce and assess product quality
Product Quality Expert (PQE)	A participant who has the knowledge needed to adapt a product quality model in the context of organizational needs
Requirements Engineer (RE)	A participant responsible for eliciting, analyzing, specifying and validating requirements who can also categorize quality requirements using a product quality model
Reviewer (R)	A role responsible for detecting potential interactions between categorized quality requirements which can also create an interaction profile when a negative interaction is discovered

3.4 Activities

The process used to monitor product quality requirements consists of three main activities: classify quality requirements, check potential interaction between quality requirements and review incidents. The first two activities can be performed during the review points in an iterative development life-cycle, while it is suggested that the last activity be executed when the project is at the closing stage.

A1. Classify Quality Requirements Activity. The requirements engineer reviews the product quality requirements in order to classify them using a tailored product quality model. This activity can also be performed when product requirements are changed.

A2. Check Potential Interactions between Quality Requirements Activity. Once the quality requirements have been categorized, the categories identified are used to verify potential interactions between quality requirements in the context of a particular product component. If interactions are discovered in the comparison process, an interaction profile should be written to describe its characteristics.

A3. Review Incidents. The interaction profiles are analyzed in order to identify any problems in the software project, quality means, or interaction model that need to be updated. The review report is stored in the knowledge base. The basis employed to update the tailored product quality model and the target product quality goals document is the interaction between quality requirements report.

Table 3: Terms related to contextual factors.

Term	Description
Contextual factor	An aspect from the environment that influences either the way in which software is developed or the resulting software product
Contextual facet	A coherent set of contextual factors
Product facet	This includes contextual factors such as maturity, quality, size, system type, customization and programming language (Petersen and Wohlin, 2009).
Process facet	This describes the work-flow of the development. It includes activities, work-flows, and artifacts (Petersen and Wohlin, 2009)
People facet	This includes aspects related to project participants' skills and experience in addition to the assigned (project/organization) positions' jobs and roles
Organizational facet	This includes the organizational structure, organizational unit, certification, and distribution (Petersen and Wohlin, 2009)
Market facet	This represents the customers and competitors. The market facet includes number of customers, market segments, strategy and constraints (Petersen and Wohlin, 2009)

4 EXPLORATORY CASE STUDY

4.1 Case Study Design

An exploratory case study was carried out with the aim of understanding how interactions between quality characteristics are identified, including those factors considered relevant to characterize negative interactions. One small Spanish software company that had been certified as a testing laboratory with standard ISO/IEC 17025 participated in this study. Two people from this company participated in the interviews.

This firm was selected opportunistically on the basis of existing academic-industry relationships. The company provides consultancy and support for

process assessment in the ISO/IEC 15504 in addition to assessing product quality based on ISO/IEC 25010. The company's profile is thus appropriate as regards researching the interactions between quality requirements.

The case study consisted of two phases. The principal goal of the first phase was to identify a list of potential contextual factors that could have influenced the interactions that occurred between quality requirements. A list of factors was extracted from interactions between requirements described by Robinson et al. (2003) in addition to context facets (Petersen and Wohlin, 2009). The former reference provided factors to take into account when characterizing an interaction, while the latter reference provided factors that could be used to describe the facets of product, process, market, organizational and people. The contextual facets and factors are described in Table 3.

The list of factors allowed a template to be built in order to support the definition of the interaction profile. The template was used as a guideline for the semi-structured interviews. This template was checked by two researchers prior to its usage. Only minor details regarding the characterization of the interaction profile were found, and these were remedied by adding an appropriate explanation.

The two participants, a directive manager and a quality leader from the company under study, were informed of the aims of this inquiry. As a result, they agreed to the audio recording of the interview sessions and to filling in questionnaires and templates. Each interviewee had more than four years of experience in assessing software quality from the process and product perspective. The information provided is principally based on their experience as regards assessing software products.

During the interview sessions, several notes were taken as regards the guidelines used. These notes were verified with audio files. In addition, the type of interactions identified by both participants was explored in order to establish appropriate evidence for this research. Data collection thus relied on the interviewer's notes, audio files and the two participants' responses. The analysis of these sources allowed the triangulation of evidence with regard to the interactions identified by both participants and the contextual factors identified as being relevant.

In the second phase, a study of the monitoring quality requirements process was conducted (Figure 2). Since the process is designed to support the monitoring of quality requirements during a software project, it can be applied at different stages of the software product lifecycle. The exploratory case study was focused on the second activity, called A2. Check potential interaction between quality characteristics.

The second phase was conducted a week after the first phase. The goal was to determine the feasibility of implementing this process in a software organization. The instruments used to test the monitoring process were elaborated on the basis of the result of the first phase. This activity included the development of a guide explaining the purpose of the process, input and output products, activities and tasks to carry out and how to fill in the interaction profile template. A questionnaire was also elaborated in order to obtain information regarding the feasibility of using the process in an industrial setting.

4.2 Interviews Results

The interviewees found that it was relevant to deal with interactions between quality characteristics. At the time of the interview, this company was starting a new web-based project with the objective of supporting an organization whose main goal is providing visually impaired persons with web information content. Since they were defining the main features for this web-tool, the analysis of interactions between quality characteristics was focused on this project.

The interaction profile identifies the elements that participate in an interaction. For instance, for security and usability quality characteristics, the reviewer role found relevant interactions between quality sub-characteristics as depicted in Table 4.

Negative interactions, labelled with (-) are considered in the authenticity – operability and authenticity – accessibility pairs. The positive interaction labelled with (+) corresponds to the availability – user error protection pair. The remaining pairs are assessed with the label (O) since the interviewee did not find evidence to categorize them. The following paragraphs describe how the contextual facets contribute toward characterizing the negative interactions.

In the *product facet*, the product is in its conception stage in which the usability of the web application is the main quality characteristic under study. The organization plans to develop the product from scratch using traditional web technologies (e.g. HTML) but needs to reach an agreement with the customer as regards other programming language requirements. One of the main issues to be considered in the project is the screen sizes and the

Table 4: Interaction model filled in by an interviewee.

Product usability → / Security	Appropriateness recognozability	Learnability	Operability	User error protection	User interface aesthetics	Accesibility
Autenthicity	O	O	-	O	O	-
Confidentiality	O	O	O	O	O	O
Conformance	O	O	O	O	O	O
Attack detection	O	O	O	O	O	O
Availability	O	O	O	+	O	O
Integrity	O	O	O	O	O	O
Non-repudiation	O	O	O	O	O	O
Traceability	O	O	O	O	O	O

interaction mechanisms to be used (such as buttons). At software level, they are not considering special libraries to enhance the interaction of these kinds of users.

Upon considering the information regarding the *process facet*, one interviewee pointed out that the review activities, mainly after an increment is delivered, allow the occurrence of interactions between quality characteristics to be checked. An executable version of the web tool and the interactions with a potential user would be particularly useful to provide information about the effectiveness of the interaction mechanisms implemented. However, if interactions between quality characteristics occur, they do not have any practice, method or tool with which to support their management or resolution. In addition, they do not have training to identify interactions between quality characteristics, although the four members of the company have been involved in software maintainability for four years and have spent a year dealing with usability issues.

With regard to the *organizational facet*, this firm works by means of projects, and in this case the team size is four members. With regard to the *market facet*, there are no market segments or constraints that may influence the occurrence of interactions between quality characteristics.

When dealing with the interaction between quality characteristics, they reported negative interactions between accessibility and authenticity. These sub-characteristics belong to usability and security quality characteristics. They found it particularly difficult to provide access support for all types of users, including those suffering from blindness. As an argument they commented that “a common approach employed to register users in a web system is that of using CAPTCHAs, but they distort a label as regards differentiating between a real user and a bot.” However, this mechanism requires an in-depth study in the context of this web application owing to the profile of intended users.

Since this project is at its conception stage, it is difficult to establish an interaction profile for accessibility and authenticity. According to the data provided, the interaction between this pair of quality sub-characteristics only occurs when this group of users is taken into account. One of the main criteria for the interaction could be based on sharing information (resources) from a web application. The elements that can be considered are quality requirements and contextual factors related to the definition of a software product, including quality requirements for each type of target users. The quality requirements might be discussed with the customer using the quality characteristics described in a quality model. The organization is not currently addressing how to resolve the interaction, since it is in the first stage of the project.

The interviewees additionally highlighted positive interactions between a pair of quality characteristics. They reported that integrity (security) has a positive influence on user error protection (usability). The rationale for this relation is that the security mechanisms implemented ensure that only the user with modification access can change data records. This interaction relies on their previous experience in developing and assessing systems.

The product facet was that which was most relevant as regards identifying contextual factors, particularly the aspects regarding quality attributes, system type, supporting technology, and programming language. Other aspects such as product maturity, size or customization were not addressed by the interviewees. The process facet is, on the other hand, relevant as regards identifying the review session. This can be used to check whether the product increment meets requirements, including quality requirements. The approach used to identify potential issues with quality requirements, including those related to interactions between them, has until now been that of reviewing executable software. However, the interviewees did not have information about other practices, techniques and tools that can be used to identify and resolve conflicting interactions between quality characteristics.

The remaining contextual facets were irrelevant for the interviewee. Factors concerning the influence of people facet, organizational facet, or market facet on identification or causes of interactions between

quality characteristics were barely addressed. The explanation for this can be derived from the project profile, as it is in its conception stage and considers requirements from only one stakeholder.

The instrument also provides information about factors involved in defining an interaction: stakeholders' profiles, features required for each stakeholder, and the fact that their abilities influence the potential identification of interactions between quality characteristics. These interactions may be specific for certain features in the software system and they may affect only a certain group of users. However, the importance of the features involved in the interaction and the customer's priority as regards how to satisfy the stakeholders' needs can negatively influence the acceptability of a software product.

The second phase was focused on studying the monitoring of quality requirements process. The guidelines used to conduct the study include both the activity diagram and the interaction profile template which are used to test the suitability of the monitoring of product quality requirements process. The main part of this process is the characterization of the interaction between quality characteristics or sub-characteristics. A questionnaire was also prepared in order to assess the suitability of this process. The questionnaire was answered after the participant had identified the interactions and filled in the process template.

Table 5: Responses to diverse factors used to describe an interaction profile.

Factor	Response
Basis Which quality requirements are involved in an interaction? Do the contextual factors have an influence on a given quality requirement?	In this project, adding security requirements may have a negative influence on accessibility and operability since the software features are only available to certain types of users.
Criterion Which reasons are considered to lead to these interactions? The interactions occur owing to system structure, competence of the resources themselves, precedence of tasks, among others	New security components affect system structure.
Establish the degree of the interaction What is the scope of the interaction between quality requirements? What features, components or users' categories are involved?	Application type and target users might impact on the degree of interaction between quality characteristics. The stakeholder's experience in the security field may influence the quality of security requirements. The expert's knowledge can be used to establish a security mechanism to reduce the influence of highly secure mechanisms on accessibility or operability.
Probability of occurrence What is the probability of a conflicting interaction occurring?	The interactions occur during the software development under the constraints considered.
Impact of the interaction What is the effect of the interaction on the software project? For instance: Catastrophic, inconvenient, system failure, system reboot, unsatisfied users.	The main effect: Application does not meet basic quality requirements. Unsatisfied users and application cannot be delivered to target users.
Type of interaction What is the type of this interaction? It is a perceived interaction when it is described at requirements level. It is an implementation interaction when it is based on the analysis of implementation means	Perceived interaction.
Context What contextual factors influence the interaction between quality characteristics? For instance: stakeholders involved, users' goals, scenarios, events that trigger the interaction between quality characteristics, among others	Main contextual factors: Application users and application type.

With regard to the template for the interaction profile, the participant was clearly able to fill in identification data: project id, date, reviewer, type of software and artifacts analyzed. In this case, the participant assessed a desktop application, analyzing the code, executable software and technical documents.

The interaction model can serve to identify potential interactions between quality sub-characteristics. The participant reported a negative interaction between the pairs: security-operability, security – accessibility, and authenticity – accessibility. What is more, positive interactions were reported between integrity (security) and user error protection (usability).

The participant responded with a general statement for each interaction identified. He assessed basis, criterion, degree of interaction, probability of occurrence, impact of the interaction, type of interaction, and context. Each paraphrased response is presented in Table 5.

The questionnaire used to assess the feasibility of the monitoring of quality requirements process shows the following:

- The process objectives, roles descriptions, inputs and outputs are clearly described. The participant scored these four items with the agree option. The templates and the process task are described as being fair, and the participant scored these two items with the neutral option.
- The process objectives, descriptions of roles, inputs, outputs and templates are easy to understand, and the participant scored these four items with the agree option. Only the task category was scored with the neutral option.
- The process application in a potentially real situation produces different results. The template is the easiest element to apply, while the tasks were scored as neutral.

The objectives of the process are not, however, easy to apply (scored with the disagree option). In a note, the participant highlighted that there is currently a lack of suitable information about interactions between quality characteristics derived from quality assessment projects. One of the main applications for understanding interactions from assessment evidence is that of providing support as regards determining achievable quality goals or maximizing the relevant quality characteristics for a software product. In addition, the quality goals should be linked to specific practices in order to observe whether they really contribute toward resolving conflicting interactions.

5 RESULTS AND DISCUSSION

As a result of this validation of the monitoring of product quality requirements process, some evidence was obtained about the potential usefulness of the SQIMF framework. The benefits and issues are described in the following paragraphs.

The interaction matrix is useful for gathering data about the potential interaction between product quality characteristics. In the first phase of this exploratory case study, the interactions between quality sub-characteristics, which appertain to usability and security, were easily filled in. The interaction between usability and security is reported in literature (Mairiza et al., 2010), but few empirical studies address the interaction issues from a sub-characteristic level. This is therefore an important finding to be considered when dealing with interactions.

The product-related context facet, included in the interaction profile, influences the identification of interactions between quality characteristics. Relevant factors of this facet are both the requirements that should be considered in order to satisfy intended users' quality needs and the type of software product under development. This result is consistent with the literature concerning quality requirements that establishes that the types of applications influence the identification of relevant quality characteristics (Berntsson Svensson et al., 2012; Chen et al., 2013).

The process facet, included in the interaction profile, shows that the interviewees consider that the client influences how the software development can be performed. But they consider that the best time to analyze potential interactions between quality requirements is during the deployment stage, when the executing code can demonstrate the way in which different requirements were implemented. This task is therefore very relevant when deciding during which software development phases it is possible to use the SQIMF framework.

The remaining context facets seem to be irrelevant as regards identifying interactions between quality characteristics, since those interviewed did not provide any comments about their potential effect. This result can be explained by the stage at which this company is dealing with interactions between quality characteristics. Although its main focus is on quality assurance, the awareness of the effect of interactions between quality characteristics on software product and project resources is recent. In addition, the project under consideration is at its conception stage, and the quality requirements were

not therefore specified until the interview session.

The specific questions about how to identify interactions, their profile and how to resolve them showed that the main components needed to identify interactions are the stakeholders and the product. The interactions can be related to certain features to be implemented in the software in order to meet the quality requirements of specific stakeholders. However, the information provided depended on the perceptions of the interviewee since the project is at its conception stage.

Monitoring the product quality requirements process can support the identification of interactions, and can particularly serve to characterize the type of interaction identified. This knowledge can be used in the conception stage of a software project to negotiate with the customer with regard to the resources needed to address the requirements of all intended user groups. However, the specific tasks require additional refinement for their use in a software project. This refinement may include additional examples as regards using the process templates.

The monitoring product quality requirements process is defined to be used throughout all the stages of the software product life cycle and it can be adapted to the specific needs of the projects under study. The SQIMF framework requires quality requirements to be classified before an interaction model is used to identify potential interactions. The project under consideration was, however, at its conception stage, during which requirements have not yet been specified. This task was thus skipped, and only the activity denominated as "check potential interactions between quality characteristics" was performed. Indeed, this is the main task to be performed in the process. The outcomes can be considered valuable as regards improving this process.

The exploratory case study provides in-depth information about the interactions between quality characteristics. Although the guidelines of Runeson et al. (2012) were followed, there are limitations when a case study is conducted. As regards the external validity, the exploratory case study carried out cannot be generalized to other companies with a similar context to that of the participant company because this is an exploratory case. However, the characterization of the organization can provide useful insights into the development of a theory with which to define the type of interactions that can be identified at the conception stage of a software product and how these can impact on the project planning activities. In addition, the identified interactions between usability and security are consistent with current literature (Mairiza et al., 2010; Theofanos and Pfleeger, 2011).

In order to increase the reliability of the exploratory case study, both the artifacts used in the case study and the questionnaire were checked by two researchers. The study was conducted using a template approach, which is considered appropriate for research in software engineering owing to the ease with which it allows a clear chain of evidence to be obtained (Runeson et al., 2012). Moreover, the findings were discovered by using different sources, signifying that data triangulation (Runeson et al., 2012) was applied.

Since this is an exploratory case study, the investigation of causal relations is not one of the main activities. The case study was therefore conducted by following the recommendations and developing and checking the instruments used. With regard to construct validity, the exploratory case study was carried out in two phases. The potential contextual factors that might influence the interaction between quality characteristics were therefore commented on with the interviewee in the first phase. The guidelines used to monitor the quality requirements process additionally include descriptions to help users when filling them in. However, it was not possible to ask the participants about several process activities since the execution of process activities was performed by one participant alone.

6 CONCLUSIONS

In this paper a process that can be used to monitor interactions between quality requirements is proposed. The 'check potential interaction between quality requirements' activity was validated in an exploratory case study. The main findings of this research are described in terms of the factors that influence the identification and characterization of interactions between quality characteristics and the feasibility of using the process to monitor quality requirements in industrial settings.

In general terms, the identification of interactions between quality characteristics is performed at an abstract level. This view makes it difficult to address the characteristics of an interaction such as the degree of interaction and their potential impact on a software product or project. For instance, the main effect of a negative interaction, as reported by one interviewee, is that users are unsatisfied with the product. Monitoring interactions between quality

characteristics during software projects is therefore necessary in order to uncover the real impact on the software project. In addition, the identification of an interaction could be enhanced if the probability of occurrence and risk-based techniques are used.

The outcomes of the exploratory case study showed that the suggested process is appropriate for use in software organizations, although it needs to be adjusted to improve the suggested tasks. The template employed to describe the interaction profile was considered easy to use and apply in a software project.

The perspective used to identify the interactions of quality characteristics is related to the software project leaders/managers' experience. The organization under study, which is focused on product quality based on measures, considers that this approach should be included to assess interactions between quality characteristics. This view, although useful as regards establishing objective evidence about interactions, needs appropriate models with which to interpret the values obtained by each quality indicator. In addition, other perspectives with which to both improve product quality and minimize the impact of negative interactions should also consider appropriate process support.

Software organizations need appropriate support when managing interactions between quality characteristics. The monitoring quality requirements process provides partial support as regards determining the interaction profile. As future work, this process will be deployed during all stages of software development to check under what circumstances interactions occur. In addition, a software tool will be developed in order to support the activities in the process.

The monitoring product quality requirements process is focused solely on the identification of interactions between quality requirements by means of their respective quality characteristics, but it is also necessary to consider when and how to resolve negative interactions. The SQIMF profile provides a process with which to support decision making in order to balance quality characteristics. This process will therefore be used to study the approaches that software companies can implement to achieve a balance between product quality requirements.

ACKNOWLEDGEMENTS

This work has been funded by the VILMA and INGENIOSO projects (Consejería de Educación, Ciencia y Cultura - Junta de Comunidades de Castilla La Mancha) and Fondo Europeo de Desarrollo Regional FEDER, Ref.: PEII11-0316-2878 and Ref. PEII11-0025-9533) and GEODAS-BC project (TIN2012-37493-C03-01 funded by the Spanish Ministerio de Economía y Competitividad and by FEDER (Fondo Europeo de Desarrollo Regional).

REFERENCES

Aaen, I., Arent, J., Mathiassen, L., Ngwenyama, O., 2001. A Conceptual MAP of Software Process Improvement. Scandinavian Journal of Information Systems 13(1), 123-146.

Al Balushi, T.H., Sampaio, P.R.F., Loucopoulos, P., 2013. Eliciting and prioritizing quality requirements supported by ontologies: A case study using the ElicitO framework and tool. Expert Systems 30(2), 129-151.

Allen, J., Kitchenham, B., Konrad, M., 2006. Theme Q. The relationships between processes and product qualities, in: Forrester, E. (Ed.), A Process Research Framework. Software Engineering Institute, Carnegie Mellon, pp. 19-28.

Ameller, D., Ayala, C., Cabot, J., Franch, X., 2013. Non-functional requirements in architectural decision making. IEEE Software 30(2), 61-67.

Barney, S., Petersen, K., Svahnberg, M., Aurum, A., Barney, H., 2012. Software quality trade-offs: A systematic map. Information and Software Technology 54(7), 651-662.

Berntsson Svensson, R., Gorschek, T., Regnell, B., Torkar, R., Shahrokni, A., Feldt, R., 2012. Quality requirements in industrial practice-an extended interview study at eleven companies. IEEE Transactions on Software Engineering 38(4), 923-935.

Boehm, B., In, H., 1996. Identifying quality-requirement conflicts. IEEE Software 13(2), 25-35.

CMMI, P.T., 2010. CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033). Software Engineering Institute, Carnegie Mellon University Access: Nov. 2012 from http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm.

Chen, L., Babar, M.A., Nuseibeh, B., 2013. Characterizing architecturally significant requirements. IEEE Software 30(2), 38-45.

Chiam, Y.K., Staples, M., Ye, X., Zhu, L., 2013. Applying a selection method to choose Quality Attribute Techniques. Information and Software Technology 55(8), 1419-1436.

Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J., 2000. Non-functional requirements in software engineering. Kluwer Academic Publisher.

Dahlstedt, Å., Persson, A., 2005. Requirements Interdependencies: State of the Art and Future Challenges, in: Aurum, A., Wohlin, C. (Eds.),

Engineering and Managing Software Requirements. Springer Berlin Heidelberg, pp. 95-116.

García-Mireles, G.A., Moraga, M.Á., Garcia, F., Piattini, M., 2013a. Identificación de interacciones entre las características de calidad del software, in: Moreno, A., Vara, J.M. (Eds.), XVIII Jornadas de Ingeniería del Software y Bases de Datos JISBD2013, Universidad Complutense de Madrid, Madrid, España, pp. 141-154.

García-Mireles, G.A., Moraga, M.Á., Garcia, F., Piattini, M., 2013b. The Influence of Process Quality on Product Usability: A Systematic Review. CLEI Electronic Journal 16(2), 1-13 in: http://www.clei.org/cleiej/paper.php?id=278.

García-Mireles, G.A., Moraga, M.Á., Garcia, F., Piattini, M., 2014. Methods for supporting management of interactions between quality characteristics, in: Filipe, J., Maciaszek, L. (Eds.), 9th International Conference on Evaluation of Novel Approaches to Software Engineering INSTICC, Lisboa, pp. 93-100.

García-Mireles, G.A., Moraga, M.Á., García, F., Piattini, M., 2012. Towards the Harmonization of Process and Product Oriented Software Quality Approaches, in: Winkler, D., O'Connor, R., Messnarz, R. (Eds.), EuroSPI2012, CCIS 301. Springer Berlin Heidelberg, pp. 133-144.

García-Mireles, G.A., Moraga, M.Á., García, F., Piattini, M., 2013c. A framework to support quality trade-offs from a process-based perspective, in: McCaffery, F., O'Connor, R.V., Messnarz, R. (Eds.), EuroSPI2013,CCIS 364. Springer-Verlag Berling Heidelberg, pp. 96-107.

ISO, 2001. ISO/IEC 9126 Software Engineering - Product Quality," in Part 1 - Quality model.

ISO, 2008. ISO/IEC 12207 Systems and software engineering — Software life cycle processes.

ISO, 2010. ISO/IEC FCD 25010: Systems and software engineering - system and software product quality requirements and evaluation (SQauRE) - System and software quality models.

Lehtola, L., Kauppinen, M., Kujala, S., 2004. Requirements prioritization challenges in practice, in: Bomarius, F., Iida, H. (Eds.), 5th International Conference on Product Focused Software Process Improvement. Springer-Verlag Berlin, Kansai Sci City, Japan, pp. 497-508.

Loucopoulos, P., Sun, J., Zhao, L., Heidari, F., 2013. A systematic classification and analysis of NFRs, 19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime, Chicago, IL, USA, pp. 208-217.

Mairiza, D., Zowghi, D., 2010. An ontological framework to manage the relative conflicts between security and usability requirements, 3rd International Workshop on Managing Requirements Knowledge, MaRK'10, Sydney, Australia, pp. 1-6.

Mairiza, D., Zowghi, D., Nurmuliani, N., 2010. Towards a catalogue of conflicts among non-functional requirements, Fifth International Conference on Evaluation of Novel Approaches to Software Engineering, Athens; Greece, pp. 20-29.

OMG, 2008. Software & Systems Process Engineering Metamodel specification (SPEM) Version 2.0.

Petersen, K., Wohlin, C., 2009. Context in industrial software engineering research, in: IEEE (Ed.), 3rd International Symposium on Empirical Software Engineering and Measurement ESEM 2009. , Lake Buena Vista, FL, USA, pp. 401-404.

Phillips, L.B., Aurum, A., Svensson, R.B., 2012. Managing software quality requirements, in: Cortellessa, V., Muccini, H., Demirors, O. (Eds.), 38th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA) / 15th EUROMICRO Digital Systems Design (DSD) IEEE, Cesme, Turkey, pp. 349-356.

Robinson, W.N., Pawlowski, S.D., Volkov, V., 2003. Requirements Interaction Management. ACM Computing Surveys 35(2), 132-190.

Runeson, P., Höst, M., Rainer, A., Regnell, B., 2012. Case Study Research in Software Engineering: Guidelines and Examples. John Wiley and Sons.

Thakurta, R., 2013. A framework for prioritization of quality requirements for inclusion in a software project. Software Quality Journal 21(4), 573-597.

Theofanos, M.F., Pfleeger, S.L., 2011. Guest Editors' introduction: Shouldn't all security be usable? IEEE Security and Privacy 9(2), 12-17.

Unterkalmsteiner, M., Gorschek, T., Islam, A.K.M.M., Cheng, C.K., Permadi, R.B., Feldt, R., 2012. Evaluation and Measurement of Software Process Improvement— A Systematic Literature Review. IEEE Transactions on Software Engineering 38(2), 398-424.