



# Towards automatic evaluation of the Quality-in-Use in context-aware software systems

Sergio Salomón<sup>1</sup> · Rafael Duque<sup>2</sup> · José Luis Montaña<sup>2</sup> · Luis Tenés<sup>1</sup>

Received: 21 December 2020 / Accepted: 21 December 2021 / Published online: 5 February 2022  
© The Author(s) 2022

## Abstract

Context-aware systems adapt their services to the user's intentions and environment to improve the user experience. However, how to evaluate the quality of these systems in terms of user perception and context recognition is still an open problem. Our goal in this work is to evaluate the Quality-in-Use (QinU) for context-aware software systems according to the ISO/IEC 25010 standard and in an automated manner. This evaluation is oriented to be model-based, with domain specification and log data as input, while quality metrics and representations of users' behavior as output. In this process, we use probabilistic models to discover user patterns, heuristic metrics as QinU estimation, clustering techniques to obtain user profiles according to their QinU, and feature selection to identify relevant factors of context. We propose a framework for assessing the QinU in context-aware software systems called Framework for Assessing Quality-in-use of Software (FAQuiS). FAQuiS includes a set of models to represent all dimensions of context, a methodology to apply the quality analysis to any system, and a set of tools and metrics to support and automate the process. We seek to test the impact and ease of integration in the industry for this framework. A case study in a company allows us to validate the applicability in a real environment. We analyze the mechanisms that support the QinU evaluation in context-aware systems, the feasibility of the QinU quantification, and the suitability of the integration in companies. Compared to previous works, our proposal offers a novel data-driven approach with general-purpose and industrial viability. FAQuiS can be used as a solution to assess the QinU based on the ISO 25010 standard and the models of user behaviors in different contexts. This solution analyzes the context changes in the user interaction, can quantify the quality loss in these contexts, and does not require big efforts to be integrated into a software development process.

**Keywords** Quality-in-Use · Software quality assurance · Context modeling · Interaction analysis

## 1 Introduction

It has become a common reality that the users interact with software systems that react to their environment and try to respond to their needs at all times. These kinds of systems, known as Context-Aware Software Systems (CASSs), suggest the most appropriate actions for completing a task, make personalized recommendations based on time and geographic location, or show suitable labels for a photograph according to the elements present in it. The terms

of context and context awareness were introduced by Schilit and Theimer (1994) and later defined more broadly by Abowd et al. (1999), who describe the context as “any information that can be used to characterize the situation of an entity (a person, place, or object) that is considered relevant to the interaction between a user and an application, including the user and applications themselves”. Following this definition, we can consider the user itself, the application, and any other relevant entity to the interaction to be part of the context. In addition, a context-aware system is “a system that uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task” (Abowd et al. 1999). Thus, from the point of view of the user, these systems allow a more comfortable, agile, and useful interaction, sometimes without even require the explicit action of the user. Therefore, this interaction paradigm has a direct effect on the user experience of the system.

---

✉ Sergio Salomón  
ssalomong@axpe.com

<sup>1</sup> Axpe Consulting Cantabria SL, 39600 Camargo, Spain

<sup>2</sup> Department of Mathematics, Statistics and Computer Science, Universidad de Cantabria, 39005 Santander, Spain

In the last decades, the models of software quality have evolved greatly at the same time the software and technology changed significantly. From this evolution, three different perspectives can be identified: internal quality, external quality, and quality-in-use. First, the internal quality considers the static properties of the software, which only depend on software design and implementation. A few examples of these characteristics are software complexity, size, and modularity. Internal quality is the most commonly evaluated aspect during the development process. Next, the external quality considers software behavior in testing or production environments. Examples of external characteristics are the software performance in a specific device or the memory consumed by the application. Finally, the Quality-in-Use (QinU) is defined as “the user’s view of the quality of a system containing software, which is measured in terms of the result of using the software, rather than properties of the software itself” (Bevan 1999). While the internal and external perspectives assess the product quality, the QinU assesses the effect of the interaction between user and software and the user experience. Moreover, this perspective takes into account the adaptability to context as one of its characteristics. The QinU analysis is a challenge to be considered in the context-aware systems, as it can evaluate how the systems are providing relevant information related to the context and the user’s task.

In this research, we study the problem of evaluating automatically the QinU of context-aware software with multiple users and dynamic context. We propose the framework FAQuiS (Framework for Assessing Quality-in-use of Software) to assess the QinU of any CASS. Our objective is to automatize the QinU analysis, and for that this proposal only requires the essential input from experts to specify the domain knowledge of the use case. In our framework, we apply an approach where we model the different contexts to evaluate the user’s interaction in each one. First, we use probabilistic models to quantify the interaction process and deal with uncertainty. In order to estimate QinU, these techniques are combined with a set of heuristic metrics to assess the QinU characteristics defined in the ISO/IEC 25010 standard (ISO/IEC 25010 2011). Moreover, we support the analysis of a large number of users by using clustering techniques to obtain user profiles and look for possible causes of the QinU detriment by identifying the distinctive attributes and patterns of each profile.

For the proposed framework, we consider a validation study of its application in the software industry. This solution is oriented to be included in the software development cycle of a company, offering a generalizable QinU analysis integrated into the development without requiring big efforts. Thus, we apply our approach and validate the results in a real development process of the company Axpe Consulting, in order to analyze a case study. Axpe Consulting is a

multinational of software services and information technology and provides services in the sectors of software development, systems integration, and software quality outsourcing. Therefore, it offers a suitable environment where we can study the impact of our proposal.

The organization of this manuscript is as follows. Section 2 reviews the main contributions in quality evaluation of context-aware systems and process mining research. Section 3 introduces our framework for the QinU evaluation, the collection of models defined, the methodology to follow, and the tools implemented. Next, Sect. 4 shows the case of study where our framework is applied in a real development process in the industry. Section 5 discusses the contributions of this work and compares it with previous works. Finally, Sect. 6 presents the conclusions of our research.

## 2 Related work

Quality models are usually represented as hierarchical structures that contain characteristics and sub-characteristics of software. The ISO/IEC 25010 standard, known as System and Software Quality Requirements and Evaluation (SQuaRE), integrates two models: (i) a software quality model that is focused on internal and external properties, and (ii) a QinU model to measure “overall quality of the system in its operational environment for specific users, for carrying out specific tasks” (ISO/IEC 25010 2011). This QinU model includes five characteristics (effectiveness, efficiency, freedom from risk, satisfaction, and context coverage). However, SQuaRE is only a generic reference that does not specify how to assess these characteristics or how to carry out an evaluation process adapted to a specific type of system.

Customized QinU models have been derived from SQuaRE to provide specific support for evaluating non-context-aware software systems (Al-Nanih et al. 2009; Alnanih et al. 2013; Orehovalčki et al. 2013; Osman and Osman 2013; Souza-Pereira et al. 2021). These solutions are designed for particular applications or domains, such as Ambient Assisted Living systems (Erazo-Garzon et al. 2021), and are not intended to generalize to any system. Seffah et al. (2006) propose a unified hierarchical usability model composed of factors, criteria, and 127 specific metrics that can be selected to manually evaluate any system. Rauschenberger et al. (2013) present a tool to measure the user experience of interactive products through questionnaires. Hynninen et al. (2018) propose a proof-of-concept measurement of the product quality and QinU based directly on runtime metrics. Fogli and Guida (2018) define a methodology that involves experts who should inspect the system to rate a set of QinU metrics. Kim and Kim (2019) apply Analytic Network Process to assess the importance weights of quality attributes for general-purpose software. Furthermore,

the QinU evaluation can also be the result of applying data mining and machine learning algorithms. Rana and Staron (2015) propose a framework for quality (internal, external, and in-use) assessment using pattern recognition and classification to automatically infer high order quality characteristics from measurable attributes. This framework does not consider context characteristics. Alshareet et al. (2018) present an approach for QinU prediction using custom metrics extracted from projects documentation and neural networks to classify levels of QinU from these metrics. Many proposals based the QinU evaluation on users' reviews and opinion mining (Leopairote et al. 2013; Qian et al. 2016; Jiang et al. 2019; Atoum 2020). These approaches use natural language processing and sentiment analysis in order to map the comments onto QinU characteristics and value them by the polarity classification. Leopairote et al. (2013) build a quality ontology based on ISO 9126 for the mapping, while others use topic modeling (Qian et al. 2016; Atoum 2020) or feature words (Jiang et al. 2019) instead.

Regarding the dimension of context, Sousa Santos et al. (2017) carry out a review of test case design techniques for CASSs. This review concludes that the proposals focus on the following software quality characteristics of SQuaRE: functional suitability, compatibility, portability, usability, performance efficiency, and reliability. Therefore, these proposals lack an assessment of QinU and they focus on concepts of the external or/and internal quality of the system. Moreover, Carvalho et al. (2017) point out that SQuaRE does not contemplate five software characteristics (context awareness, mobility, calmness, transparency, and attention) that are appropriate for a ubiquitous system. Also, Erazo-Garzon et al. (2020) review the quality assessment in Ambient Assisted Living systems and identify a research opportunity in context coverage analysis and the need to deepen the research of QinU measurement. Capdevila et al. (2021) highlight that the emergence of new interactions and interactive paradigms (e.g., voice, augmented reality, Internet of Things) requires new methodologies for QinU evaluation. Ben Ayed et al. (2016) carry out a study that verifies the impact of the context in the QinU of a mobile application. This study considers three elements to represent the context (user profile, physical environment, platform) and quantifies metrics for the characteristics of SQuaRE in each context. Augusto et al. (2019) describe a methodological approach to consider the context to derive the scenarios in which the system should be tested. The techniques that have been applied for context modeling include UML activity diagrams (Mirza and Khan 2018), formal methods (Djoudi et al. 2016), based on bi-graphical reactive systems (Cherfia et al. 2017) and context-aware flow graphs (Lu et al. 2006).

In a preliminary study of our approach (Salomón et al. 2019), we propose a method to model user behavior in these dynamic contexts by processing log data. Thus, the

analyst becomes aware of the interactions that users perform in different contexts, the user patterns that emerged during the interaction process, and how the system features are exploited. The motivation for generating models from log traces to learn user behavior representations originated in the Process Mining field (van der Aalst 2012). These approaches apply mining (van der Aalst et al. 2004), evolutionary (van der Aalst et al. 2005), heuristic (Weijters and Ribeiro 2011), and Markovian (Gadler et al. 2017) methods to discover directed graphs with nodes representing actions and arcs that define temporal or causal relationships between actions. The process mining activities are classified (van der Aalst 2012) as process discovery (producing a model from event logs), conformance checking (comparing a model with reality), or process enhancement (improving an existing model). For instance, by applying these techniques, process models can be discovered to analyze behaviors and check compliance according to prescriptive models. While it is not the usual goal of this field, some of these methods have been exploited in studies related to software quality or context awareness. Caron et al. (2013) present an approach to detect compliance failures in business processes and assess potential risks. Fernandez et al. (2009) and de Medeiros et al. (2004) have applied these techniques to model user activity in dynamic contexts related to CASSs. Also, these models are useful in the requirements engineering field to find unexpected user behaviors, which implies the elicitation of new requirements that allow adapting the system to new processes and users' goals (Ghasemi and Amyot 2020). To the best of our knowledge, there are no studies using process mining methods to analyze the QinU specifically.

The review of the previous works allows us to affirm that the proposal of specific support for QinU assurance in a CASS will fill a gap in the current research contributions. This conclusion is a consequence of identifying the following limitations in the aforementioned works:

1. Assessments of QinU have been carried out automatically in non-context-aware software systems. The assessments are mainly based on algorithms that analyze the comments of the users and therefore rely on explicit opinions from users. Additionally, these approaches do not consider complete user behavior representations whose analysis provides holistic assessments of their activities in different contexts.
2. Many approaches of the QinU evaluation are focused on specific software systems and apply custom quality models that are not generalizable to other cases.
3. Methods for evaluating CASSs are based on tests related to the internal and external quality of systems. These proposals are limited to the verification of test output, but there is a lack of assessments in relation to the effects they have on the user experience (the QinU perspective).

- Process mining techniques have been used to represent user behavior in contexts that change dynamically. These models have been exploited to identify user behaviors, but they have not been mapped to QinU characteristics.

The aforementioned limitations justify and motivate our work to build a framework that provides support for QinU assurance tasks. The support of this framework is aimed at managing context and user behavior models that will be analyzed to calculate metrics that quantify the QinU characteristics indicated by the SQuaRE standard. These low-level metrics will be used to define user profiles with similar QinU estimations in order to provide higher-level analysis.

### 3 Framework

In this work, we propose a framework for the assessment of the quality-in-use of CASSs. We name the framework **FAQuiS**: “**F**ramework for **A**ssessing **Q**uality-in-use of **S**oftware”. FAQuiS aims to provide an environment for the QinU analysis and assurance to software development teams with an automatic approach. In this setting, the evaluators, developers or quality analysts, identify the relevant context factors of a CASS, establish the QinU requirements, while the tools of the framework assist and performed an automated analysis of the QinU from log data and system specifications of the CASS.

FAQuiS is formed by: (i) a set of heuristic metrics to estimate QinU characteristics; (ii) a set of metamodels to define the models of user, tasks, and environment; (iii) a methodology to apply the evaluation process to a particular CASS; (iv) a set of tools to generate the specifications, process the data and estimate the Quality-in-Use of the system. All of the components are represented in Fig. 1. The following sections described each of these components in more depth.

### 3.1 Quality-in-Use metrics

Our proposal is based on the ISO 25010 model (ISO/IEC 25010 2011), represented in Fig. 2. This standard quality model defines the concept of Quality-in-Use by means of five characteristics: (i) **effectiveness**, as the accuracy and completeness with which users achieve specified goals; (ii) **efficiency**, defined by the resources used in relation to the accuracy and completeness with which users achieve goals; (iii) **freedom from risk**, as the user perception of how much a product or system mitigates the potential risk to economic status, human life, health, or the environment; (iv) **satisfaction**, for the degree to which user needs are satisfied when a product or system is used in a specified context of use and by means of the sub-characteristics of comfort, pleasure, trust, and usefulness; (v) **context coverage**, by the degree to which a product or system can be used with effectiveness,

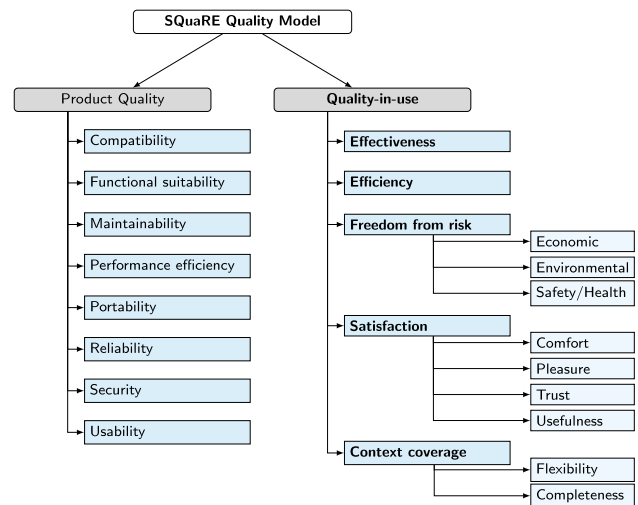
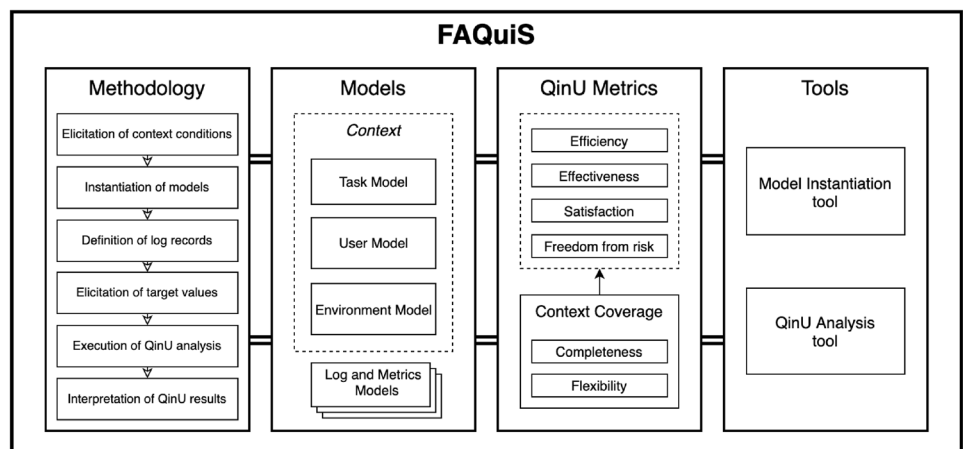


Fig. 2 Product quality and Quality-in-Use characteristics in the SQuaRE model

Fig. 1 All elements that composed the framework FAQuiS



efficiency, freedom from risk, and satisfaction in both specified contexts of use (context completeness) and contexts beyond those initially explicitly identified (flexibility).

These characteristics and sub-characteristics are contemplated in the quality analysis of FAQuiS through the heuristic estimation of multiple metrics. The designed metrics are an interpretation based on the description of the characteristics included in the ISO 25010 model. For each of the characteristics, many features are measured automatically from the interaction data and models. First, the **effectiveness**, since it is related to the accuracy of goals completion, is measured by the number of key tasks finished and the amount of concordance between user behaviors and behaviors expected by developers. Next, the **efficiency**, as it is associated with productivity and resources used, is considered as the amount of time completing tasks, as well as the number of actions and interactive spaces needed. The **freedom from risk** is represented by the frequency and impact of user interactions that imply some risk for the user (if any), split into economic, safety, or environmental types. The **satisfaction** is estimated through the sub-characteristics considered in the standard: (i) comfort, by indicators of ease, such as repetition of actions and system customization; (ii) pleasure, by indicators of personal needs fulfillment (acquiring new skills and providing personal identity) and tendency to continue the interaction with the system (in different sessions, with user input); (iii) trust, by proof of confidence in the system and its intended behavior, such as usage of all available tools, repetition of risk interactions, and lack of delay in the execution; (iv) usefulness, by the amount of activity (tasks, actions, spaces) in the user interaction, as well as the number of successful responses from the system. Similarly, the **context coverage** is estimated through its sub-characteristics: (i) context completeness, measuring all previous metrics for each intended context of use (if any); (ii) flexibility, measuring all previous metrics for unexpected or unidentified contexts of use (if any).

In addition, the chosen metrics can address other characteristics suitable for ubiquitous systems mentioned in Sect. 2 (context awareness, mobility, transparency), as they are computed for all dynamic contexts (context coverage) or take into account the implicit responses by the system. For more detailed description about all metrics used, see Table 4.

### 3.2 Models and metamodels

We use a set of models to represent the context of the interaction in a CASS. According to the definition of context of use, we can say that the context is formed by the user, the tasks, the technological equipment (hardware, software, and materials), and the physical and organizational/social environment in which a product is used. Based on this definition, we consider a task model to represent all actions supported

by the system, a user model for the user traits and skills, and an environment model composed of the other aspects that affect the interaction: the physical context, the organizational context, and the technological context. All these models represent entities that influence the process of interaction in CASSs. By using these models, we obtain a machine-readable specification of the CASS domain information, encapsulate the factors of each entity in the interaction, and interpret data from the log repositories.

FAQuiS process the previous models for analyzing the user interaction by its context. These models require a meta-model to be instantiated from, which will define some of the generic attributes of any CASS: the actions and tasks in the system, the traits of the users' community, and the conditions of the context. The metamodels allow generalizing this framework to any particular application as they establish common templates for each model that can be adapted to the requirements of a CASS. We show in Fig. 3 a simplification of the context metamodel, including the elements and relations of all models. Each of the models is detailed along with its key elements in the following sections.

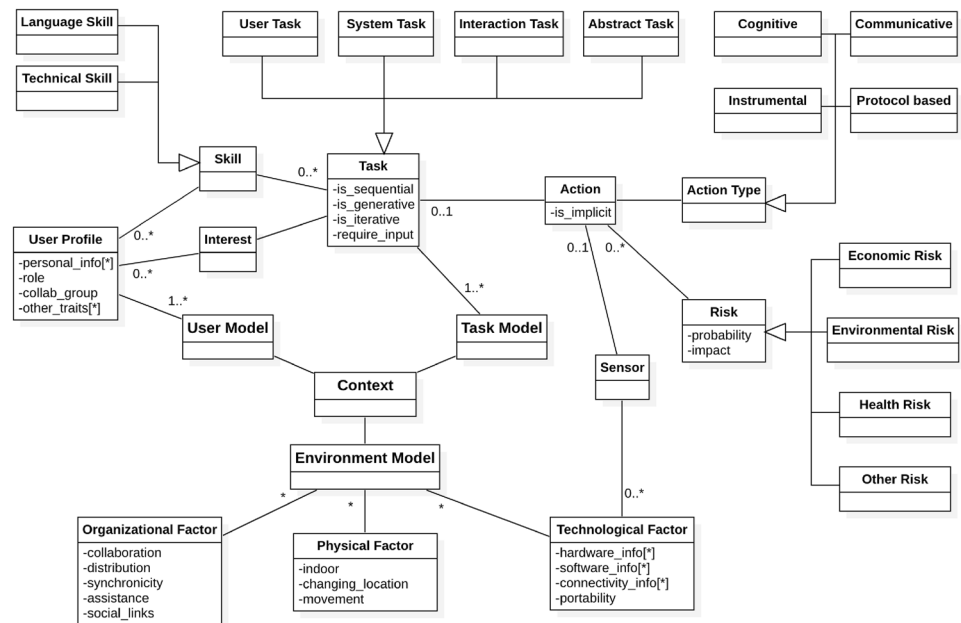
#### 3.2.1 Task model

We use a task model in our framework to analyze the activity flow and to represent the different actions supported by the CASS. Task analysis is a technique widely used in the field of Human-Computer Interaction to model user behavior. The models are hierarchical structures that decompose user tasks into sequences of small activities. These structures do not include elements of the user interface that supports the actions. Moreover, the models are designed before building the user interface. They are useful to know how users carry out their tasks, but they do not model any implementation details of the system.

Following this direction, we propose a task metamodel that allows different abstraction levels and task composition by means of a hierarchical tree structure. In the tree structure, the root represents the system itself, and children nodes represent activities or subtasks of the parent node that can be related by sequential constraints or choice constraints. The leaf nodes (the most concrete level) represent the actual interactions in the system, the ones that should be recorded in the log data. These interactions can be regular explicit actions from the user, or also implicit ones. An implicit human-machine interaction is *“an action, performed by the user that is not primarily aimed to interact with a computerized system but which such a system understands as input”* (Schmidt 1999).

In the task model, each task (a node in the tree structure) can indicate or include different elements such as its task type, relation to user interests, if it needs certain skills or user inputs, and if it can be executed repeatedly.

**Fig. 3** Context metamodel of FAQuiS



- **Task type:** Classification according to the actor and the function of the task. The types considered are (i) user tasks, problem-solving, comparing, planning tasks that are carried out by the user only; (ii) system tasks, alert, feedback, comparison, grouping, visualize, locate, or overview tasks that are executed by the system; (iii) interaction task, tasks of selection, edit, control, monitoring, response, or configuration where the user and the system interact; (iv) abstract tasks, other tasks where neither the system nor the user are involved.
- **Skills:** Abilities required to complete the task successfully. These could be language skills, technical skills, or other skills (for example, skills related to the particular domain of the task or system).
- **Required user input:** If the task required some additional input from the user (e.g., text or numeric information introduced through a field, multiple selections in a form).
- **Task relation constraints:** Existence of dependencies between subtasks under the same global (parent) task. In this case, they could be present a sequential relation (i.e., expected patterns in the interaction) or could belong to a process of choice (i.e., options for the user).
- **Iterative:** If the task can be performed repeatedly or recurrently.
- **Action type:** Classification of the actions according to the obtained result and the use of a shared context (collaboration or communication). The types considered (Duque et al. 2013) are (i) communicative actions, when users exchange messages with each other; (ii) protocol-based actions, when users access common spaces or coordinate tasks; (iii) instrumental actions, when users modify an artifact in a common space; (iv) and cognitive actions, when users utilize an element of a space without modifying it.
- **Explicit/implicit:** The action might require that the user interact explicitly, or on the opposite, it may be captured implicitly (e.g., geolocation captured implicitly while the user travels).
- **Risk information:** Existence of risk in the execution of the actions, as well as the type of the risk (health, environmental, economic, or other) and the values of impact and probability (or frequency) of that risk.
- **Sensors:** Devices that are used in this action to obtain information about the environment (e.g., gyroscope, accelerometer, GPS).

On the other hand, each concrete action (a leaf node in the structure) can include additional information features such as associated sensors, risk factors of the action (health, environmental or economical), and purpose in a collaborative setting.

FAQuiS uses this model to process the task, goals, and activities structure of the system. Later, this can be exploited with every user to identify how the system is used, the task completion exhibited and patterns followed in their interaction. This model will be unique (instantiated once by the evaluator) as it represents the task design of the whole system.

### 3.2.2 User model

In FAQuiS, the user model specifies the types of users in the system according to their different traits. By means of this model, the evaluator specifies features such as groups of users by their interests, skills, roles, or any other characteristic included in the user metamodel that may be relevant in the use of a system.

- **Personal data:** Personal information that could be studied in the quality analysis (e.g., age group, gender, nationality).
- **Interests:** Information related to the preferences or hobbies of the user.
- **Skills:** Special knowledge or abilities that may influence the interaction with the system (e.g., language skills, technical skills, skills related to the system domain).
- **Role in the system:** The user class when the system assigns different functions to each user (e.g., administrator, moderator, content editor, regular).

This set of attributes has the power to describe significant profiles within the users' community of the system. Furthermore, as this kind of profile can group multiple particular users identified by these traits, it works as a representation of the aggregated users' data. Thus, the model allows the framework to identify significant changes in the QinU evaluation according to the user type or the skills required to complete a particular task. This way, it is possible to handle many users (for example, in Big Data cases) in a summarized structure and avoid the use of sensitive data of particular users in the quality analysis.

### 3.2.3 Environment model

Finally, we design an environment metamodel to include the different contextual aspects that surround the interaction of the user. The model is divided into three different parts of the environment information and constraints: physical context, organizational context, and technological context.

First, the physical context is related to the location where the interaction takes place. This context is essential to consider features such as how the location and user mobility influence the interaction, if the system adapts to it, and if the quality is homogeneous regardless of location changes.

- **Location:** Geographic place where the interaction happens that may change over time.
- **Outdoor/indoor:** If the interaction takes place in a static closed space or outside.
- **Movement:** State of motion of the user while the interaction happens.

Next, the organizational context defines the interaction conditions related to other agents. The objective is to characterize shared contexts where another person can interact with the user. This aspect covers social links and collaboration between users, or assistance when using the system. These interactions may involve communication with the user or even modify a space in the system itself. The Computer-Supported Collaborative Work (CSCW) research scope generally characterizes systems with functionalities for cooperation between users through two dimensions: space and time (Penichet et al. 2007). The features provided by the system must determine if the shared context between users occurs in the same place or people are geographically distributed (this is the space dimension) and if they interact simultaneously or asynchronously (time dimension).

- **Social links:** Existence of relations or links between users through the system.
- **Collaboration:** Context of groups of users that collaborate to achieve a common goal.
- **Assistance:** Availability of assistance when the user interacts with the system.
- **Distribution:** If the users interact in the same space (e.g., building, office, class) or they are geographically distributed.
- **Synchronicity:** If the interaction happens simultaneously or separates enough over time (asynchronously).

Last, the technological context represents the key features of hardware, connectivity, portability of the device used, and software information.

- **Hardware:** Features of the devices used to interact with the system (e.g., desktop computer, mobile device, computer attributes).
- **Software:** Features related to the Software that could change the interaction (e.g., Software version, web platforms, and desktop platforms).
- **Sensor:** Device that detects changes in the environment (e.g., GPS, accelerometer, camera).
- **Connectivity:** Network features used when the interaction takes place (e.g., Wi-Fi signal, connection issues, connection speed).
- **Portability:** If the device allows mobility when it is used.

Through this model, FAQuiS can consider the different contexts appearing during the normal interaction of the users. This is used to recognize the conditions that influence the interaction over time. Therefore, the framework can identify relevant differences in the QinU by the conditions of the interaction when the quality analysis is executed.

### 3.3 Methodology

We describe the methodology to apply FAQuiS to evaluate any CASS. This methodology involves a set of activities to be performed in order to adapt the target system for the assessment. The developers or evaluators should conduct these activities in order to identify the particular conditions of the system, generate admissible interactions records, and define requirements for the QinU metrics. These activities encode the domain knowledge and requirements, generating the expert input needed for the automation processes. The framework tools assist in some of the activities and carried out the automated analysis, providing the QinU results that evaluators can use for decision-making. Table 5 summarizes all elements in the proposed methodology: activities, tasks, input and output of each task, and framework tools involved. The activities of the methodology are detailed in the following sections.

#### 3.3.1 Elicitation of the context conditions

The first step of the methodology is to identify the particular aspects or circumstances of the context that are, or could be, relevant for the interaction with the system. We as evaluators need to establish the essential and influential conditions of the environment in the CASS that will be evaluated, since irrelevant factors will increase the complexity to interpret the QinU analysis later on. Following the aspects of *Context of use* that are specified in ISO 9241 (International Organization for Standardization 2010; Maguire 2001), we consider the information of the environment model that will play an important role in users' interaction to define the context. For example, the conditions of physical context should be considered if the CASS exploits the user geolocation, implicitly or explicitly; an analogous case for the organizational context if there is collaboration or distribution between users. On the other hand, conditions that are not pertinent or unrelated should be omitted. In addition, we should consider the users' characteristics and the system itself (tasks and goals) as well. The factors considered should be aligned with the metamodels introduced in Sect. 3.2. From this activity, the output will be a set of context requirements needed to instantiate the metamodels.

This activity can be divided into the following tasks:

1. Identify all the tasks and actions supported by the system.
2. Identify the relevant user characteristics (age, gender, skills, roles).
3. Identify the existence of relevant conditions in the technological context (hardware, devices, connectivity, software versions).

4. Identify the existence of relevant conditions in the physical context (use outdoors, dynamic changes over time).
5. Identify the existence of relevant conditions in the organizational context (social links, collaboration, synchronicity, assistance).

#### 3.3.2 Instantiation of the models of context

Once the context conditions are identified, the models of context should be instantiated following the respective metamodels (user, tasks, environment). For this activity, we can use the model instantiation tool (Sect. 3.4), as well as the document of context requirements generated in the previous activity, in order to indicate all factors elicited earlier. The model instantiation tool will provide a graphical representation of the metamodel, as a form, that can be filled out. In this step, multiple factors of the environment can be specified (like multiple contexts present in the interaction) along with the identifier that will represent this set in the log (see next task). Some aspects of the environment model (physical, organizational and technological) could be unused if none of their factors are applicable. Also, multiple user profiles with different traits can be described by the user model. On the other hand, only one task model must be instantiated (as the tree structure).

As a result of the activity, the evaluator defines the models for tasks, users, and the environment with all the dimensions relevant for the particular CASS. Thus, some dimensions can be omitted if they do not play an important role in the system (e.g., if the system is not influenced by the location of the user). The main goal is to generate the most complete representation of the context possible to achieve an accurate QinU analysis.

#### 3.3.3 Definition or adaptation of the log records

We must consider the operation of log generation in the design (or adaptation) of the software system in order to collect the data necessary for the QinU analysis. Therefore, it is required to decide what variables will be captured in the CASS that are influential for the assessment. Then, the structure of the event log has to be defined with the expected format. In this case, some variables are essential to apply FAQuiS: a user identifier, a task identifier, and the timestamp of the interaction. More variables can be included as well to recognize relevant sets of context states in the interaction. The variable types for the log records are shown in Table 1.

For this activity, we use the models instances to identify the variables of context factors that must appear in the event log. Once it is established the log structure, we should include the respective software requirements to allow the log generation in the CASS. The software should be designed



**Table 1** Different fields that can be specified in the model of log

Variable	Description
User	Identifier of the user of the interaction event
Date-time	Timestamp of the moment of the interaction event
Action	Identifier of the action performed by the interaction event
Context factor	Identifier of a context factor from the physical, organizational or technological dimension present in the interaction event

and implemented, or adapted in case of an already built system, considering these new requirements. In addition, we can use the model instantiation tool (Sect. 3.4) to create a **log model** to allow processing the log data. After completing the activity, the software system generates log records that are essential for the assessment in FAQuiS, and a log model is instantiated to identify the relevant and required variables.

1. Selection of variables captured in the logs.
2. Design and adaptation of the system to generate the logs.
3. Instantiation of the log model.

### 3.3.4 Elicitation of the target QinU indicators

In order to understand the goodness of the QinU analysis, the optimal QinU measures should be elicited. This set of values, defined by the developer or evaluator, will form an ideal case of the QinU metrics (Sect. 3.1) to be compared with the real cases extracted from the data. To this aim, FAQuiS provides, through the instantiation tool, a metrics model to deactivate and select the suitable metrics and to describe this ideal case (Sect. 3.4). We establish the optimal values according to the context conditions of the initial activity and the goals chosen for the users in the system. This will allow the framework to evaluate the QinU metrics obtained in the analysis according to the divergence from these target values. The resulting specification of the optimal indicators will be further processed by the assessment tool. The metrics not limited by optimal values will be evaluated according to the range extracted from log data.

In this step, we also must specify the minimum requirements of QinU. These criteria have to be defined for each of the QinU characteristics: effectiveness, efficiency, satisfaction, freedom from risk, and context coverage. The minimum fulfillment is selected for each characteristic depending on the software requirements of the particular system. Once the QinU analysis is executed, the automatic comparison between the metrics obtained and the ideal case (defined in the metrics model) computes the degree of fulfillment of each QinU metric. Using the minimum requirements defined in this activity, we can verify the QinU characteristics that do not meet our earliest requirements.

1. Select the suitable QinU metrics.
2. Elicitation of the optimal values for the metrics.
3. Specification of minimum requirements of QinU characteristics.

### 3.3.5 Execution of the QinU analysis

In this step, the evaluator executes the QinU analysis using the assessment tool (Sect. 3.4) of FAQuiS. Here, the following inputs should be provided: the log data of the CASS, the log model created and all models of context instantiated. The assessment tool will process these inputs to carry out the automatic procedures: computing the needed QinU metrics, identifying the different user profiles, and extracting patterns of user behavior. The tool generates the results of the analysis as output. These results present QinU estimated values and their associations with profiles of behavior or context features. Also, the tool creates charts automatically adapted to the data and results dimension.

### 3.3.6 Interpretation of the QinU analysis

As the final activity of the methodology, we must interpret the results of the QinU analysis extracted by the assessment tool of FAQuiS. In this case, the tool provides: (i) numeric results and charts of the QinU metrics for each user profile, and the divergence to the ideal case; (ii) relevant context conditions in each profile; (iii) state-machine representations of action and context changes for each profile.

From these outputs, we must evaluate the causes of the QinU problems that the CASS exhibit. The first output shows us the fulfillment of our requirements of QinU in relation to the ones met by the system, the quantification of each QinU characteristic, and the differences of QinU indicators between each profile. The second output presents common features of context for the profiles, which might point at problematic factors for the CASS (e.g., unrecognized context, lack of adaptability, design flaw). The third output presents the behavior of a profile by the changes between actions and contexts. Resulting from these, we should define new action plans to improve the system to resolve those problems for QinU assurance.

### 3.4 Tools and procedures

The framework includes several tools to assist the evaluator with the specification of model instances and target QinU values, as well as to execute the automatic QinU analysis and visualize the results. Figure 4 presents a diagram of the use of the tools along with the associated input and output elements. In the next sections, we describe the specification tool and the tool for the QinU assessment.

#### 3.4.1 Specification tool

FAQuiS includes a tool for model instantiation with the aim of generating the required input specifications for the assessment process: the context models, the log model, and the metrics model (from activities of the methodology Sects. 3.3.2, 3.3.3 and 3.3.4 respectively).

The tool for **model instantiation** allows the evaluator generating instances of the context models described in Sect. 3.2, or the model of log described in Sect. 3.3.3. To this end, the evaluator can specify the attributes for each model used in the framework according to the requirements and context conditions that have been elicited. Also, the evaluator can define a metrics model, described in Sect. 3.3.4, to deactivate certain metrics or specify ideal values. The objectives of this tool are to generate machine-readable specifications of the CASS information and to provide the evaluator a graphic way to indicate this.

This tool uses the respective metamodels (tasks, users, environment) to represent the specification. Through the user interface, each of the attributes presented in Sect. 3.2 can be introduced into the respective model. Regarding the model of log (i.e., the set of variables to be processed in the log processing), the evaluator specifies each of the fields in the interaction records that identify user, date and time, and any particular context state or environment information. Table 1 shows all variables types that can be defined. Through the

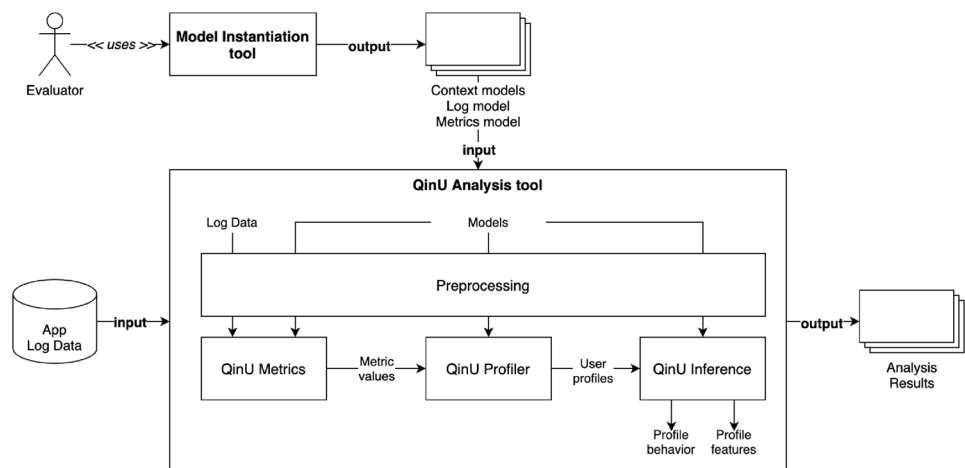
metrics model, it is possible to indicate the optimal QinU metrics (Sect. 3.1) that should be expected for a CASS. In this case, the evaluator can deactivate metrics and select only the relevant ones for a particular system. After this, the target values can be specified after the requirements obtained in the previous elicitation. This set of values defined an “ideal case” of use of the CASS. The assessment tool uses this set as the goal for the QinU analysis. Finally, the tool checks any invalid values and inconsistencies in any of the models created before generating the specifications that are used as input for the assessment tool.

#### 3.4.2 QinU assessment tool

The tool for the **QinU assessment** is the one responsible for the analysis of the QinU characteristics from the data. This tool receives as input the target metrics and models (user, tasks, contexts, and log description) from the previous tool, and the log records from the CASS. By means of these inputs, the tool assesses automatically the QinU of the system, using machine learning techniques such as probabilistic modeling, clustering, and pattern extraction, and provide the analysis results to the evaluator (activities Sects. 3.3.5 and 3.3.6 of the methodology). In order to perform the QinU analysis, the tool takes the following steps: (i) preprocessing the log data and model updating; (ii) estimation of the QinU characteristics; (iii) user profiling by their QinU results; (iv) analysis of the QinU profiles; (v) generation of the results (values, patterns, and charts).

First, the tool processes the models instances provided. Through these models, the tool is able to handle the CASS structure and the usage exhibited by the user (task model), collect the singular traits of the user (user model), and interpret the different context dimensions that change or have an impact on the user interaction (environment model). Next, motivated by the process discovery techniques of Process Mining, the tool processes

**Fig. 4** Use of the tools of FAQuiS



the log registers and discovers significant features of the users' activity (previously studied in (Salomón et al. 2019)). Through this phase, we build Markovian structures (Weighted Finite Automata or WFA (Droste et al. 2009)) by computing pseudo counts of actions and context changes and training them through the Maximum A Posteriori (MAP) estimation method. These structures model the transitions (as directed arcs) between states or actions and use the estimated probabilities as weights. Thus, we capture the activity and behavior patterns in these structures with a probabilistic approach. As a consequence, the weights can be used to extract the most relevant patterns (i.e., most frequent and probable ones).

Next, the tool computes QinU indicators to quantify quality characteristics (effectiveness, efficiency, satisfaction, freedom from risk) within all context states (context coverage). Therefore, we use heuristic functions composed of sets of metrics (described in Sect. 3.1 and detailed in Table 4) to estimate the QinU characteristics from the data, the models, and the patterns extracted. The estimations are scaled using ideal values of the metrics model (optionally indicated by experts) and ranges computed from the log data. This step outputs vectors of QinU normalized values for each user in the data.

After the QinU metrics are computed, the tool applies unsupervised learning to identify profiles in the users' community. Thus, users are grouped by similar values of their QinU vectors using a clustering technique [choosing between K-Means++ (Arthur and Vassilvitskii 2007) or Hierarchical Agglomerative (Müllner 2013)] with Euclidean distance. The number of groups is chosen automatically according to the optimal values of the Silhouette score, Calinski-Harabasz score, or Davies-Bouldin score (Liu et al. 2010). Again, we build WFAs to discover the combined behavior patterns for each extracted group of users as a whole. These state-machine models enable the analysis of patterns of global interaction with a graphical representation.

Then, we can analyze the obtained profiles using supervised learning techniques to extract relevant features in them. For this, the tool performs feature selection from all context factors and user traits in the profile by means of information entropy, and ensemble classifiers (Random Forest with Gini criterion) (Saeyns et al. 2008). In this process, we train the ensemble models to classify interactions by their associated QinU profile (the target class), extract the important features in the classification, and filter the irrelevant ones based on information entropy of their variance. This step allows the tool to rank the factors' influence and recognize possible causes of QinU differences between the profiles. This way, if a factor or condition appears significantly in a profile, it could be related to the QinU problems of the profile (e.g., a user profile with lower efficiency could be correlated with certain context states or the lack of certain user skills).

Finally, the tool presents as output the estimated QinU characteristics and their fulfillment, the possible causes of QinU loss, and the graphical representations of interaction obtained from the WFAs. Figure 5 shows the complete flow of the QinU analysis performed by this tool.

## 4 Case study

The case study method was applied to validate the impact of our proposal in the context of a project of a real software development company. A case study is “an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident” (Yin 2008). Thus, the case study method was selected because it is aligned with our goal of analyzing the validation of the proposal in a real environment. For this purpose, the case study was carried out in the context of a project of Axpe Consulting, an IT consulting and software development company.

Axpe Consulting is a certified Testing Maturity Model integrated (TMMi) organization that takes this model as a reference to perform test activities. TMMi is a framework (Vemulapalli 2015) that includes guidelines to carry out testing activities. Axpe Consulting applies these TMMi practices to test software in each iteration of the development process. Axpe Consulting is also a certified Capability Maturity Model Integration (CMMI) Level 5 organization (Chrisis et al. 2011). CMMI includes process and product quality assurance. The quality assurance processes of Axpe Consulting had been focused on the internal and external software quality, but it had not assessed QinU. We considered that this aspect makes Axpe Consulting suitable to carry out a case study of the impact of a QinU assessment framework.

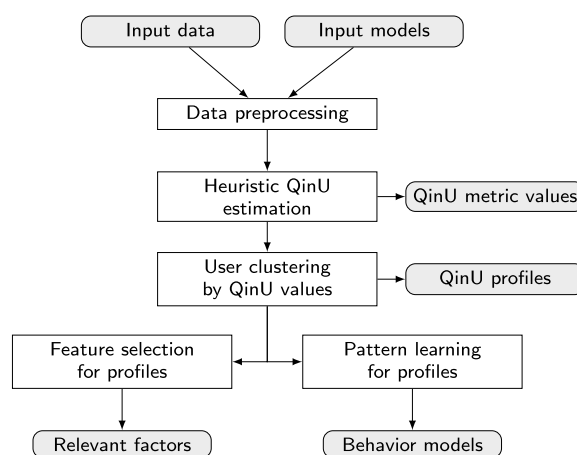


Fig. 5 Flow of the automated process of QinU analysis (QinU assessment tool)

A single case was carried out and this can be considered as unique-holistic according to the classification of Yin (2008) since it enables us a global study of the impact of applying the framework in a project of Axpe Consulting. The goal of the Axpe Consulting project was developing *Collbets*, a mobile context-aware application that supports collaborative sports betting.

The protocol template proposed by Brereton et al. (2008) was followed to carry out the case study. The goal of the study is the framework described in Sect. 3. The **Main Research Question** (MRQ) of this case study was established as follows: Is FAQuiS suitable for carrying out QinU evaluations of CASSs in software companies? This question is addressed with an analysis of the support for assessing QinU of CASSs using the characteristics of the ISO 25010 model and a study of the suitability of the methodology for guiding companies. Thus, we defined the following Specific Research Questions (SRQ).

- **SRQ1:** Is the framework support suitable for evaluating the QinU of CASSs?
- **SRQ2:** Is it feasible to apply the framework to measure the QinU characteristics of the ISO 25010 model?
- **SRQ3:** Is the framework methodology suitable for guiding companies on carrying out QinU evaluations in real projects?

The unit of analysis was the methodology and the software tools of the framework. The participants in the case study were the company (project manager and users of the framework) and users of *Collbets*. The data collection techniques used can be classified as follows (Lethbridge et al. 2005).

- **First degree:** Interviews with Axpe Consulting participants to collect qualitative data of the support provided by the framework and its application to the *Collbets* project. Users of *Collbets* answered questionnaires that collected data on their perception of the quality of the application.
- **Second degree:** The activities of the framework methodology generated models and values of the QinU metrics that were considered as part of the case study data.

#### 4.1 Intervention

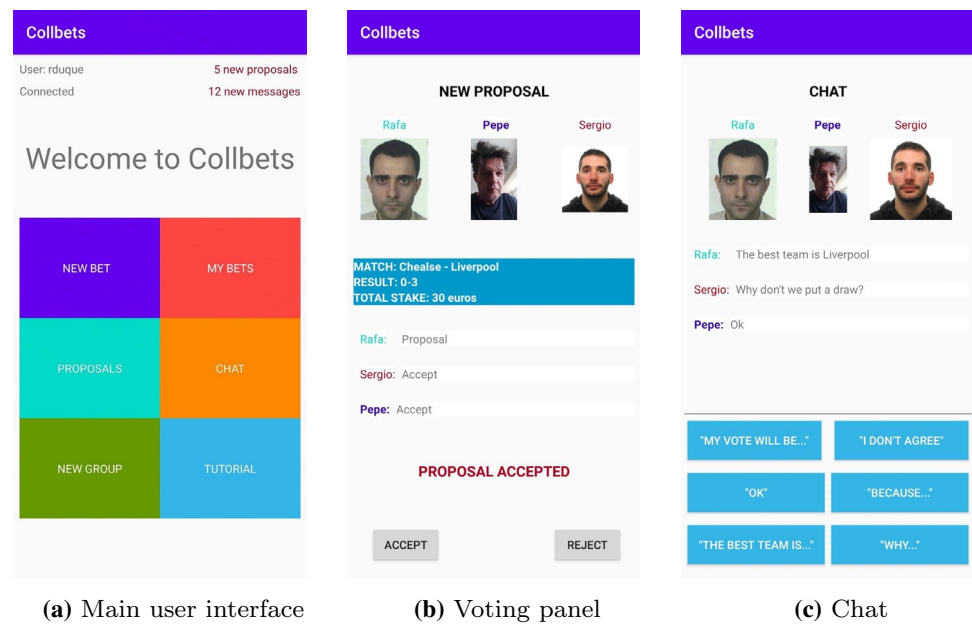
This subsection describes the execution of the case study. The framework methodology is integrated into the software development process of Axpe Consulting, requiring changes in three main phases: the elicitation of software requirements, the design and construction, and the quality control once the software is released. The new operations in each phase are described as follows.

- **Elicitation and analysis of requirements:** the context conditions must be identified and the QinU minimum requirements according to each characteristic should be established. Next, the models of context should be instantiated, and the log variables have to be specified.
- **Software design and construction:** in this phase, the software design has to contemplate an operation to capture log records of the expected format, following the context conditions that have been previously selected.
- **Software release and quality analysis:** in this phase, the QinU assessment tool is applied to estimate the QinU of the software in order to determine quality control. At this point, the optimal QinU indicators for the assessment have to be elicited to define the ideal case of QinU. The collected log registers and model instances of the context are feed into the evaluation tool. Then, the quality requirements of the QinU model can be analyzed to verify the degree of fulfillment.

The case study puts all the elements of the framework into action with the assessment of the QinU of *Collbets*, a smartphone application, shown in Fig. 6, that supports collaborative sports betting. *Collbets* supports two types of human-machine interactions: explicit and implicit. Explicit interactions are those tactile interactions that the user performs with the screen of the smartphone to use *Collbets* features for coordinating bets with other users. Implicit interactions are used to characterize aspects of the user's context (location, activity, etc.) and provide services that are adapted to each context. *Collbets* manages two types of implicit interactions to characterize the context of use: (i) GPS location is used to suggest sports events near the user's location and nearby users of the app, and (ii) user inactivity is detected by the app to adapt the interface actions for synchronous or asynchronous collaboration.

The system includes two types of mechanisms to react to different contexts. The first mechanism reacts to the physical context to suggest sports events near the user's location and to detect nearby users of the app and suggest starting a face-to-face communication to coordinate new bets with them. The second mechanism reacts to the type of collaboration context (synchronous or asynchronous). *Collbets* shows the photographs of all the users who are connected and the actions they carry out in real-time when a synchronous collaboration context is detected. However, *Collbets* displays a summary of the interactions carried out (number of proposals, number of acceptances and rejections, etc.) in asynchronous contexts so that after a period of inactivity the user does not have to review each of the actions carried out by others.

*Collbets* includes six main spaces in which the user can interact: (i) *new bets* to create a new betting proposal to be sent to other users, (ii) *my bets* to know the state of previous

**Fig. 6** Collbets application interface

user' bets, (iii) *proposals* to accept or reject bet proposals of other users, (iv) *chat* to talk with other users, (v) *new group* to create a group of bettors and (vi) *tutorial* collects a set of instructions about the process of carrying out bets with the app.

In order to apply the framework in the case study, we follow the steps of the methodology introduced in Sect. 3.3. Thus, the first task we conduct is the elicitation of the context conditions according to the tasks, users, and the environment. First, we identify the tasks and actions that Collbets supports to manage bets, send messages, and see sports events. All actions supported by Collbets are shown in Table 6 with their associated types and risks (none identified, as there was no real investment for the bets made in the experiment). These low-level activities, the ones captured in the log of the system, then compose the tasks of “Send message”, “Manage group”, “Manage bet proposal”, “Create new bet” and “See tutorial”. Next, we consider the user characteristics that may be relevant in Collbets. In Collbets, all users had similar traits and skills, no personal information was considered relevant, and many users were interested in sports betting. Afterward, we identify the environmental

aspects of Collbets: (i) in the organizational context we consider the conditions of collaboration between users to complete bets and synchronous or asynchronous interaction; (ii) we consider the changes in the physical context according to the GPS location of the users; (ii) we identify technological contexts in relation to the connectivity status when the app is used. These contexts are shown in Table 2. Once we analyze all the context conditions, we instantiate the models of tasks, users, and the environment using the tool provided in the framework. This way, we model all the relevant contexts that should be analyzed as influential factors in Collbets. In Fig. 7, we show an example of a model instance in the tool provided. Next, we must define the log structure according to the variables that are captured in log data. In these records, six variables are included: timestamp, user identifier, action identifier, synchronous or asynchronous context, and collaboration context. While there are context factors of the app according to the connectivity and location of the users, in our collected data we did not register significant changes that influenced the interaction. Then, the target QinU values are specified. For this step, QinU metrics based on risk actions are omitted (see *freedom from risk*, *trust*, and *usefulness*

**Table 2** Elicited environment factors of the application *Collbets*

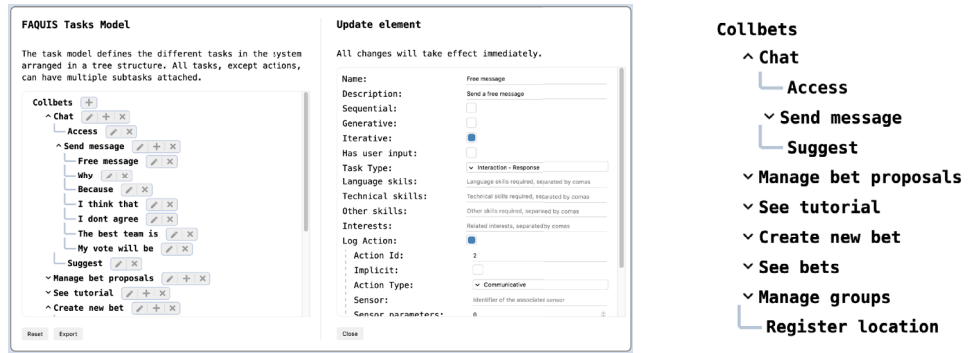
Environment aspect	Context	Collbets response
Organizational	Synchronous context	Adaptation of the voting panel with images of connected users and actions they carry out
	Asynchronous context	Adaptation of the voting panel based on user history
	Collaboration context	Users collaborate using actions created to this end
Physical	Location context	Suggestions for nearby events and users
Technological	Connectivity context	Connectivity status may influence users interaction

metrics in Table 4), and the suitable metric values are chosen according to the scope of the case study: users were asked to complete 5 bets during the data collection process.

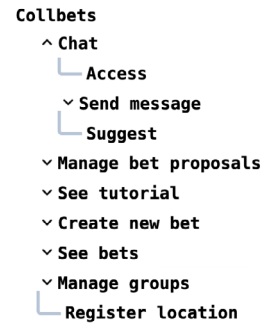
After applying FAQuiS with the previous inputs, we obtain the following outputs of the analysis: QinU values

estimated for the different profiles and contexts, frequent and distinct features of these profiles, and interaction representations of actions and contexts transitions. Here, we include the QinU metrics obtained (Fig. 8a) and metrics in each profile (Fig. 9), the metric differences according to the

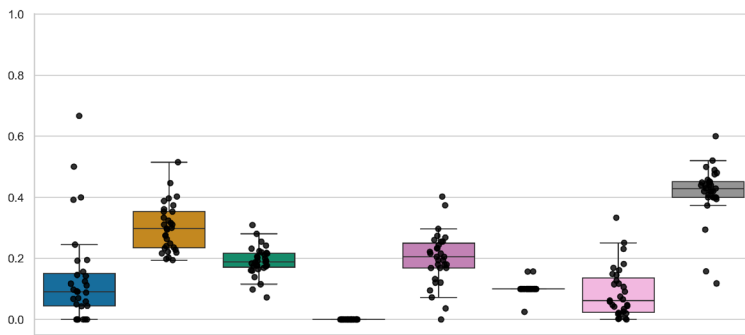
Fig. 7 Collbets task model



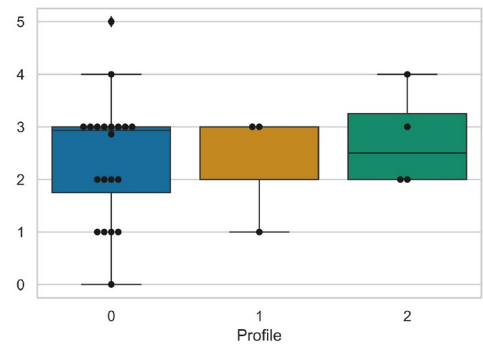
(a) Model instantiation tool.



(b) Task tree.



(a) Estimated QinU metrics (normalized): effectiveness, efficiency, satisfaction, risk mitig., usefulness, trust, comfort, and pleasure.



(b) Satisfaction (from 0 to 5) for each profile.

Fig. 8 QinU metrics obtained for all users and their satisfaction values expressed in a survey (each point is the value of a single user)

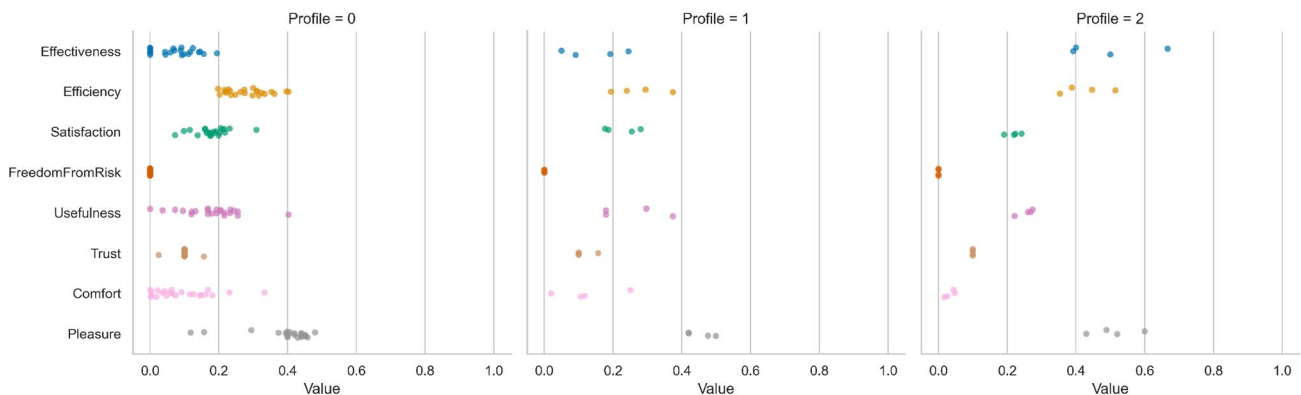


Fig. 9 QinU values for the user profile (each point is a user)

synchronous context (Fig. 10) and collaboration contexts (Fig. 11), and fragments of the relevant patterns of interaction (behavior structures) from the profiles (Fig. 12). For the sake of completeness, we also present the satisfaction values that were collected through a survey from the users (Fig. 8b). We represent those values within the estimated QinU profiles, but there are no noticeable differences.

In the charts, we observe the fulfillment measured by the tool (see Fig. 8a), where most of the profiles do not

meet minimum requirements. In this case, freedom from risk was not estimated, as there was no real risk perceived by the users. From the three profiles recognized by the QinU assessment tool, we can appreciate the differences between their QinU values (Fig. 9): users from profile 2 present better values of effectiveness, efficiency, and pleasure; profile 0 capture most of the users in the system with greater variance in their values but worse results than the other two.

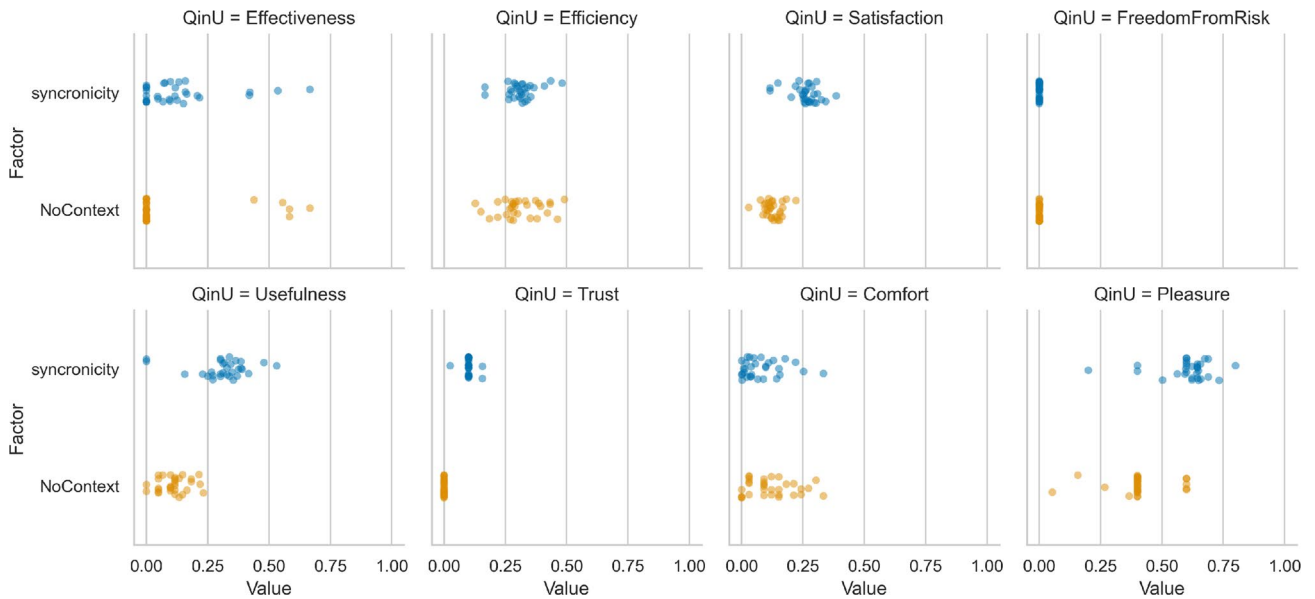
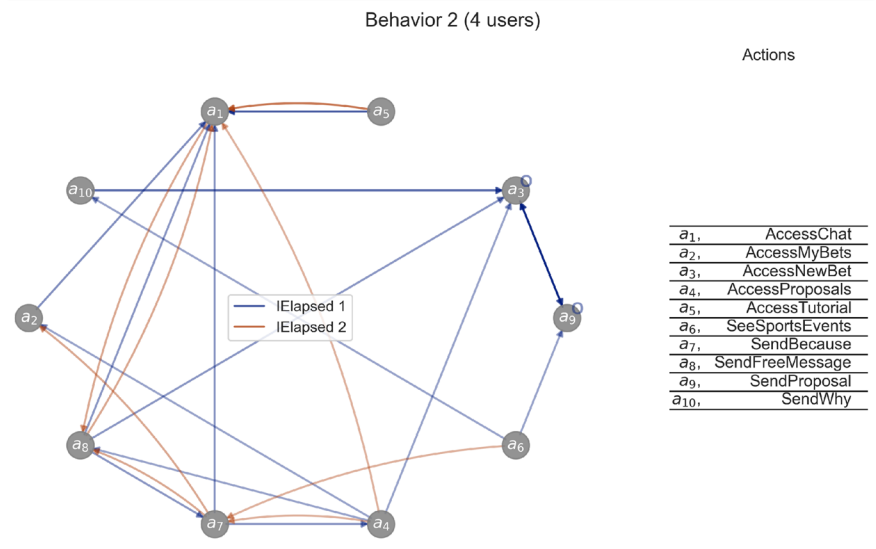
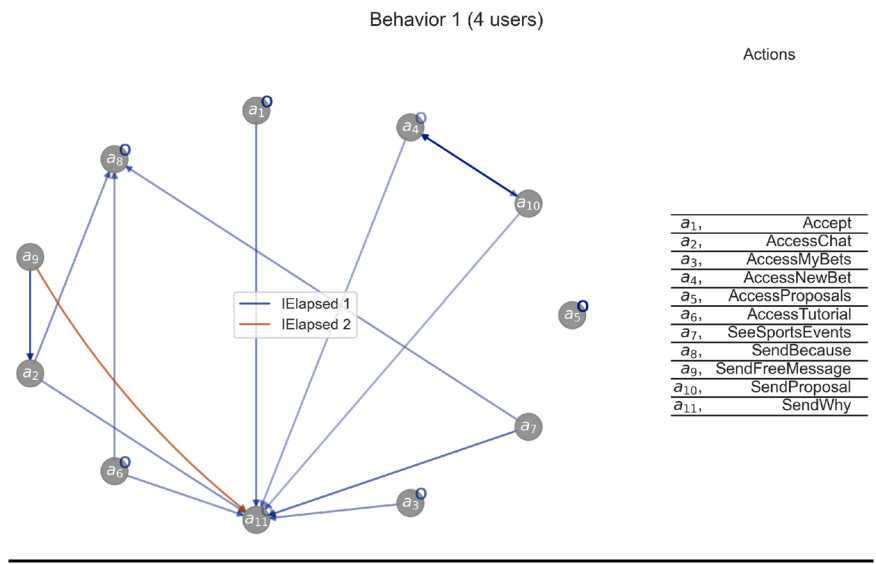
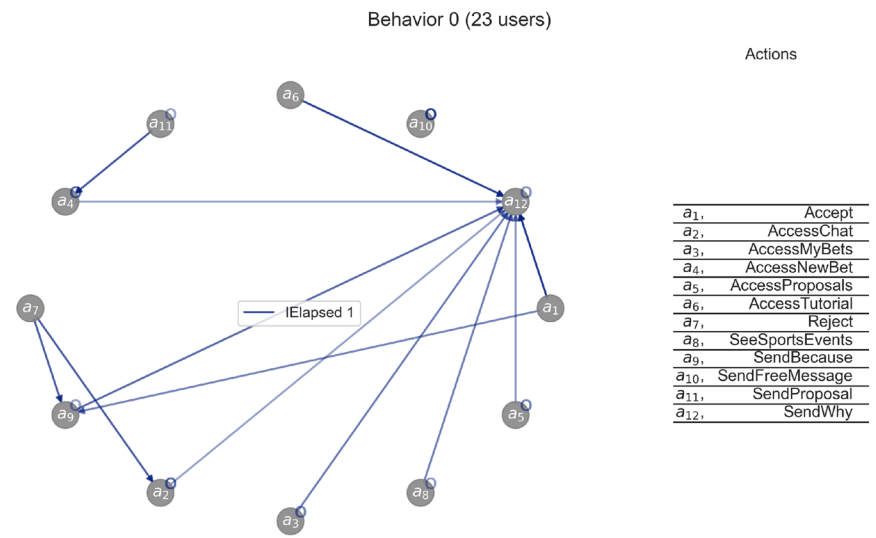


Fig. 10 Context coverage metrics estimated for synchronous contexts and no context (i.e., asynchronous cases)



Fig. 11 Context coverage metrics estimated for collaboration contexts

**Fig. 12** Interaction behavior patterns represented for each profile. Edge color represent synchronous (blue) and asynchronous (orange) context. Only significant patterns (directed edges) are shown





FAQuiS identified significant differences in the different contexts of the app. We can observe in the metrics of the synchronous context (see Fig. 10) that, in comparison to the asynchronous cases, the effectiveness, usefulness, satisfaction, pleasure, and trust are greater while there is no significant difference in efficiency and comfort. Also, FAQuiS enabled the analysis of the collaboration contexts that appeared during the interaction (through collaboration groups). Figure 11 show nine contexts that, while their values are limited to the interaction during collaboration, exhibit differences in some of the metrics. It is specifically significant the variance in the groups (where maximum values are obtained only by one user by effectiveness or comfort) and the difference of trust values (where only three groups have values greater than zero, probably related to the use of more tools and options of the app).

Finally, the behavior structures (Fig. 12) display the relevant patterns in the chat usage and bet voting, as well as the contexts of synchronous or asynchronous interaction during those actions. We can observe that most interactions from profile 0 were synchronous, while the ones from profile 2 also occur in an asynchronous context. Also, profile 2 does not include actions to resolve the bet proposals among its patterns.

## 4.2 Data analysis and results

This section analyzes the results of evaluating FAQuiS through the three specific research questions of the case study.

### 4.2.1 SRQ1: Is the framework support suitable for evaluating the QinU of CASSs?

The mechanisms to identify contexts and user activity patterns that are related to certain levels of QinU was one of the aspects that Axpe Consulting valued the most. According to the comments of the project team, these models are useful to identify specific features of the system that must be modified to improve QinU and to know the influence of the contexts in the assessment results. More specifically, the project team highlighted three important elements.

The first element is the graphical visualization of the activity of user clusters in different contexts. The graphical representation of how users with similar activity patterns interact in different contexts allowed the project team to obtain insight into how the system is used and the impact it has on QinU. The project team argues that these graphical visualizations helped them to interpret the impact that context changes have on the QinU of Collbets.

The second element is the organizational dimension of the context. Axpe Consulting traditionally had conceived that the main variables of the context are the user's location

and nearby devices that can be connected. However, the project team considered that was essential to include other users as entities of the context. This is a consequence that the current systems usually integrate features for collaboration and communication. In these systems, the activity of a user is influenced by other users who share a collaborative context. For example, the project team concluded that the most important finding of the QinU evaluation is that Collbets should include new mechanisms that discriminate synchronous and asynchronous collaboration contexts and adapt the system functionalities to each case.

Finally, the last highlighted element is the context modeling. Framework support for modeling contexts of use played an important role in the Collbets evaluation process. This support allowed the project team to have the flexibility to define which elements of the context (location, other users, etc.) should be considered to study their impact on the QinU of Collbets. For the project team, this support is suitable to model the contexts that the system under evaluation must cover and later verify the levels of QinU in each of them.

After analyzing these data from Axpe Consulting, we can conclude that the framework is a solution to assess QinU considering and interpreting the impact that different contexts have on the evaluation (SRQ1).

### 4.2.2 SRQ2: Is it feasible to apply the framework to measure the QinU characteristics of the ISO 25010 model?

To address this issue, the project team was requested to analyze the adequacy of the metrics used by the framework (see Table 4) to quantify the QinU characteristics of the ISO 25010 (SQuaRE) model. In the analysis, the project team highlighted different aspects for each of the QinU characteristics.

Regarding effectiveness and efficiency, the metrics that quantify these characteristics are considered functional and adequate. These metrics clearly measure the resources expended (efficiency) to achieve specified goals (effectiveness) related to the tasks supported by the system. Moreover, the project team considers that is feasible to apply software tools to calculate automatically these metrics.

About the context coverage, the project team finds that the metrics quantify the purpose of this characteristic. They consider that these values should be interpreted with the help of the relevant profile factors and the graphic interaction models since they allow us to understand how the system adapts to different contexts, which is the main purpose of the context coverage characteristic in the SQuaRE model.

The satisfaction characteristic is related to some hedonic aspects of the user. For this reason, the project team considers that the most appropriate is that users answer surveys to rate their satisfaction and explain the causes of their

level of satisfaction. However, the project team values very positively to complement surveys with the use of FAQuiS support to calculate automatically metrics of satisfaction sub-characteristics (comfort, usefulness, pleasure, and trust). This is justified by the project team in that FAQuiS support calculates metrics that allow us to identify situations indicative of satisfaction problems that are not reported by the users (for example, users that delete an *app* of their smartphone shortly after downloading and installing it, but without using the surveys to communicate the low satisfaction experience).

The freedom from risk characteristic is treated by FAQuiS as a particular perception of the user about the risks from using the system. This user perception does not involve a systematic assessment of the security of the system. The project team considered that the metrics of these characteristics are a positive complement to the systematic security test carried out by Axpe Consulting. The project team emphasizes that security tests are essential to verify that the perception of unacceptable risk mitigation is due to vulnerabilities of the system instead of, for example, flaws of the user interface to inspire confidence in carrying out certain tasks.

We can conclude from the analysis of the data provided by the project team, the Collbets users, and the metrics calculated in the case study that it is feasible to apply the support of the framework to quantify the QinU characteristics of the SQuaRE model (SRQ2). Moreover, QinU profiles, graphic interaction models, and security tests must be used as complements to help the project team interpreting the values of the QinU metrics.

#### 4.2.3 SRQ3: Is the framework methodology suitable for guiding companies on carrying out QinU evaluations in real projects?

The application of the framework to the Collbets project allows us to verify that it can be used by companies to carry out QinU evaluations. The review of related works (Sect. 2) shows a lack of technological support to carry out QinU evaluations adapted to each system and also considering the influence of the context on the QinU. The framework tools enable QinU evaluations that are directed by models of the context in which users interact with the system.

The framework methodology includes a set of activities that were performed by the project team to evaluate the QinU of Collbets. The project team did not replace those activities applied previously by Axpe Consulting during the software development processes. The new activities of the framework methodology were added to those previously considered by Axpe Consulting in the phases of requirement analysis, design, construction, and quality control. This ease the integration of the framework methodology in the company. In summary, the application of the framework

and information provided by the company allows us to conclude that the methodology was suitable for guiding the project team on carrying out the QinU evaluation of Collbets (SRQ3). Additionally, the project team gained insight about the methodology activities to design the ideal case (i.e., optimal QinU metrics) and to generate log traces.

One of the most innovative activities for the company was the definition of the ideal case. According to the project team, the ideal case method involves a very novel effort to specific target values of the QinU metrics during the phase of requirement analysis. This novel task requires additional effort by the project team and the stakeholders to elicit these requirements. However, the project team argued that this effort is justified by the advantage of having an objective interpretation of QinU metrics values based on the degree of similarity with a case considered ideal by experts, developers, or evaluators, to carry out an activity in a specific context.

Finally, the project team pointed out that Axpe Consulting also provides software quality assurance outsourcing services. To apply the framework in these services, software maintenance activities must be carried out to integrate log generation features into an already built system. Most systems do not usually generate log files with the level of detail required by the framework. For this reason, this maintenance activity may be the one that requires the most effort to apply the framework to already built systems.

### 4.3 Validity threats

Threats to the validity of the case study are analyzed below according to the four dimensions of validity specified by Runeson et al. (2012): construct validity, internal validity, external validity, and reliability.

*Construct validity.* The framework takes the SQuaRE model as the reference standard to assess QinU characteristics. This avoids discrepancies in the purpose of the QinU assessment as the standard is widely accepted. Moreover, the researchers held meetings to explain how each of the activities of the framework methodology was applied and clarify the goals of the research. Therefore, these meetings can be considered as a means of avoiding doubts about the framework support and the research questions.

*Internal validity.* To avoid other factors affecting the case study, Axpe Consulting had personnel dedicated exclusively to apply the framework to assess the QinU of Collbets. This enabled the team of Axpe Consulting to avoid time restrictions and sharing resources from other tasks.

*External validity.* The case study has selected Axpe Consulting as a representative software development company that had not considered QinU in its evaluation processes. Therefore, the results of this case study can be considered of interest for companies with these characteristics. Moreover,

the process of applying the framework can be generalized to other companies. The two main tasks to adapt the framework are (i) to instantiate the framework models and (ii) to incorporate mechanisms in the software systems to generate log files with the interactions of the users in the format followed by the framework.

*Reliability.* The case study has included a set of documented and structured activities to apply the framework and a set of research questions. These activities and questions do not depend on the researchers participating in the case study. Therefore, it is estimated that the case study and its results should be similar. The differences in the results of a new case study could be caused by a modification of the characteristics of the company or a combination in the system whose QinU is evaluated.

## 5 Discussion

FAQuiS offers a solution for assessing the QinU in any software system or CASS. This framework uses several techniques from machine learning and process mining to generate automatic analyses of the QinU based on log data and domain specification. In this process, the quality characteristics from ISO SQuaRE (2011) are considered to estimate QinU values. FAQuiS includes specific activities and tasks that experts (e.g., developers or quality evaluators) need to carry out in order to apply the framework to any use case. This framework has been validated by its impact and integration in the software industry, conducting a case study in an actual software development company.

This work takes the approach proposed and evaluated in (Salomón et al. 2019) as groundwork. In the latter work, probabilistic models of behavior were discovered from log data and then correlations between QinU metrics and behavior clusters of users were studied. In FAQuiS, this idea has been expanded to generalize the approach to any CASS by integrating expert input and context modeling. The QinU analysis has been improved with many more metrics to measure QinU characteristics and with the extraction of patterns in contexts and quality profiles. FAQuiS incorporates a methodology and implemented tools to include the QinU assessment in software companies without requiring big efforts of implementation and integration.

In this research, a validation approach of FAQuiS is presented according to its impact and integration in the industry. This is done via a case study in the company Axpe Consulting. For this evaluation, Yin's methodology (Yin 2008) and Brereton's template (Brereton et al. 2008) for case study protocol is followed in order to assess the application of the framework in a software project. This approach is convenient for the proposal since it is intended to be applied by software companies in their development processes. The

main research question is established, about the suitability of FAQuiS to evaluate QinU of CASSs in companies, as well as several specific research questions, concerning: (1) the QinU evaluation in CASSs; (2) the feasibility of the QinU quantification; and (3) the suitability for guiding companies in actual projects. The validity of FAQuiS is verified (Sects. 4.2 and 4.3) with the aim of be integrated in actual software development projects. This case study also gives insight into the ease of integration and little effort in order to apply the solution in the industry.

FAQuiS provides many advantages for the QinU assessment problem in relation to previous works. Table 3 presents a comparison to the existing studies relevant for the problem. The comparison takes into consideration several features of the approaches that have been key in FAQuiS design and are described in the list below.

- i. FAQuiS is valid for application in CASSs, as it evaluates the context coverage and can analyze the context changes with the model representations. Many related studies are applied to CASSs and consider context changes, but just a few are focused on testing these systems (Augusto et al. 2019; Ben Ayed et al. 2016; Erazo-Garzon et al. 2021).
- ii. FAQuiS is intended to be general-purpose in order to be applied to any software system. This is achieved by the domain (environment, tasks, users) specification from experts required as input. A few proposals are intended to be adapted to the specific case by selecting the suitable measures or metrics (Rana and Staron 2015; Rauschenberger et al. 2013; Seffah et al. 2006), or even calculating the importance weight for each metric (Kim and Kim 2019). The approaches based on user reviews (Atoum 2020; Jiang et al. 2019; Leopairote et al. 2013; Qian et al. 2016) have the potential for generalization, but the adaptation process is not analyzed nor described. Also, (Augusto et al. 2019) describes a generalizable approach, but it is oriented to test context aspects (of CASSs) specifically.
- iii. The industrial validity of FAQuiS is tested with a case study in an actual software development company. This has provided insight into the framework impact and integration for the industry. Some related approaches mention the integration in the industry (Hynninen et al. 2018) or even describe application methodologies (Fogli and Guida 2018; Rauschenberger et al. 2013), but there are no validation analyses about the impact and required effort.
- iv. FAQuiS evaluation of QinU is based on an ISO standard, as it measures all QinU characteristics defined in the ISO/IEC 25010 SQuaRE. There are related studies based on this standard or others (such as ISO/IEC 9126–4), and some of them partially as not all QinU

**Table 3** Comparison of QinU evaluation methods taking into consideration: (i) validity for **Context-Aware Software Systems (CASS)**; (ii) evaluation with **general-purpose** (i.e., independent of the system that is evaluated); (iii) its **industry** application/integration is tested; (iv) based on **ISO Quality** standards and specifications; (v) based on **model representations** of the user and context of use; (vi) use of data-driven techniques for automation such as **machine learning (ML) or data mining**; (vii) use of **process mining (PM)** techniques with event log data

Research work	i CASS	ii General	iii Industry	iv ISO	v Models	vi ML	vii PM
Al-Nanah et al. (2009)	–	–	–	*	–	–	–
Alnanah et al. (2013)	–	–	–	*	–	–	–
Alshareet et al. (2018)	–	–	–	✓	–	✓	–
Atoum (2020)	–	*	–	*	–	✓	–
Augusto et al. (2019)	✓	*	–	–	–	–	–
Ben Ayed et al. (2016)	✓	–	–	✓	✓	–	–
Erazo-Garzon et al. (2021)	✓	–	–	✓	–	–	–
Fernández et al. (2009)	✓	–	–	–	*	✓	✓
Fogli and Guida (2018)	–	–	*	–	–	–	–
Gadler et al. (2017)	–	–	–	–	*	✓	✓
Hynninen et al. (2018)	–	–	*	✓	–	–	–
Jiang et al. (2019)	–	*	–	*	–	✓	–
Kim and Kim (2019)	–	✓	–	✓	–	–	–
Leopairote et al. (2013)	–	*	–	✓	–	✓	–
de Medeiros et al. (2004)	✓	–	–	–	*	✓	✓
Orehovački et al. (2013)	–	–	–	*	–	–	–
Osman and Osman (2013)	–	–	–	*	–	–	–
Qian et al. (2016)	–	*	–	*	–	✓	–
Rana and Staron (2015)	–	✓	–	✓	–	✓	–
Rauschenberger et al. (2013)	–	✓	*	–	–	–	–
Salomón et al. (2019)	–	–	–	*	✓	✓	✓
Seffah et al. (2006)	–	✓	–	✓	–	–	–
Souza-Pereira et al. (2021)	–	–	–	✓	–	–	–
<b>FAQuiS</b>	✓	✓	✓	✓	✓	✓	✓

The marks (✓) and (\*) indicate if the approach fulfills the feature completely or partially respectively

characteristics are used. Most of the approaches are inspired by standards and then create custom quality models (Al-Nanah et al. 2009; Alnanah et al. 2013; Orehovački et al. 2013; Osman and Osman 2013) or evaluate a subset of the QinU characteristics (Atoum 2020). Also, some approaches infer the QinU characteristics from the content in text comments (Atoum 2020; Jiang et al. 2019; Qian et al. 2016) instead of considering the standard specifications. Leopairote et al. (2013) build an ontology of the quality terms in order to include the standard descriptions. Finally, other proposals select or design metrics based on the standard QinU descriptions (Alshareet et al. 2018; Ben Ayed et al. 2016; Hynninen et al. 2018; Rana and Staron 2015; Seffah et al. 2006; Souza-Pereira et al. 2021) like FAQuiS approach.

- v. This framework is based on model representations of the user behavior and context changes that are exploited for the QinU analysis. This makes it possible to consider the patterns of context and user activity in the QinU evaluation. The only study (except for preliminary work in (Salomón et al. 2019)) following a similar approach for the QinU assessment is (Ben Ayed et al. 2016), which uses a context model to consider contextual data. While the process mining studies exploit this kind of model, they do not aim for a QinU evaluation.
- vi. FAQuiS' proposal apply machine learning techniques in order to automate the QinU analysis. These techniques are probabilistic modeling of the user patterns, clustering of user QinU metrics into profiles, and feature selection for relevant factors of the context. There

are many other studies applying machine learning and data mining approaches for automation and knowledge discovery. These are mostly based on natural language processing and sentiment analysis of user reviews (Atoum 2020; Jiang et al. 2019; Leopairrote et al. 2013; Qian et al. 2016), but also pattern recognition and classification of metrics (Rana and Staron 2015) or neural networks for metric classification (Alshareet et al. 2018). All process mining studies are considered as data mining approaches as well.

- vii. FAQuiS use a process mining approach by using log records to discover Markovian structures of user behavior and to estimate the QinU characteristics. There are many referenced studies that apply process mining approaches to discover models of user activity and behavior. Among these, (Gadler et al. 2017) also applies Markovian structures (Hidden Markov Models). However, there are not any studies addressing the QinU assessment with these techniques.

Through these previous aspects, it is highlighted the contribution of the proposal to the research topic. FAQuiS introduce some novelties and has advantages in comparison to the current state-of-the-art proposals, such as general-purpose, context integration, and partial automation for the QinU assurance process. Nevertheless, there are still some open challenges in this line of research. In this regard, it can be pointed out the limitations of flexibility and interpretability of the QinU evaluation, more mechanisms for problem identification and correction (i.e., causes of the QinU deficiencies), and means to detect relations between satisfaction surveys and estimated metrics. Because of this, future work should be aimed towards these challenges.

## 6 Conclusions

Software quality models incorporate characteristics related to the QinU that are focused on how a system covers different contexts. These characteristics are of particular relevance in the field of CASSs, which collect information from the context to adapt the features and services. This motivated our work to build FAQuiS, a model-based framework for assessing QinU considering the influence of the context.

FAQuiS is made up of a set of metamodels, a methodology, and support software tools. The metamodels define the structure that must follow a set of descriptive models of the system, the user types, and the environmental factors. The methodology includes a series of activities and tasks that guide FAQuiS users on how to carry out QinU evaluation processes directed by these models. The support tools enable the creation of the specific models and process repositories with the user actions in different contexts, estimating metrics that quantify the QinU characteristics of the ISO 25010 standard. In addition, these tools discover models of the behavior of the users in the interaction contexts and user profiles characterized by their QinU metrics and relevant factors of context.

Axpe Consulting, a software development company, applied FAQuiS to assess the QinU of a mobile app that supports collaborative betting and has context-sensitive features. A case study allowed us to evaluate FAQuiS' impact in a real environment. FAQuiS methodology was a complete guide for the company to carry out the QinU evaluation of the app. The project team incorporated the framework methodology, and they did not have to replace any of the previous activities of software development or testing process. The team considered that FAQuiS is a solution to assess the QinU using the ISO 25010 standard and modeling the user behavior in different contexts. In addition, the project team especially highlights the support of FAQuiS for modeling the different contexts and interpreting the quantitative values of the metrics.

Future work will be aimed at generating corrective measures for the detected deficiencies of QinU. Furthermore, FAQuiS can be made more flexible with tools that allow evaluators to define their own custom metrics to quantify QinU characteristics. Also, new case studies will be carried out in other companies and in projects that develop other types of CASSs (Internet of Things, Smart cities, etc.).

## Appendix

See Tables 4, 5 and 6.

**Table 4** FAQuiS metrics to estimate quality-in-use characteristics

Characteristic	Sub-char	Metric description
Effectiveness	–	<ul style="list-style-type: none"> <li>- Percentage of tasks with instrumental actions completed</li> <li>- Number of artifacts generated</li> <li>- Degree of similarity between user patterns and the ones established in the task model</li> </ul>
Efficiency	–	<ul style="list-style-type: none"> <li>- Number of spaces used</li> <li>- Negatively, time values obtained for the task completion</li> <li>- Number of actions executed</li> <li>- Number of tasks conducted</li> <li>- Ratio of actions executed in time</li> <li>- Ratio of artifacts generated in time</li> </ul>
Freedom from risk	Economic Environment Health/Safety	- Summation of interactions with the respective type of risk (economic, environmental or health related) according to their associated probability/frequency and impact
Satisfaction	Comfort	<ul style="list-style-type: none"> <li>- Time values spent conducting tasks that change system settings</li> <li>- Number of implicit interactions that get an implicit response required</li> <li>- Number of interactions on the same action (i.e., repetitions)</li> </ul>
	Pleasure	<ul style="list-style-type: none"> <li>- Negatively, number of tasks that require user input and the user interrupts them</li> <li>- Tendency to repeat tasks that require user input in different sessions</li> <li>- Percentage of tasks successfully finished that require new skills to the user</li> <li>- Variance in execution time between different sessions of tasks that require new skills to the user</li> <li>- Tendency to resume interrupted work sessions</li> </ul>
	Trust	<ul style="list-style-type: none"> <li>- Negatively, number of tasks that include risks and the user avoids executing</li> <li>- Number of tasks that include risks and the user executes repeatedly</li> <li>- Patterns of actions that do not follow the sequence of the task model because of an unexpected response from the system</li> <li>- Negatively, time values exhibited for executing tasks that include risks</li> <li>- Percentage of finished tasks started in all work sessions</li> <li>- Negatively, number of “backward” actions executed</li> <li>- Number of executed actions that require a (collaborative) response from another user</li> <li>- Time values obtained when executing actions that require a protocol response from another user</li> <li>- Number of implicit actions that require a response from the ubiquitous system</li> <li>- Negatively, time lapses obtained from implicit interactions that require a response from the ubiquitous system</li> </ul>
	Usefulness	<ul style="list-style-type: none"> <li>- Number of tasks that include actions with risks or instrumental type and are repeated in different sessions</li> <li>- Number of patterns with a successful response of implicit interaction</li> <li>- Percentage of tasks finished</li> <li>- Percentage of actions used</li> <li>- Percentage of spaces used</li> <li>- Percentage of system actions in relation to user actions</li> <li>- Percentage of tasks that are successfully performed by the user with support of implicit system responses</li> <li>- Percentage of tasks that are successfully performed by the user with protocol support (i.e., includes any protocol-based action)</li> </ul>
Context	Completeness	- All of the above metrics for each context
coverage	Flexibility	- All of the above metrics with unrecognized context

**Table 5** Methodology overview of FAQuiS connecting the elements of the framework

Activity	Specific tasks	Input	Output	Tool
Elicitation of the context conditions	Identify all the tasks and actions supported by the system	Domain knowledge	Context elicitation	–
	Identify the relevant user characteristics	Domain knowledge	Context elicitation	–
	Identify the existence of relevant conditions in the technological context	Domain knowledge	Context elicitation	–
	Identify the existence of relevant conditions in the physical context	Domain knowledge	Context elicitation	–
	Identify the existence of relevant conditions in the organizational context	Domain knowledge	Context elicitation	–
Instantiation of the models of context	Specify tasks and attributes of the system	Context elicitation	Tasks model	Model instantiation
	Specify users and their relevant traits	Context elicitation	Users model	Model instantiation
	Specify environment factors along organizational, physical and technological dimensions	Context elicitation	Environment model	Model instantiation
Definition or adaptation of the log records	Selection of variables captured in the logs	Context elicitation	Log requirements	–
	Design and adaptation of the system to generate the logs	Log requirements	Log data	–
	Instantiation of the log model	Log requirements	Log model	Model instantiation
Elicitation of the target QinU indicators	Select the suitable QinU metrics	Domain knowledge	Metrics model	Model instantiation
	Elicitation of optimal values for the metrics	Domain knowledge	Metrics model	Model instantiation
	Specification of minimum requirements of QinU characteristics	Domain knowledge	QinU requirements	–
Execution of the QinU automated analysis	–	Log data, All models	QinU results	QinU assessment
Interpretation of the QinU analysis	Visualization and analysis of the results	QinU results, QinU requirements	QinU insights	QinU assessment
	Decision-making for QinU improvement	QinU insights	Action plan	–

**Table 6** Actions in the application *Collbets* from the case study presented

App Space	Action Description	Action Type	Risk	
Structured chat	Access to this space	Cognitive	–	
	Send a free message	Communicative	–	
	Send a “Why...” message	Communicative	–	
	Send a “Because” message	Communicative	–	
	Send a “I think that” message	Communicative	–	
	Send a “I don’t agree” message	Communicative	–	
	Send a “The best team is” message	Communicative	–	
	Send a “My vote will be” message	Communicative	–	
	Suggest predefined messages	- (System)	–	
	Proposals	Access to this space	Cognitive	–
		Reject a proposal	Protocol-based	–
		Accept a proposal	Protocol-based	–
		A proposal was accepted by the group	Cognitive	–
A proposal was rejected by the group		Cognitive	–	
New group	New Name	Protocol-based	–	
	Add a person	Protocol-based	–	
	Select a group	Protocol-based	–	
	Rename a group	Protocol-based	–	
	Send an invitation	Protocol-based	–	
	Accept an invitation	Protocol-based	–	
	Reject an invitation	Protocol-based	–	
	New bet	See sports events	Cognitive	–
Access to this space		Cognitive	–	
Send a proposal		Instrumental	–	
My bets	Access to this space	Cognitive	–	
Tutorial	Access to this space	Cognitive	–	

**Acknowledgements** The authors want to acknowledge the financial support from the operative program FEDER (Fondo Europeo de Desarrollo Regional) and SODERCAN (Sociedad para el Desarrollo de Cantabria) within the program “I+C=+C 2019 - APOYO A PROYECTOS DE I+D EN EL ÁMBITO DE LAS TIC”.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Abowd GD, Dey AK, Brown PJ, Davies N, Smith M, Steggle P (1999) Towards a better understanding of context and context-awareness. In: Proceedings of the 1st international symposium on handheld and ubiquitous computing, Springer-Verlag, Berlin, Heidelberg, HUC '99, pp 304–307
- Al-Nanhi R, Al-Nuaim H, Ormandjieva O (2009) New health information systems (HIS) quality-in-use model based on the GQM approach and HCI principles. In: Jacko JA (ed) Human-computer interaction. Interacting in various application domains. Springer, Berlin, pp 429–438. [https://doi.org/10.1007/978-3-642-02583-9\\_47](https://doi.org/10.1007/978-3-642-02583-9_47)
- Alnanhi R, Ormandjieva O, Radhakrishnan T (2013) A new quality-in-use model for mobile user interfaces. In: 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement, pp 165–170
- Alshareet O, Itradat A, Doush IA, Quttoum A (2018) Incorporation of ISO 25010 with machine learning to develop a novel quality in use prediction system (QiUPS). *Int J Syst Assur Eng Manag* 9(2):344–353. <https://doi.org/10.1007/s13198-017-0649-x>, <http://link.springer.com/10.1007/s13198-017-0649-x>
- Arthur D, Vassilvitskii S (2007) K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, SODA '07, pp 1027–1035
- Atoum I (2020) A novel framework for measuring software Quality-in-Use based on semantic similarity and sentiment analysis of software reviews. *J King Saud Univ Comput Inf Sci* 32(1):113–125,



- <https://doi.org/10.1016/j.jksuci.2018.04.012>, <http://www.sciencedirect.com/science/article/pii/S1319157817305268>
- Augusto JC, Jose Quinde M, Oguego CL (2019) Context-aware systems testing and validation. In: 2019 10th International Conference on Dependable Systems, Services and Technologies (DESSERT), pp 7–12, <https://doi.org/10.1109/DESSERT.2019.8770048>
- Ben Ayed E, Kolski C, Magdich R, Ezzedine H (2016) Towards a context based evaluation support system for quality in use assessment of mobile systems. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp 004350–004355, <https://doi.org/10.1109/SMC.2016.7844915>
- Bevan N (1999) Quality in use: meeting user needs for quality. *J Syst Softw* 49(1):89–96. [https://doi.org/10.1016/S0164-1212\(99\)00070-9](https://doi.org/10.1016/S0164-1212(99)00070-9)
- Brereton P, Kitchenham B, Budgen D, Li Z (2008) Using a protocol template for case study planning. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, BCS Learning & Development Ltd., Swindon, GBR, EASE 2008, pp 41–48
- Capdevila MG, Saltiveri TG, Garrido JE, Müller OH, Ruas LC (2021) Do current user testing practices meet the needs of the new interactive paradigms? In: Proceedings of the XXI International Conference on Human Computer Interaction, Association for Computing Machinery, New York, NY, USA, Interacción '21, <https://doi.org/10.1145/3471391.3471416>, <https://doi.org/10.1145/3471391.3471416>
- Caron F, Vanthienen J, Baesens B (2013) Comprehensive rule-based compliance checking and risk management with process mining. *Decis Support Syst* 54(3):1357–1369. <https://doi.org/10.1016/j.dss.2012.12.012>, <https://linkinghub.elsevier.com/retrieve/pii/S0167923612003788>
- Carvalho RM, Castro Andrade RM, Oliveira KM, Sousa Santos I, Bezerra CI (2017) Quality characteristics and measures for human–computer interaction evaluation in ubiquitous systems. *Software Qual J* 25(3):743–795. <https://doi.org/10.1007/s11219-016-9320-z>
- Cherfia TA, Belala F, Barkaoui K (2017) A bigraph-based framework for specification and analysis of context-aware systems. *Int J Crit Comput-Based Syst* 6(4):322–342. <https://doi.org/10.1504/IJC-CBS.2016.081808>
- Chrissis MB, Konrad M, Shrum S (2011) CMMI for development: guidelines for process integration and product improvement, 3rd edn. SEI Series in Software Engineering, Addison-Wesley, Upper Saddle River, NJ, <http://my.safaribooksonline.com/9780321711502>
- de Medeiros AKA, van Dongen BF, van der Aalst WMP, Weijters AJMM (2004) Process mining for ubiquitous mobile systems: an overview and a concrete algorithm. In: Proceedings of the Second CAiSE Conference on Ubiquitous Mobile Information and Collaboration Systems, Springer-Verlag, Berlin, Heidelberg, UMICS 04:151–165. [https://doi.org/10.1007/978-3-540-30188-2\\_12](https://doi.org/10.1007/978-3-540-30188-2_12), [https://doi.org/10.1007/978-3-540-30188-2\\_5F12](https://doi.org/10.1007/978-3-540-30188-2_5F12)
- de Sousa Santos I, de Castro Andrade RM, Rocha LS, Matalonga S, [de Oliveira] KM, Travassos GH (2017) Test case design for context-aware applications: Are we there yet? *Inf Softw Technol* 88:1–16, <https://doi.org/10.1016/j.infsof.2017.03.008>, <http://www.sciencedirect.com/science/article/pii/S0950584917302513>
- Djoudi B, Bouanaka C, Zeghib N (2016) A formal framework for context-aware systems specification and verification. *J Syst Softw* 122:445–462, <https://doi.org/10.1016/j.jss.2015.11.035>, <http://www.sciencedirect.com/science/article/pii/S0164121215002599>
- Droste M, Kuich W, Vogler H (2009) Handbook of weighted automata, 1st edn. Springer Publishing Company, Berlin (**Incorporated**)
- Duque R, Bravo C, Ortega M (2013) An ontological approach to automating collaboration and interaction analysis in groupware systems. *Knowl-Based Syst* 37:211–229
- Erazo-Garzon L, Erraez J, Cedillo P, Illescas-Peña L (2020) Quality assessment approaches for ambient assisted living systems: a systematic review. In: Botto-Tobar M, Zambrano Vizuete M, Torres-Carrión P, Montes León S, Pizarro Vásquez G, Durakovic B (eds) Applied technologies. Springer International Publishing, Cham, pp 421–439
- Erazo-Garzon L, Illescas-Peña L, Cedillo P (2021) A quality in use model for ambient assisted living (AAL) systems. In: Botto-Tobar M, Montes León S, Camacho O, Chávez D, Torres-Carrión P, Zambrano Vizuete M (eds) Applied technologies. Springer International Publishing, Cham, pp 643–660
- Fernández C, Lázaro JP, Benedi JM (2009) Workflow mining application to ambient intelligence behavior modeling. In: Stephanidis C (ed) Universal access in human-computer interaction. Intelligent and ubiquitous interaction environments. Springer, Berlin, pp 160–167
- Fogli D, Guida G (2018) Evaluating quality in use of corporate web sites: an empirical investigation. *ACM Trans Web* 12(3):15:1-15:35
- Gadler D, Mairegger M, Janes A, Russo B (2017) Mining logs to model the use of a system. In: 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), pp 334–343, <https://doi.org/10.1109/ESEM.2017.47>
- Ghasemi M, Amyot D (2020) From event logs to goals: a systematic literature review of goal-oriented process mining. *Requir Eng* 25(1):67–93. <https://doi.org/10.1007/s00766-018-00308-3>
- Hynninen T, Kasurinen J, Taipale O (2018) Framework for observing the maintenance needs, runtime metrics and the overall Quality-in-Use. *J Softw Eng Appl* 11(04):139–152. <https://doi.org/10.4236/jsea.2018.114009>, <http://www.scirp.org/journal/doi.aspx?DOI=10.4236/jsea.2018.114009>
- International Organization for Standardization (2010) ISO 9241-210:2010 - Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems
- ISO, IEC 25010, (2011) ISO 25010—Systems and software quality requirements and evaluation (SQuaRE)—system and software quality models. *Iso/Iec Fdis* 25010:2011
- Jiang Y, Hu T, Zhao H (2019) Users' comment mining for app software's Quality-in-Use. In: Sun Y, Lu T, Yu Z, Fan H, Gao L (eds) Computer supported cooperative work and social computing. Springer, Singapore, pp 510–525
- Kim SH, Kim WJ (2019) Evaluation of software quality-in-use attributes based on analysis network process. *Cluster Comput* 22(S1):2101–2114. <https://doi.org/10.1007/s10586-018-2309-6>, <http://link.springer.com/10.1007/s10586-018-2309-6>
- Leopairote W, Surarerk A, Prompoon N (2013) Evaluating software quality in use using user reviews mining. In: The 2013 10th International Joint Conference on Computer Science and Software Engineering (JCSSE), IEEE, pp 257–262, <https://doi.org/10.1109/JCSSE.2013.6567355>, <https://ieeexplore.ieee.org/document/6567355>
- Lethbridge T, Sim S, Singer J (2005) Studying software engineers: data collection techniques for software field studies. *Empir Softw Eng* 10:311–341. <https://doi.org/10.1007/s10664-005-1290-x>
- Liu Y, Li Z, Xiong H, Gao X, Wu J (2010) Understanding of internal clustering validation measures. In: 2010 IEEE International Conference on Data Mining, pp 911–916
- Lu H, Chan WK, Tse TH (2006) Testing context-aware middleware-centric programs: a data flow approach and an rfid-based experimentation. In: Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering, Association for Computing Machinery, New York, NY, USA, SIGSOFT '06/FSE-14, pp 242–252, <https://doi.org/10.1145/1181775.1181805>

- Maguire M (2001) Context of use within usability activities. *Int J Hum-Comput Stud* 55(4):453–483. <https://doi.org/10.1006/ijhc.2001.0486>, <https://linkinghub.elsevier.com/retrieve/pii/S1071581901904860>
- Mirza AM, Khan MNA (2018) An automated functional testing framework for context-aware applications. *IEEE Access* 6:46568–46583
- Müllner D (2013) Fastcluster: fast hierarchical, agglomerative clustering routines for R and Python. *J Stat Softw* 53(9), <https://doi.org/10.18637/jss.v053.i09>, <http://www.jstatsoft.org/v53/i09/>
- Orehovački T, Granić A, Kermek D (2013) Evaluating the perceived and estimated quality in use of web 2.0 applications. *J Syst Softw* 86(12):3039–3059, <https://doi.org/10.1016/j.jss.2013.05.071>, <http://www.sciencedirect.com/science/article/pii/S0164121213001362>
- Osman NB, Osman IM (2013) Attributes for the quality in use of mobile government systems. In: 2013 International conference on computing, electrical and electronic engineering (ICCEEE), pp 274–279, <https://doi.org/10.1109/ICCEEE.2013.6633947>
- Penichet V, Marin I, Gallud J, Lozano M, Tesoriero R (2007) A classification method for CSCW systems. *Electron Notes Theor Comput Sci* 168:237–247, <https://doi.org/10.1016/j.entcs.2006.12.007>, <http://www.sciencedirect.com/science/article/pii/S1571066107000394> (proceedings of the Second International Workshop on Views on Designing Complex Architectures (VODCA 2006))
- Qian Z, Wan C, Chen Y (2016) Evaluating Quality-in-Use of floss through analyzing user reviews. In: 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), pp 547–552, <https://doi.org/10.1109/SNPD.2016.7515956>
- Rana R, Staron M (2015) Machine learning approach for quality assessment and prediction in large software organizations. In: 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), IEEE, vol 2015-Novem, pp 1098–1101, <https://doi.org/10.1109/ICSESS.2015.7339243>, <http://ieeexplore.ieee.org/document/7339243/>
- Rauschenberger M, Schrepp M, Perez-Cota M, Olschner S, Thomaszewski J (2013) Efficient measurement of the user experience of interactive products. How to use the user experience questionnaire (UEQ). Example: Spanish language version. *Int J Interact Multimed Artif Intell* 2(1):39. <https://doi.org/10.9781/ijimai.2013.215>
- Runeson P, Host M, Rainer A, Regnell B (2012) Case study research in software engineering: guidelines and examples, 1st edn. Wiley Publishing, Hoboken
- Saeyes Y, Abeel T, Van de Peer Y (2008) Robust feature selection using ensemble feature selection techniques. In: Daelemans W, Goethals B, Morik K (eds) Machine learning and knowledge discovery in databases. Springer, Berlin, pp 313–325
- Salomón S, Duque R, Montaña JL, Tenés L (2019) A method for analyzing the Quality-in-Use in collaborative contexts. In: Proceedings of the XX International Conference on Human Computer Interaction, Association for Computing Machinery, New York, NY, USA, Interaccion '19, <https://doi.org/10.1145/3335595.3335633>
- Schilit BN, Theimer MM (1994) Disseminating active map information to mobile hosts. *Network Mag Global Internet Wkg* 8(5):22–32. <https://doi.org/10.1109/65.313011>
- Schmidt A (1999) Implicit human computer interaction through context. *Personal Technol*. <https://doi.org/10.1007/BF01324126>
- Seffah A, Donyaee M, Kline RB, Padda HK (2006) Usability measurement and metrics: a consolidated model. *Softw Qual J* 14(2):159–178. <https://doi.org/10.1007/s11219-006-7600-8>, <http://link.springer.com/10.1007/s11219-006-7600-8>
- Souza-Pereira L, Ouhbi S, Pombo N (2021) Quality-in-Use characteristics for clinical decision support system assessment. *Comput Method Programs Biomed* 207:106169. <https://doi.org/10.1016/j.cmpb.2021.106169>, <https://linkinghub.elsevier.com/retrieve/pii/S0169260721002431>
- van der Aalst W (2012) Process mining. *ACM Trans Manag Inf Syst* 3(2):1–17. <https://doi.org/10.1145/2229156.2229157>, <https://dl.acm.org/doi/10.1145/2229156.2229157>
- van der Aalst W, Weijters T, Maruster L (2004) Workflow mining: discovering process models from event logs. *IEEE Trans Knowl Data Eng* 16(9):1128–1142. <https://doi.org/10.1109/TKDE.2004.47>
- van der Aalst WMP, de Medeiros AKA, Weijters AJMM (2005) Genetic process mining. In: Ciardo G, Darondeau P (eds) Applications and theory of Petri nets 2005. Springer, Berlin, pp 48–69
- Vemulapalli SK (2015) Test planning with TMMi practices: assuring the quality by applying continuous test planning methods with TMMi practices, vol 1. CreateSpace Independent Publishing Platform, North Charleston
- Weijters AJMM, Ribeiro JTS (2011) Flexible heuristics miner (FHM). In: 2011 IEEE symposium on computational intelligence and data mining (CIDM), pp 310–317, <https://doi.org/10.1109/CIDM.2011.5949453>
- Yin RK (2008) Case study research: design and methods (applied social research methods), fourth edition. edn. Sage Publications

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.