

The effect of requests for user feedback on Quality of Experience

**Farnaz Fotrousi, Samuel A. Fricker &
Markus Fiedler**

Software Quality Journal

ISSN 0963-9314

Volume 26

Number 2

Software Qual J (2018) 26:385–415

DOI 10.1007/s11219-017-9373-7



Your article is published under the Creative Commons Attribution license which allows users to read, copy, distribute and make derivative works, as long as the author of the original work is cited. You may self-archive this article on your own website, an institutional repository or funder's repository and make it publicly available immediately.

The effect of requests for user feedback on Quality of Experience

Farnaz Fotrousi^{1,2} · Samuel A. Fricker^{1,2} ·
Markus Fiedler¹

Published online: 31 May 2017

© The Author(s) 2017. This article is an open access publication

Abstract Companies are interested in knowing how users experience and perceive their products. Quality of Experience (QoE) is a measurement that is used to assess the degree of delight or annoyance in experiencing a software product. To assess QoE, we have used a feedback tool integrated into a software product to ask users about their QoE ratings and to obtain information about their rationales for good or bad QoEs. It is known that requests for feedback may disturb users; however, little is known about the subjective reasoning behind this disturbance or about whether this disturbance negatively affects the QoE of the software product for which the feedback is sought. In this paper, we present a mixed qualitative-quantitative study with 35 subjects that explore the relationship between feedback requests and QoE. The subjects experienced a requirement-modeling mobile product, which was integrated with a feedback tool. During and at the end of the experience, we collected the users' perceptions of the product and the feedback requests. Based on the users' rational for being disturbed by the feedback requests, such as “early feedback,” “interruptive requests,” “frequent requests,” and “apparently inappropriate content,” we modeled feedback requests. The model defines feedback requests using a set of five-tuple variables: “task,” “timing” of the task for issuing the feedback requests, user's “expertise-phase” with the product, the “frequency” of feedback requests about the task, and the “content” of the feedback request. Configuration of these parameters might drive the participants' perceived disturbances. We also found that the disturbances generated by triggering user feedback requests have negligible impacts on the

✉ Farnaz Fotrousi
Farnaz.fotrousi@bth.se

Samuel A. Fricker
Samuel.fricker@bth.se

Markus Fiedler
markus.fiedler@bth.se

¹ Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden

² School of Engineering, University of Applied Science and Arts Northwestern Switzerland FHNW, 5210 Windisch, Switzerland

QoE of software products. These results imply that software product vendors may trust users' feedback even when the feedback requests disturb the users.

Keywords Quality of experience · QoE · User feedback · User perception · Human factors

1 Introduction

User feedback is essential for managing and improving software products (Pagano and Brügge 2013). User feedback informs software companies in identifying user needs, assessing user satisfaction, and detecting quality problems within a system (Fotrousi et al. 2014). User involvement is an effective means for capturing requirements, and, when feedback is considered in decisions about system evolution, it has positive effects on user satisfaction (Kujala 2003).

A well-known indicator for measuring user satisfaction is Quality of Experience (QoE). QoE is defined as “the degree of delight or annoyance of the user of an application or service” (Le Callet et al. 2012). The QoE indicator is sensitive to the fulfillment of user needs. High QoE values reflect users' enjoyment in using a suitable system (“delight”). Low QoE values reflect users' dissatisfaction in using an unsuitable system (“annoyance”).

QoE is believed to be affected by three factors: the system, the context in which the system is used, and the software users (Reiter et al. 2014). System factors include the properties and characteristics of a system that reflect its technical quality, such as its performance, usability, and reliability (ISO/IEC 25010). System characteristics reflect the Quality of Service (QoS) of a product (Varela et al. 2014). The context reflects the user environment, which is characterized by physical, social, economical, and technical factors. The users, ultimately, are characterized by rather stable demographic, physical, and mental attributes, as well as more volatile attributes, such as temporary emotional attitudes. When interpreting user feedback, all three factors must be taken into consideration, since all of these factors, and not only the software system, affect human emotions (Barrett et al. 2011).

Some studies have empirically evaluated the impacts of systems, their contexts, and human factors on QoE. Most of these studies have investigated the impact of the system factor, including, particularly, the QoS. For example, Fiedler et al. (2010) investigated a generic relationship between QoS and QoE and presented a mechanism for controlling QoE in telecommunication systems. Other studies have investigated the impact of the human factor (Canale et al. 2014) or the context factor (Ickin et al. 2012) on QoE.

By nature, these impact evaluation studies necessitate frequently asking users for feedback on software products, software features, groups of features, or users' actions (e.g., pressing a button). Especially in QoS-oriented studies, such feedback is necessary to interpret the recorded QoS data (Fotrousi et al. 2014). Automated support for feedback requests enables the quick and easy collection of data from a large number of users (Ivory and Hearst 2001).

However, asking for user feedback may disturb users and introduce bias in their QoEs. Research has shown, for example, that users may be disturbed by badly timed (Adamczyk and Bailey 2004; Bailey et al. 2001) or overly frequent feedback requests (Abelow 1993). While research has objectively investigated the impact of feedback requests on users' annoyance, no work has yet subjectively investigated this issue or explored how users rationalize their annoyance. Furthermore, the extant literature has not yet investigated whether the QoE of the product under evaluation is affected by users' annoyance. As a result, we do not know

whether the QoE of a software product may be trusted in cases involving nuisance (Jordan 1998). This uncertainty is particularly important if nuisances are created easily and rapidly.

This paper evaluates whether disturbing feedback requests affect the QoE of a software product. We used a simple probe to collect extensive user feedback, including quantitative QoE ratings and qualitative user rationales. To generate a wide variety of feedback constellations, the probe was triggered randomly as users were implementing a variety of tasks. Some of the users' tasks required little attention, while others required the users to concentrate. The random prompting of different concentration levels for different tasks generated a wide variety of situations in which the users were asked for feedback. At the end of the product usage, a post-questionnaire was administered to collect each user's overall perception of the feedback requests and experience of using the software product. We analyzed the collected data to identify the users' rationales for being disturbed by the feedback requests, to determine whether the feedback requests affected the quality judgment of the software product, and to discover whether the feedback mechanism implemented in the probe was used to provide feedback on the feedback requests.

The main contribution of this paper is an understanding of the extent to which disturbing feedback requests affect users' QoEs, which is an area that has been largely overlooked in previous research. Meanwhile, based on users' subjective reasoning for being disturbed by the feedback tool, we propose a feedback request model, which parametrizes the characteristics of the feedback request. Finally, we discover whether feedback tools can be used to capture the disturbances of the feedback requests. The findings in this study will guide researchers and practitioners in designing user feedback mechanisms to collect informative user feedback, which will assist in enhancing software engineering activities, such as requirement engineering, user-based software development, and the validation of software products.

The remainder of the paper is structured as follows: Sect. 2 provides an overview of the study background and related work. Section 3 describes the research questions, the research methodology, and the threats to validity. Section 4 describes the results and the analysis used to answer the research questions. Section 5 discusses the results. Section 6 summarizes and concludes the paper.

2 Background and related work

User feedback reflects information about users' perception of the quality of a software product. Such perceptions can result in positive feelings such as delight, engagement, pleasure, satisfaction, and happiness or negative feelings such as disengagement, dissatisfaction, sadness, or even combinations of the feelings. The perception differs based on the users' expectations (Szajna and Scamell 1993) in different social contexts (Van der Ham et al. 2014).

User feedback is captured in written, verbal, and multimedia formats directly from users or indirectly through the interpretation of users' activities. A questionnaire is an example of methods gathering data in the written format by questioning user feedback. The user feedback can be collected through a long questionnaire (Herzog and Bachman 1981) capturing more data rather than a short questionnaire (Kim et al. 2008) capturing fewer data but from many users. The short questionnaire can be paper-based or online-based forms. The short questionnaire may also be triggered (Froehlich et al. 2007) regularly or at a particular moment of experiencing a prototype or a released product. The annotating method is another example of written user feedback that users provide comments or rates for snippets of an image (Ames and Naaman 2007) or a video (Fricker et al. 2015) when the users have some opinions to share. The interview (Ahtinen et al. 2009) is an example of methods gathering verbal user feedback. The user

feedback can also be recorded in the form of a multimedia such as an audio or a video. User sketch method (Tohidi et al. 2006) is an example of methods for collecting the activity-based user feedback. A user feedback tool includes one or multiple user feedback mechanism(s) implementing one or multiple user feedback methods respectively for collecting user feedback.

The feedback is collected in the form of qualitative or quantitative measures. A qualitative measure provides a verbal and comparative description of the users' opinions. A quantitative measure is a numerical form of data that is usually referred to a number. Mean Opinion Score (MOS) is a known quantitative metric usually scaled ordinal between 5 and 1 (excellent, good, fair, bad, poor) that subjects assign to their opinion (ITU-T 2003) to measure Quality of Experience (QoE).

Raake and Egger (2014) define QoE as the degree of delight or annoyance of a user, who experiences a software product, service or system. QoE results from the evaluation of the user whether his or her expectations are fulfilled in the light of the user context and personality. Quality of Experience combines the terms *Quality* and *Experience*. *Quality* is an attribute of a software product that refers to the goodness of the software product. *Experience* is an attribute of the user entity that refers to the stream of users' perception including feelings. QoE, as the combination of the two terms *Quality* and *Experience*, is the user's judgment of the perceived goodness of the software as a cognitive process on top of the experience (Raake and Egger 2014).

Along with the development of a user's experience, the perceived quality of the experience is likely to change over time (Karapanos 2013). During the experience development, the user initially gets familiar with the product and learns the product's functionalities. The user excitement and frustration generated in the familiarization phase may affect the QoE of the software product. However, when the user establishes the functional dependency and is attached emotionally to the software product in the next phases (Karapanos 2013), the judgment of the QoE would be more accurate.

The system, the context, and the human factors may also impact on the judgment of users' perception and affect QoE of a software product (Reiter et al. 2014; Roto et al. 2011). The three factors reflect the reason behind a particular perception of a user in an experience. Context and human factors can determine how the system factors impact on QoE (Reiter et al. 2014). As an example, the same software product may leave different quality perceptions when this is used on a small-size touch screen phone in a car or on a personal computer at home.

The system factors point out to the technical characteristics of a software product or services. The functionality of a software product, delay in data transmission, and a content of a media are examples of the system factors. Most of the system factors are relevant to the technical quality of the product or service referring as Quality of Service (QoS). The QoS factors are about the end-to-end service quality (Zhang and Ansari 2011), network quality (Khirman and Henriksen 2002), and suitability of service content (Varela et al. 2014). The QoS factors tends to differ among application domains like: speech communication (Côté and Berger 2014), audio transmission (Feiten et al. 2014), video streaming (Garcia et al. 2014), web browsing (Strohmeier et al. 2014), mobile human-computer interaction (Schleicher et al. 2014), and gaming (Beyer and Möller 2014). As an example in speech communication (Côté and Berger 2014), the quality of the transmitted speech such as loudness, nearness, and clearness may affect QoE.

The context factors refer to the user environment characterized by physical, temporal, economical, social, and technical context factors (Reiter et al. 2014). We can exemplify the physical, temporal, social, and economical factors respectively by an experience occurs in an indoor or outdoor physical environment, in a certain time of day, based on an individual or a group work experience and with a specific subscription type. The technical context factors are the system factors that are contextually related to the software product or service. As an example

of the technical context factors, we can mention the characteristics of the feedback tool and a device that the software product has interconnection with, such as the design layout, screen size, and resolution of the device (Mitra et al. 2011).

The human factors characterize demographic, physical nature, mental nature, and emotional attitudes of human users (Le Callet et al. 2012). The level of expertise and visual acuity of users are examples of demographic and physical factors, respectively. Needs, motivations, expectations, and moods exemplify the mental factors. Among the human factors, the emotion factor has the strongest relationship with experience (Kujala and Miron-Shatz 2013). For example, the user's frustration in an experience may turn into anger, and the pleasant experience makes the user happy. The users' perception of the product's quality is influenced by a variety of emotions (Fernández-Dols and Russell 2003). Therefore, emotions are important factors to be considered while studying QoE.

There are studies that have empirically evaluated the impacts of the system, context, and human factors on QoE of a product or service. Fiedler et al. (2010) investigate a generic relationship between system factors and QoE. The authors present a QoE control mechanism, where MOS is a function of QoS metrics such as response time in the telecommunication area. Ickin et al. (2012) investigated the factors that influence QoE in mobile applications. The study findings reveal the effect of context factors such as battery efficiency, phone features, and cost of application or connectivity on QoE. The study also showed the effect of human factors such as user routines and user lifestyle on QoE. Such impact studies are dependent on a frequent automatic collection of user feedback to interpret the quantitative analytics of the system quality that are also automatically collected. Automatic frequent asking for user feedback may disturb users and may bias the judgment of users for QoE of the software product.

We found no work that evaluated whether the request for feedback would affect QoE of a software product. It is quite imaginable that a feedback request would be a part of the system, context, and human factors that influence on QoE. Triggering the feedback requests, whose functionality may be perceived as a part of the product (i.e., system factor), interrupts the user's task. The interruption that occurs in a certain context like mobile context (i.e., context factor) may disturb the user (i.e., human factor) especially when the user perceives performing the task as the primary and providing the feedback as the secondary task (Adamczyk and Bailey 2004). Disturbing the user by the feedback requests prompts user's perception that causes a sensation or set of sensations toward a negative emotion (Solomon 2008). However, there is a gap in the literature whether the negative emotion caused by the feedback requests would be a factor that influences on users' perception of the software product quality.

Lack of understanding users' rationale for being disturbed by the feedback requests and the relations between the feedback requests and QoE of a software product would make the product owner unable to judge the appropriateness of the collected user feedback. In spite of appropriate feedback requests that motivate users to provide rich effective feedback (Broekens et al. 2010), inappropriate feedback requests may bias the collected user feedback that may affect the reliability and robustness of the decisions, which the product owner makes.

3 Research methodology

3.1 Objectives

The overall objective of this study is to evaluate whether the feedback mechanism affects the feedback obtained about the software product. We aim to determine whether a disturbing

feedback request negatively affects users' perceptions of the software product for which the feedback is requested. Therefore, we look for identifying the subjective disturbing aspects of feedback requests during the collection of feedback for a software product. We study whether the interruption is the only disturbing factor and, if not, seek to identify other possible disturbing factors of a feedback request based on users' reasoning. Finally, we seek to discover whether the feedback mechanism that disturbs users is useful for collecting feedback about such disturbances. Feedback about the disturbances informs product owners of the problems that the users have experienced with the implemented feedback mechanism.

We summarize the objectives as follows:

- OBJ1: Understanding users' reasoning for being disturbed by feedback requests
- OBJ2: Finding out the extent to which disturbing feedback requests affect users' perceptions of a software product's quality
- OBJ3: Understanding whether user feedback is helpful for understanding the disturbances caused by feedback requests

3.2 Research questions

We designed the study to answer the following research questions (RQ1, RQ2, and RQ3), which we mapped to the above objectives (OBJ1, OBJ2, and OBJ3, respectively):

- RQ1: How do users rationalize the disturbance of feedback requests?
- RQ2: To what extent do disturbing feedback requests affect the QoE of software products?
- RQ3: Do users provide feedback about feedback requests?

The overall research efforts help to discover whether the collected user feedback can be trusted even if the users are disturbed by the feedback collection process. The answer to RQ1 determines the aspects of feedback requests that could disturb users. Using these findings, we model feedback requests corresponding to a software product. The model guides the selection of a suitable feedback mechanism to assist researchers and practitioners in collecting unbiased feedback. The answer to RQ2 identifies the relationship between the feedback requests and the users' perceptions of the quality of a software product. The answer to this question helps practitioners ensure that their feedback tools do not influence the quality of users' perceptions of a software product's quality. The answer to RQ3 identifies whether users provide feedback on feedback requests when they are asked to give feedback about the software product. This answer will guide researchers and practitioners in determining whether they can use the user feedback provided for a software product to evaluate the feedback requests generated by the feedback tool.

3.3 Study design

The study used a mixed qualitative-quantitative research approach, which was designed based on multiple embedded case studies (Yin 2014). Figure 1 presents an overview of the study design to address the research questions. For the data collection, a feedback tool was used to request feedback randomly from participants while they were using a software product. At the end of the product's usage, the users' perceptions of the feedback requests and the experiences of using the product were collected through a post-questionnaire. The user feedback about the

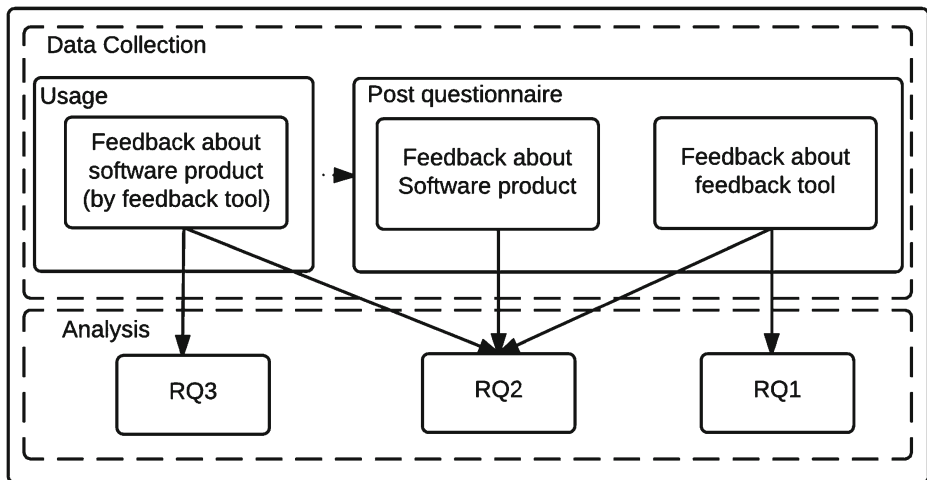


Fig. 1 Overview of the study design

software product during usage, as well as the user feedback that was provided in the post-questionnaire about the feedback requests and the software product, was analyzed individually to answer the research questions.

3.3.1 Selection of the software product and the feedback tool

As the unit of analysis, we investigated individuals' feedback to determine whether the feedback was about the software product or about the feedback requests. All participants in this study used the same software product and the same feedback tool with the same configuration for requesting feedback.

The QoE probe described by Fotrousi (2015) was used as the feedback tool for collecting QoE data from a requirement-modeling software called Flexisketch (Golaszewski 2013; Wüest et al. 2012, 2015). We integrated the QoE probe into the Flexisketch tool. Figure 2 presents a user

Fig. 2 Feedback tool

The screenshot shows the 'fi star Quality of Experience' interface. It prompts the user to 'Please rate your experience with the feature you just used:'. Below this prompt are five yellow smiley face icons representing different levels of experience: 'Excellent' (happy face), 'Good' (neutral face), 'Fair' (neutral face), 'Poor' (sad face), and 'Bad' (very sad face). Each icon has a corresponding radio button below it. At the bottom of the interface, there is a text input field labeled 'Please provide why you feel that way' and two buttons: 'Cancel' and 'Submit'.

interface of the feedback tool. This tool generated requests for feedback continuously and randomly in the middle of users' interactions with a software product. The feedback tool asked participants to rate their experiences with the feature that they had just used and to provide a rationale for their choice. Although the time and frequency of requests could be configured, in this study, a sample configuration was set up that allowed the user feedback to be collected randomly.

3.3.2 Participants

The participants were 35 software engineering students at the graduate level, who were familiar with the concepts of requirement modeling. Attempts were made to achieve as large variations as possible among the participants. The participants varied in age, requirement modeling knowledge, and experiences with a requirement-modeling tool.

3.3.3 Study procedure

From the perspective of the participants, the primary goal of the assigned task was to evaluate hands-on requirement engineering practices. The participants were free to complete the assigned task at anytime and anywhere that they find suitable within the given deadline of 2 weeks. The assigned task in the course was not graded; however, if the students could pass the assignment, they were rewarded with better grades in their two other course assignments. The course assignment was not mandatory, and the students who were not interested in this assignment could skip that and choose alternative an assignment to receive the same reward.

The participants, in their roles as requirement engineers, were asked to translate a real-world requirements walkthrough into a requirements model. The participants had to complete their tasks individually by studying the provided workshop video of a Drug Supply Manager solution, analyzing the discussed requirements, and modeling the requirements.

The video was captured from a requirement engineering workshop, where the participants were discussing the issues related to the distribution of drugs to patients. The issues could impact the safety of the patients. A requirement engineer, two pharmacists, a patient representative, a software developer, a solution architect, a medical device expert, and a barcode technology expert were attending the workshop. In the workshop video, the pharmacists, among other participants, were looking for a solution to be able to trace back the drug packages in the supply chain, using a globally unique barcode.

In the current study, all participants received the same task to model the requirements defined during 15 consecutive minutes of the video. The participants could choose any 15 consecutive minutes of video that they intend to model. The desired models were modeling diagrams such as a use-case, activity, and class diagram. Each participant could model the requirements using even more than one diagram. The participants were free to choose the modeling type and notations. They were told to ensure that the model specified what the stakeholders had defined during the chosen part of the video.

The participants were asked to draw their models in the Flexisketch tool installed on their touch screen devices. Once they accessed an Android tablet, Android smartphone, or a multitouch screen PC, they needed to install the Flexisketch and QoE probes based on the provided guidelines. Alternatively, they were able to use one of the laboratory's tablets to complete their task. The participants received an instruction document providing all required information.

So, each participant used Flexisketch (i.e., a modeling tool), integrated with the QoE probe (i.e., a feedback tool), to model the requirements extracted from the video workshop. While the participants were modeling the requirements, a QoE questionnaire was automatically triggered by the completion of a feature to ask for user feedback. In the feedback tool, the probability of automatic triggering of the questionnaire was set to 10%.

The user feedback was collected across different features of the modeling tool representing a range of complexities, since complexity is a factor affecting users' concentration and task performance (Zijlstra et al. 1999). For example, "save" is a simple feature with low complexity: a user simply presses a button to save the model. By contrast, the "merge" feature for merging two objects of the model is not straightforward and is categorized as a high-complexity feature. When the participants completed the modeling, they were expected to save the model, export it as an image, and then create a short requirement document including this image. The participants were free to complete the assigned task at anytime and anywhere that they find suitable within the given deadline of 2 weeks.

In the last step, the participants were asked to fill in a paper-based post-questionnaire. The questionnaire included two groups of questions about the modeling tool and the triggered feedback requests.

3.3.4 Data collection method

The data collection was performed using open- and closed-ended questions in two steps of the study procedure:

- 1- During the usage of software product: While the participants were using the requirement modeling tool, the feedback tool was triggered randomly (Fig. 2) to collect the participants' QoEs (i.e., ratings of their experiences) with the features that they had just used in the modeling tool. The feedback tool also collected the participants' rationales, which justified the ratings.
- 2- Following usage of the software product: After completing their work with the software (i.e., modeling tool), the participants were asked to answer a paper-based post-questionnaire. In the post-questionnaire, we started with general questions about the users' experiences including whether the participants had previous experience working with Flexisketch, similar requirement-modeling tools and Drug Supply Manager systems. Then, the participants were asked the starting time of the video that they had chosen for modeling and the time spent on the modeling tool. Later, we formulated two questions asking for participant feedback. The first question underlined the disturbance term, as identified in the first research question, to determine users' reasoning for being disturbed. In this question (Q12 in Table 3, Appendix), we also sought to identify the negative influences of feedback requests on modeling activity disturbances. The second question (Q9 in Appendix), asked for the overall user feedback on the software product. The questions about the feedback requests and the software product were formulated as follows:

–Feedback requests –

How good was the QoE probe in minimizing the disturbance of your modeling work?
Bad (1) Poor (2) Fair (3) Good (4) Excellent (5)

Please explain why you feel that way: _____.

–Software product –

How good was Flexisketch as a tool for modeling requirements?

Bad (1)Poor (2) Fair (3) Good (4)Excellent (5)

Please explain why you feel that way: _____.

To design the two questions, we used a 5-point Likert scale, including a mid-point (i.e., Fair (3)), to avoid negative ratings in the absence of a middle point (Garland 1991).

3.3.5 Data analysis method

The questions RQ1 and RQ3 were answered using a qualitative content analysis approach. To answer RQ2, which is the core research question of this study, we triangulated the analysis using content analysis, pattern matching, and statistical correlation analysis methods. The statistical descriptive analysis was also used to support discussion.

Content analysis The analysis procedure followed inductive and deductive content analysis approaches (Elo and Kyngäs 2008). The inductive approach was conventional, with the objective of coding data freely to generate information, and the deductive approach was based on the use of initial coding categories, which were extracted from the hypothesis, with the possibility of extending the codes (Hsieh and Shannon 2005).

Inductive content analysis Since prior knowledge on the phenomenon was limited, we performed an inductive content analysis to find answers for RQ1 and, partially, RQ2. The study started with the collection of qualitative feedback, which participants provided for the feedback requests (issued by the QoE probe) and the software application (Flexisketch) in the post-questionnaires. The analysis was conducted in the following four steps:

- Step 1 —Perform initial coding: Participants’ quotes, which referred to their qualitative feedback, were analyzed separately. For each quote, we underlined all terms that could have some relation to reflections of participants’ experiences or the impact of the software product on the participants’ perceptions. We then read each quote again and wrote down all relevant codes. We repeated the process one by one for all quotes.
- Step 2 —Form final codes: We grouped the initial codes to form final codes based on shared characteristics, which put different codes in the same categories. For example, the vocabularies that were synonyms or had the same or similar stems, meanings, or relevancies were organized in the same category of codes. Observations in other quotes also assisted in the creation and renaming of the final codes. Such groupings reduced the number of codes and increased our understanding of the phenomenon. As examples, the initial codes of “time-to-time,” “every tap,” “keep pop up,” “too often pop up,” and “frequently” all referred to the frequency of the feedback requests; these formed the final code “frequent request.”
- Step 3 —Form categories: We created categories based on a general overview of the final codes. The categories were formed based on the patterns that we recognized within the quotes and, in some cases, our interpretations of the quotes’ meanings (Potter and Levine-Donnerstein 1999). Categories merged into a higher level when the merging made sense. The categories were developed independently by the first and second

authors, and the final categories were decided in a joint meeting based on a “chain of evidence” tactic (Yin 2014). The correctness of the categories was later evaluated by the third author. Then, we organized the final categories in a matrix, comprising the connections between the participants’ quotes and the categories used by the final codes as elements. As explained in Sect. 4.1, the content analysis concluded the matrix by including three categories: *kind of user perception*, *consequence of disturbance*, and *characteristics of feedback requests*. *Characteristics of feedback requests* were also divided into the sub-categories of *task*, *timing*, *experience phase*, *frequency*, and *content*.

- Step 4 —Perform abstraction: In the last step, based on the extracted categories, we performed an abstraction that led to a generic model. We interpreted and discussed this model based on the quantitative data of the given QoE ratings for the feedback requests and the software product.

Deductive content analysis To answer RQ3, we performed a deductive content analysis. The results of this section could also support RQ2. The research was initiated with the following hypothesis formulation:

H: Participants provide feedback for the feedback requests during their usage.

Then, initial categories of codes were organized. The qualitative feedback that participants provided during the usage was coded to test the hypothesis. The analysis was conducted in the following three steps:

- Step 1 —Development of an analysis matrix. We developed a matrix to connect the participants’ quotes and the initial categories of codes. The connections were filled with the coding data provided in step 2. We used an unconstrained matrix with the possibility to extend the categories during the data coding. We expected that participants would provide feedback in the categories for *feedback request*, *software product attributes*, and *device attributes*. The first category was defined based on the hypothesis, and the next two categories were factors affecting the QoE of a product, as identified before through the inductive content analysis.
- Step 2 —Data coding: We reviewed all comments and coded in relevance to the defined categories in step 1. Although we aimed for an unconstrained matrix, no new categories were recognized during the coding. However, new sub-categories were identified. For example, for the software product attributes, we found a *performance* sub-category as a quality attribute that had not been identified during the inductive content analysis.
- Step 3 —Hypothesis testing: The coded matrix was a good tool for easily testing the hypothesis. Exploring the codes identified whether any feedback was available about the feedback requests.

Pattern matching Part of the analysis to answer RQ2 used a pattern-matching analytical technique (Yin 2014). In the pattern matching, a hypothesis to be tested—a so-called predicted pattern—was compared with the observed patterns that were concluded empirically.

Section 4.2.1 shows the results of the pattern matching research. We performed the pattern matching in the following four steps:

Step 1 —Formulate hypothesis: We formulated the research hypothesis in alignment with the research question. The research hypothesis is referred to as the predicted pattern during the study. This pattern was formulated as an *if-then* relation, where the *if* statement is the *condition* and the *then* statement is the *outcome*. We used an independent variable design with the “sufficient condition proposition” (Hak and Dul 2009), meaning that the *outcome* of the pattern is always present if the *condition* defined in the proposition is present. Therefore, if alternative patterns in the absence of the condition are confirmed, the hypothesis is disconfirmed. The hypothesis was, thus, formulated as follows:

H-P: The Quality of Experience (QoE) of the software product is always perceived to be bad if the feedback request disturbs the participant.

The *outcome* (i.e., “The Quality of Experience (QoE) of the software product is perceived to be bad”) was always present if the *condition* (i.e., “if the feedback request disturbs the participant”) was present.

Step 2 —Select appropriate cases: To investigate the hypothesis, we look for alternative patterns involving the *outcome* in the predicted pattern (i.e., “the QoE of the software product is perceived to be bad”). The absence of the outcome was the criterion for selecting cases. We chose cases in which the participants rated the QoE of the software product as good and then, from among these selected cases, looked for the presence or absence of the *condition*, as defined in the predicted pattern (i.e., “if the feedback request disturbs the participant”).

Step 3 —Observe patterns to test the hypothesis: We observed the conditions in the selected cases and then formulated the observed patterns as the result of this step. We conducted our observation in a matrix with two dimensions for the QoE of the software product and the QoE of the feedback request. We also used the participants’ justifications in the qualitative feedback relevant to the selected cases to increase the reliability of the observations.

Step 4 —Formulate test results. This step reported the confirmation or disconfirmation of the hypothesis. If the investigation could show observed patterns in the absence of the condition, it would be sufficient to disconfirm the hypothesis.

3.3.6 Statistical analysis

We used a correlation analysis to measure the relationships among the observed variables. As part of RQ2, we used the Pearson and Spearman correlation coefficient methods to investigate the linear and monotonic relationships between the QoE of the software product and the QoE of the feedback request, respectively. Furthermore, throughout the study, descriptive analysis statistics, such as average and median, were used to provide supportive information for the discussion.

3.4 Threats to validity

Following the classifications in the qualitative study (Yin 2014) and the content analysis (Potter and Levine-Donnerstein 1999), we analyzed threats to validity. We also addressed the threats regarding student participation (Carver et al. 2003).

Reliability We interpreted reliability as the rigor and honesty with which the research has been carried out. Threats to reliability affect the repeatability of the study (i.e., the ability to run the study again and achieve the same results). To address potential threats to reliability, we developed a study protocol, collected all data in a study database, and used triangulation as the main strategy for answering the research questions (Golafshani 2003). We performed data triangulation by collecting data during and after the use of the application and considered both quantitative and qualitative data. We combined quantitative and qualitative approaches for the data analysis. The second and third authors of the study reviewed the results and the analysis performed by the first author.

A key concern was the coding of the collected qualitative user feedback (Potter and Levine-Donnerstein 1999). To mitigate coding problems, the first author documented the design of the content analysis and developed detailed coding rules in a guideline that ensured that the other researchers would make the same decisions when selecting codes. The authors reviewed the coding and discussed conflicting coding results. Inaccurate punctuation and mistyped words sometimes changed the entire meaning and interpretation of a user's feedback. In cases in which the user's intended meaning was unclear, the quote was removed from the analysis.

Internal validity The threat is the extent to which the results may have been biased by confounding factors. One of the risks in this study was that the users might be disturbed by another stimulus, such as their devices or the physical environment, rather than by feedback requests. We captured the causes for such disturbances using the qualitative feedback received from the users during and after their experiences with the software product. Capturing these factors assisted us in distinguishing them during the analysis.

One factor that could have biased the entirety of the study results was the participation of students. The participating students could have felt incentivized to provide the results that their teacher(s) expected. To mitigate this threat, the first author, who executed the study, was not involved in the teaching of the concerned course. In addition, the assignment was optional for the students and not graded. The participants could voluntarily select either this assignment or another alternative assignment of comparable effort and difficulty. The participants could also opt out at any moment and choose to do another assignment.

Insufficient information for the participants is another potential confounding factor, which could affect users' disturbance. To mitigate this threat, we informed the participants that the task was part of a research project and explained the roles of the QoE probe and the Flexisketch. The participants also had access to the post-questionnaire in advance. Furthermore, we informed the participants about the monitoring of their usage data, which would be kept anonymous. Such monitoring data could be used to enhance internal validity and, to some extent, replace the actual observation of the participants as they performed their tasks.

External validity External validity concerns the ability to generalize the results obtained from a study. In this study, fourth-year software engineering students participated as subjects. They did not have knowledge of user feedback research, but they had been introduced and

extensively trained in software engineering, including in theory and team projects. In a comparable rating and feedback study, Fricker et al. (2015) could not identify discernable differences between student ratings and ratings of industry subjects and noted that their positive and negative feedback were congruent. Similarly, Höst et al. (2000) could observe only minor differences in the conception, correctness, and judgment abilities of last-year students and professionals. Not only the number of analysis units (i.e., user feedback) but also the number and kind of case (i.e., modeling of Drug Supply Management requirements) are important for generalizability.

The findings contribute toward generalization as they are applicable to the cases with similar characteristics. For instance, the findings can be applied to the cases where the users require a high level of creativity and interaction with the software (e.g., Adobe Photoshop modeling software) to perform their tasks. However, as Kennedy (1979) recommends for a single case, we leave the judgment for generalizability of the case to the practitioners, who wish to apply the findings, to determine whether the study's case is applied to their own case. In the end, to corroborate further generalization of the research results to other settings, similar research studies with other types of subjects and different software products should be conducted.

Construct validity Construct validity reflects whether a study measures what was supposed to be measured. The risk in this research was that the participants might provide feedback without really experiencing the requirements modeling product or that, in the event of this experience, they might not provide sufficient evidence in their feedback to answer the research questions. To mitigate the threat of students providing feedback without experiencing the product, the study protocol forced the participants to report the results they had achieved with the software product. In this protocol, we also established a chain of evidence to ensure that the categories were defined correctly during the content analysis. We also reported the analysis by making explicit (e.g., by reporting quotes at appropriate places) how our answers to the research questions were based on the data we collected.

Furthermore, in real environment, users could perform such tasks within few hours. However, the time pressure on the participants for performing their tasks could be a risk that might result in reducing the quality of the answers (Sjøberg et al. 2003). The time pressure might make the participants more anxious and lead different judgment (Maule et al. 2000) on the given user feedback. To reduce the threats to validity, the design of our study allowed the participants to perform their task in a relax time within 2 weeks.

The complexity of tasks is another threat to construct validity as different complexity might cause a different level of concentration and task performance (Zijlstra et al. 1999). Therefore, we considered several variations in our design to cover a wide spectrum of complexities from low-complexity (e.g., pressing a button, or watching a simple and understandable video) to high-complexity (e.g., merging two objects) tasks.

4 Results and analysis

The results show that the 35 study participants were from Europe (42.9%), China (42.9%), Africa (8.6%), and the Middle East (5.7%). Of the participants, 22.9% were female, and 77.1% were male. All were aged 23 to 37 years old, with the mean of 25.7 years. Table 1 gives an overview.

Table 1 Distribution of participants: country (left) and gender (right)

	Country					Gender		
	Africa	China	Europe	Mid-East	Total	Male	Female	Total
Frequency	3	15	15	2	35	8	27	35
Percentage	8.6	42.9	42.9	5.7	100.0	22.9	77.1	100.0

None of the participants had previously experienced the requirement modeling tool and Supply Manager applications. To conduct the task, the participants used several models of Android tablet and Android smartphone, and no use of a multitouch screen PC was reported. They participants reported their duration of using the requirement modeling tool. The responses ranged from 2 h to 4 days. From the answers collected during the post-questionnaire, the participants rated the feedback requests and the software product in the range of Good (4) to Bad (1), with a median of Fair (3). No Excellent (5) rating was collected.

Table 2 shows the number of submitted feedback on the software product or feedback tool. According to the usage log, 25 participants provided feedback on the software product during runtime. Although the 10 remaining participants had seen the feedback tool at least 2 times while performing their task, they did not submit any feedback, i.e., They declined the feedback requests. Based on the instructions given to the participants, the participants were able to decrease the likelihood of triggered feedback requests or deactivate the feedback requests. Ten user feedback less on software products means that we missed some qualitative feedbacks at the feature level, which was not critical of our analysis.

The participants submitted a total of 441 QoE ratings and 60 valid feedbacks that justified these ratings during product usage (64 feedback rationales were provided, in which four were made of meaningless letters or symbols). The QoE ratings were distributed in the range of Excellent (5) to Bad (1) (i.e., Excellent (5) 70, Good (4) 133, Fair (3) 77, Poor (2) 89, Bad (1) 72 feedback). The users provided rationales when they had both positive and negative perception (i.e., Excellent (5) 7, Good (4) 13, Fair (3) 8, Poor (2) 22, Bad (1) 10 feedback). The median of QoE ratings with rationale and without rationale (i.e., Poor (2) and Fair (3), respectively) shows that the participants have more justified the feedback ratings when they had a negative perception.

All participants returned the post-questionnaire. Thirty-three provided rationales for the ratings, while two did not. Figure 3 gives an overview of the QoE ratings of the software product and the QoE rating of the feedback requests that have been collected from the post-questionnaire. As presented in the top-left chart, the perceived quality of the feedback requests was less than the perceived quality of the software product. Since the scale defined for the QoE rating was the Opinion Score, an ordinal scale, we calculated the median as the measure of

Table 2 Number of submitted feedback

	Feedback on software product (usage log)		Feedback on software product (post-questionnaire)		Feedback on Feedback tool (post-questionnaire)	
	QoE rating	Rationale	QoE rating	Rationale	QoE rating	Rationale
Participants	25	25	35	33	35	33
Total feedback	441	60	35	33	35	33

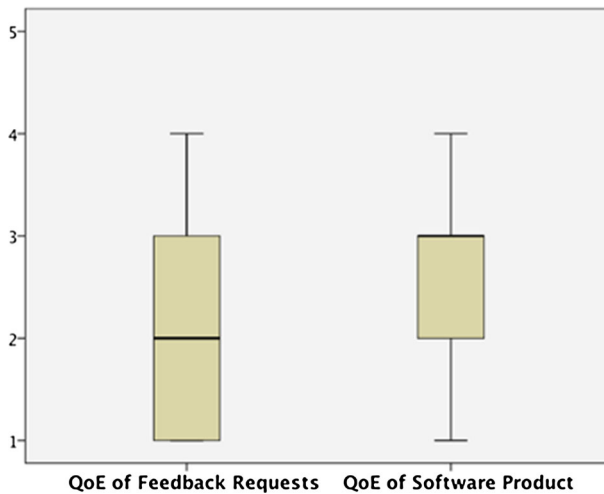


Fig. 3 Distribution of the participants' ratings for the QoE of the feedback tool and the QoE of the software product according to the post-questionnaire. The QoE scales reflect the Opinion Score from Bad (1) to Excellent (5)

central tendency: $\text{Median}_{\text{QoE of feedback requests}} = 2$, $\text{Median}_{\text{QoE of software product}} = 3$, equivalent to the Poor (2) and Fair (3) levels, respectively. The levels show that the participants were disturbed by the feedback requests. The software product was appreciated better, even though clearly not excellent. According to 5-point Likert scale used in designing questionnaires as well as our non-parametric statistical test, levels 2 and 3 are significantly different. The level 2 refers to unsatisfactory perception, while 3 shows the mid-point referring to uncertain perception.

Figure 4 shows an analysis of the influence of cultural diversity on QoE. For the majority of countries, $\text{Median}_{\text{QoE of feedback requests}} = 2$. The Chinese participants differed with a median QoE rating of 3. In addition, $\text{Median}_{\text{QoE of software product}} = 3$, except for the participants from Middle East, who rated the software product to be Good (4). No country reversed the results shown in Fig. 3, suggesting that cultural differences had no effects that would reverse the study results. The participants were disturbed more by the feedback requests than by the software product, and the Good (4) ratings were likely due to the small number of participants.

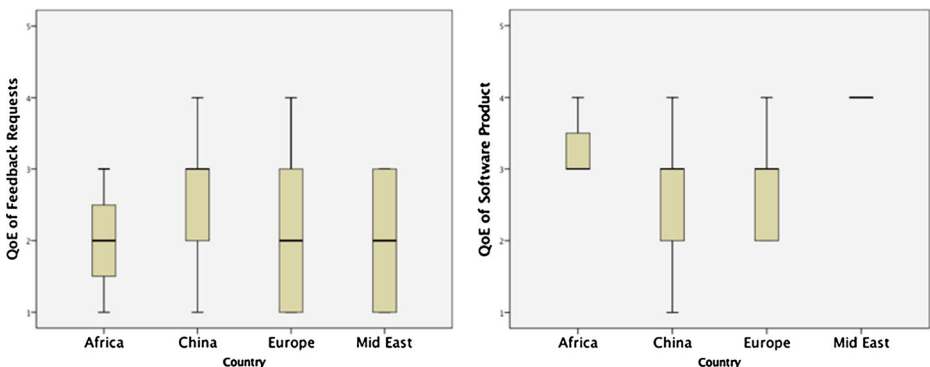


Fig. 4 Distribution of the participants' ratings for the QoE of the feedback tool and the QoE of the software product according to the post-questionnaire. The QoE scales reflect the Opinion Score from Bad (1) to Excellent (5)

4.1 Modeling of feedback requests

Based on the qualitative analysis below, we modeled a feedback request according to the users' reasoning for the disturbance level of the feedback tool. As presented in Eq. 1, our model defines a set of feedback requests for each product (p) and user (u). Each product (p) and user (u) belongs to the set of available products (P) and users (U), respectively. The FRs are a set of five-tuple variables referring to the user task (ta), the timing of the feedback request within a task (ti), experience-phase (e), the frequency (f) of the feedback request, and the content (c) of the feedback request.

$$FR = \{(ta, ti, e, f, c) \mid p \in P, u \in U\} \quad (1)$$

Model for user feedback requests developed from the inductive content analysis.

The *user's task* (ta) refers to the type of activity the user was performing with the software product when a feedback request was issued. The important user's tasks were modeling requirements and managing the model, e.g., by saving it. The *timing* (ti) is the moment within the user's tasks when the feedback request has been issued. The *expertise-phase* (e) refers to the user's stage of understanding and mastery of the product at the moment of the feedback request. For example, in a modeling tool, the experience-phase can refer to the learning period at the beginning of an experience. The *frequency* (f) of a feedback request refers to the maximum number of times that feedback is requested in a specific timing and expertise-phase relevant to the task. The *content* (c) refers to the questions included in a feedback request. The values for any of these variables might drive the perceived disturbances.

The feedback request model is a result of the inductive content analysis described in the Content analysis in Sect. 3.3. During the content analysis, we identified that the participants' quotes referred to three main categories: *kind of user perception*, *consequence of disturbance*, and *characteristics of feedback requests*. *Characteristics of feedback requests* could be divided into the sub-categories of *task*, *timing*, *experience phase*, *frequency*, and *content*. Each of the variables ta , ti , e , f , and c reflect one of these identified categories.

The categories were identified based on the users' subjective reasoning for disturbing feedback requests. The following disturbing issues were identified:

- a feedback request that was interrupting a user task;
- a feedback request that was issued to the user too early before the user experienced enough and understood the product;
- a feedback request that was issued too frequently; and
- a feedback request with apparently inappropriate content.

The first three factors were mapped to the timing within a task, the expertise-phase, and the frequency of the request for the task. The fourth factor concerned the content of the feedback request and the functionality provided to allow the user to give feedback. In the following, we show the users' reasoning for the disturbance of feedback requests. These are supported by the participants' quotes (written in italic fonts within quotation marks) to improve the credibility of the discussion.

The participants perceived that the tasks were interrupted at the macro-, meso-, and micro-levels. The participants provided their rationales for being disturbed in macro-level (e.g., modeling), meso-level (e.g., drawing diagrams or working with features, such as locating UML elements), and micro-level (e.g., performing an action, such as a click). Although the

interruption was generated at the meso-level (end of using features), however, some participants perceived the interruption in the micro-level. We argue that this incorrect perception could be due to less than a second delay of showing the feedback form. Also, another reason could be due to fragmentary user's action, where the system recognizes it as the end of using the feature (e.g., releasing the mouse button in the middle of drawing a line that the system identifies a new line). The interruption was more disturbing when the task required concentration.

"... Let me put an example, if I want to put down a square, add a text and put the text in the square, then I don't want to be disturbed while doing that. I don't mind if QoE Probe disturbs me after I've done this few concatenated steps, but this was not the case. It kept interrupting me ..."

"... sometimes you could lose a bit track of a thought process and when that happened it was quite annoying ..."

"It was annoying as it asked while I was drawing and then only half the line was finished."

A feedback request that came too early before the user had the chance to really understand the product disturbed participants, because the user expertise of whom received early feedback requests was still in the learning phase and familiarization with the product. In response to an early feedback request, a participant was unable to judge a product, feature, or action, and the judgment risked not reflect sufficiently complete, accurate, or correct feedback.

"I think it should leave at least a week for users to experience the app[lication], then they will have a better understand and experience of the Flexisketch."

The frequent feedback that was requested at multiple times during a task disturbed users. Frequent requests increased perceptions of disturbance when the same feedback requests were repeatedly asked for the same feature or action. Sometimes, the feedback request was issued so frequently that the participants perceived that the main goal of the study was to disturb them.

"Way too intrusive as it came up way too often."

"I had to write feedback multiple times for some features, while for others—never."

"It felt as if the entire purpose of the QoE Probe was to disturb my modeling work."

The feedback that was requested frequently encouraged participants to discover the mechanism behind triggering the feedback questionnaire. Due to the ambiguity of this mechanism, the users could even lose sight of the main objective of the feedback requests.

"It was really disturbing, it disappears after a while, but again I don't know it was on me or the system that solved it."

"To be honest I do not know why I need to install it."

The content of a feedback request was also mentioned as a disturbing factor, although its impact level (relevant to participants' ratings) was not considerable. The participants complained that the feedback requests had limited functionalities.

"The function [of feedback requests] is quite limited ..."

"... the functions [of feedback requests] are not as good as I wished."

Not only did unsuitable feedback requests disturb the participants, but the participants also expressed feelings of annoyance and disengagement.

“The interruptions were too many and not welcome.”

Such feelings consequently affected the quality of the provided feedback and the quality of the participants’ performance on the main tasks in the experience. Disturbed participants might be discouraged regarding the provision of feedback, or they might provide inaccurate feedback. Furthermore, participants’ task performance was reduced when the participants lost track of their thoughts and forgot their next tasks due to the interruptions. Such disturbances encouraged participants to take action, such as uninstalling the feedback tool.

“Since it pops up in the middle of working on a diagram, you don’t have much will and time to think truly carefully before answering. This probably means that the results aren’t as accurate as one could wish for.”

“...I felt it disturbing most when the QoE came up in the middle of me having an idea I needed to model. By the end of my feedback, I almost forgot what I was about to model, which was for me very annoying. ...”

“it disturbed my modeling quite a lot I was almost tempted to uninstall it.”

The majority of participants who mentioned higher levels of disturbance or efforts to take give-up actions, such as uninstalling the feedback tool in their quotes, rated the QoE of the feedback tool as a 1 or a 2. However, the participants rated the QoE of the feedback tool as a 3 or a 4 when they did not recall a high disturbance level; instead, these participants used occasional adjectives, such as “some” or “sometimes,” to describe their disturbances due to frequent/interruptive feedback requests.

4.2 The effect of disturbing feedback requests on the QoE of a software product

Disturbing feedback requests have a negligible impact on participants’ perceptions of the quality of software products. The QoE of a software product does not correlate with the disturbance ratings of the feedback requests. The results show that the QoE of a software product might not be degraded even by participant feelings of disturbance related to the feedback requests. Even though the feedback request characteristics discussed in Sect. 4.1 might disturb the participants, the quality of the software (i.e., 97% of the quotes) and the context such as the device quality (i.e., 42% of the quotes) served as the focal points of arguments to justify the QoE ratings.

The study’s results were triangulated with three individual analysis methods to facilitate studying the phenomenon from different angles. This section details these analyses.

4.2.1 Was the QoE of the software product bad when the feedback request disturbed participants?

A disturbing feedback request did not necessarily indicate that participants would negatively evaluate the QoE of the software product. In other words, the disturbances caused by the feedback requests did not always result in a bad experience of the software product. This statement was concluded as the result of disconfirming the predicted pattern we identified for this study, as follows:

P: The Quality of Experience (QoE) of the software product is always perceived to be bad if the feedback request disturbs the participant.

The analysis showed that the QoE of the software product was perceived to be good even when the feedback requests disturbed the participants. As explained in Pattern matching in Sect. 3.3, to test the pattern P, we explored the following two possible alternative patterns within the participants' quotes.

AP1: The Quality of Experience (QoE) of software product is perceived to be good, if the feedback request disturbs the participant

AP2: The Quality of Experience (QoE) of software product is perceived to be good if the feedback request does not disturb the participant

We evaluated the alternative patterns AP1 and AP2 using the participants' ratings that were collected via the post-questionnaire and the feedback tool after and during the usage respectively. Figure 5 presents the participants' ratings for the feedback requests and the QoE of the software product, collected from the post-questionnaire. The x-axis indicates the ratings of the feedback requests, and the y-axis shows the quality ratings for the software product.

The observation of the alternative patterns AP1 and AP2 in the matrix in Fig. 5 showed that when the QoE of a software product was rated Good (4) (there were no Excellent (5) ratings), in 37% of the cases, the feedback requests disturbed the participants (i.e., rated Bad (1) and Poor (2)); these results aligned with AP1. In the same scenario of QoE rating, 63% of the feedback requests did not disturb the participants (rated Fair (3) and Good (4)); these results aligned with AP2. The observation of AP1 contradicted the predicted pattern and, thus, disconfirmed it.

The similar observation was also found in the participants' qualitative motivations. For example, one participant liked the product and rated as a 4 with this rationale:

"It was fun in creating the diagrams because I was lying on my bed and creating the diagrams by using it. I like it."

However, the same participant was disturbed by the feedback requests, rating these as a 1, with the following rationale:

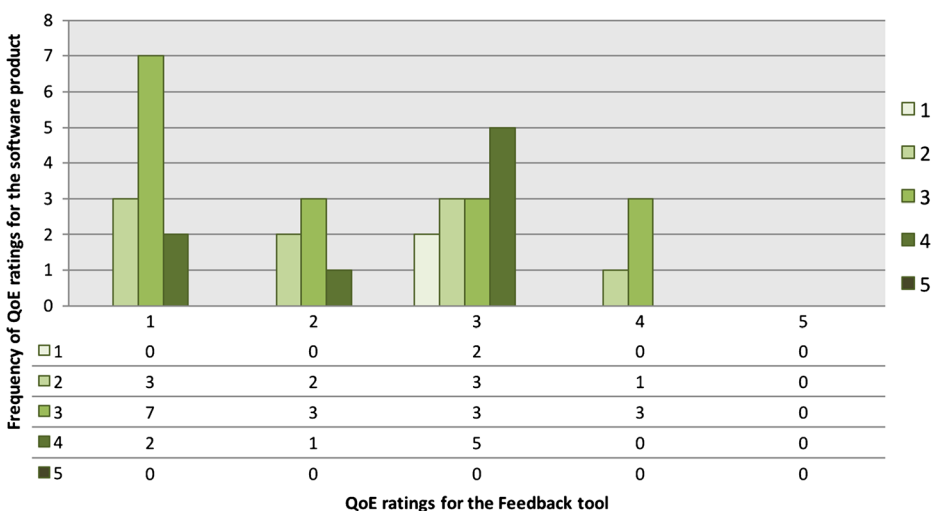


Fig. 5 Distribution of the QoE of the software product per each QoE of the feedback request (data series reflect the QoE of the software product). Data is collected via the post-questionnaire

“I was just fed up from this QoE because it was disturbing a lot while making diagrams.”

The pattern *API* could also be seen within the feedback collected from the feedback tool. There was one case in which the QoE of the software product was perceived as Excellent (5), but the participant complained about the disturbing feedback requests. The observation of *API* disconfirmed the *PI*.

The examples and the descriptive statistics showed that a disturbing feedback request did not necessarily imply a bad QoE of the evaluated software product.

4.2.2 Was the QoE of the software product statistically related to the QoE of the feedback requests?

With the provided ratings, we could not find any evidence to show a dependency between the quality ratings of the disturbing feedback request and the software product.

A correlation analysis was performed to measure the relationship between the participants' ratings given to the feedback request and the quality of the software product, as collected through the post-questionnaires. The results showed a very small, almost non-existent correlation (i.e., Pearson analysis [$= -0.056, n = 35, p > .001$] and Spearman analysis [$= -0.032, n = 35, p > .001$]). The analyses indicated a lack of linear and monotonic relationships between the participant ratings for the quality perception of the feedback request and the quality perception of the software product.

4.2.3 Were the QoEs of the software product justified with arguments about disturbing feedback requests?

The QoEs of the software product were justified with arguments about factors other than the disturbing feedback requests. The software characteristics and the experiencing context were the focal points of these arguments.

The participants also provided arguments about the quality of the software product and the experiencing context (e.g., device characteristics) that respectively addressed 97 and 42% of all feedback for justifying the QoE of a software product in the post-questionnaire. Among this feedback, no participant used any characteristics of a feedback request to justify poor QoE ratings for a software product. We could argue that the two separate questionnaires at the end of usage—one for the QoE of the feedback requests and one for the QoE of the software product—allowed the participants to distinguish the feedback tool from the software product. Therefore, the participants provided justifications for the QoE ratings of the software product regardless of the ratings they had given for the feedback requests.

However, the feedback collected by the tool during the usage could not provide enough evidence to justify the QoE ratings. Although four feedback quotes out of 64 were related to the feedback requests, these quotes did not include interpretations of the QoE ratings. For example, one participant, who complained about the interruptions of feedback request two times, gave Poor (2) and Excellent (5) ratings to the QoEs of the same feature.

Software quality attributes were the most common factors that the participants used to justify their ratings. *Functionality*, *usability*, *learnability*, *portability*, and *performance* were the quality attributes that the participants most commonly used for these justifications.

Functionality and usability of software features were the most common categories of feedback. Interestingly, of the 33 rationales provided for rating the software product in the post-questionnaire, 19 feedback rationales addressed the software's functionality and 16 feedback rationales addressed

its usability categories. Furthermore, out of 60 total feedback quotes, the feedback tool collected 36 and 16 feedback quotes about the functionality and usability categories, respectively.

The participants gave feedback about crashes and errors in product functionality. The participants were also disturbed by non-conformities with the expected functionality. They reported issues with some features that did not work properly or were not successful in fulfilling their expectations.

“...The zoom function did not zoom text as I wanted, making the model very wired, and the lines which I draw between actor/stakeholder to circles did not connect properly, annoying me as well.”

“Flexisketch seems to lack the following [functionalities]: Arrow heads for directions, copy and, paste mechanisms, screen resize functionality, Eraser functionality, Scrollbar functionality, code generation functionality...”

“Because the poor functionalities, and strong dependence on the device (for now it can only run in android system) that don’t flexible for the user.”

The participants provided feedback on the usability of features, particularly with regard to their ease or difficulty of use. Some of the participants failed to recognize the software product as user-friendly, while others admired its simplicity.

“It was okay as it had all of the features as you need, but it wasn’t user-friendly at all at least not on my phone....”

“It’s fair because the application is very simple and easy to use, but it also has many limitations.”

“The program was literally unusable in horizontal view which was a huge set-back on my smartphone. Some options disappeared while being in horizontal view.”

The participants also provided feedback on the performance of the product in relation to an overly long response time.

“The response is too slow.”

“It takes some time but maybe because of the touch screen quality.”

Even when the participants watched or read the instruction guidelines, they still faced learnability issues.

“I watched the instruction video, but I still don’t know how to draw specific items, like arrows.”

From the participants’ points of view, the context was perceived to be a part of product attributes. The participants also provided arguments about device attributes (e.g., mobility characteristics, screen size, touch-based functionality, and the operating system of the device) as context factors to justify their ratings. The participants complained about using the product on small-sized screens.

“I think it is useful when I watch the tutorials, but when I really use it, I found it is really not suitable for mobile phone.”

“Too less kind of elements can be chosen to draw a diagram. Not easy to use on a small-screen mobile device.”

“Because the poor functionalities, and strong dependence on the device (for now it can only run in android system) that don’t [make it] flexible for the user.”
“This app can be installed in mobile with Android system, which is easy to carry and edit.”

4.3 Feedback about feedback requests

Of the 64 feedback collected by the feedback tool, only four feedback rationales from two participants, representing 6% of the total qualitative feedback, concerned the feedback requests. The four feedback rationales represented only 0.9% of the total participant experience ratings. Most of the participants did not provide qualitative data (85%); instead, they only rated their experiences.

A few participants gave feedback concerning disturbing feedback requests. Experience interruptions and inappropriate question timing were two categories of disturbing feedback that the participants mentioned.

“Do not interrupt during drawing!”, “This forum really disturbs.”
“Because I am getting the rating without even getting a chance to finish my sketch,”
“The same as a previous comment.”

Exploring all of the ratings and the feedback revealed that a majority of participants did not provide qualitative feedback; however, those that did provide such feedback primarily pointed to the quality of the software and the context (as discussed in Sect. 4.2.3). The feedback was provided both to complain about and to admire the quality of the software product. However, the feedback about the feedback requests was only issued in the case of disturbance. When no issue was found, the participants did not admire the feedback requests.

Although the majority of the participants did not offer feedback on the feedback requests, the few received feedback was still useful for obtaining an accurate understanding of the problems that the participants experienced with the feedback tool.

5 Discussion

According to the findings of our study, feedback requests that are interrupting a user’s task, that are too early for what a user knows about the product, that are too frequent, or that are with inappropriate content may disturb users. The first factor is congruent with earlier research. The second and third factors are not surprising, although previous studies did not address them as the disturbing factors caused by feedback requests, but the latest factor is new.

A request for feedback that interrupts a user during a task affects the user’s task and, as a consequence, the user’s experience negatively (Bailey et al. 2001). In our study, such interruption was particularly problematic during a modeling task, which required particular attention. The interruption generated frustration because the user has to remember the task and how to proceed toward completion of the task. As suggested by Adamczyk and Bailey (2004), it is crucial to find the best moment of interruption and thereby reduce the extent of disturbance.

A feedback request that is issued to a user before he is familiar with the product is perceived to be disturbing. Such familiarization phase is important as a user needs to establish knowledge of the product and how the product is to be used. Some users do not accept the product initially, but they have better perception over prolonged use (Karapanos 2013). Also, the familiarization is accompanied by a change of thoughts, feelings, and expectations about the product (Karapanos 2013). An initially positive judgment of a product may become negative or vice versa. Thus, when confronted with a feedback request that is too early, the user may be unable to judge the product or may give feedback that is incorrect. According to our results, the knowledge about this inability is felt by the user as a disturbance. It is important to match the timing of a feedback request with the user's knowledge about what the request is seeking feedback for.

A rapid re-occurrence of requests for feedback disturbs users. This insight is interesting because it extends the understanding of how temporal aspects of feedback requests affect the product user. Even well-timed requests for feedback may be disturbing if they are issued too frequently. Especially disturbing is the repetition of requests if the user has already submitted feedback that was well thought through and well formulated. It is a must for a feedback mechanism to consider the history of the feedback dialog with a user.

New is that a feedback request that offers too limited functionality in the eyes of the user can disturb as well. This insight is interesting because related work has focused on the aspect of timing feedback requests. According to our data, it is also important that the feedback request gives the user the ability to provide feedback in a way that is intuitive and desired by the user. Our chosen combination of a Quality of Experience rating and a text field for user feedback was perceived to be too limited by some users. Additional capabilities may be needed, such as screenshots, voice, video recordings, or photographs (Seyff et al. 2011).

It is interesting to compare these results with the Qualinet definition of QoE (Le Callet et al. 2012) that we apply here for a feedback tool. According to that definition “Quality of Experience (QoE) is the degree of *delight or annoyance* of a person whose experiencing involves an *application, service, or system*. It results from the person's evaluation of the fulfilment of his or her *expectations and needs* with respect to the *utility and/or enjoyment* in the light of the person's *context, personality, and current state*.” A feedback tool *annoys* users if the parameters are not configured well. Users may feel *delighted* while giving feedback if the feedback has strong *utility*, such as the anticipated improvement of the product in a future release. The study has shown that the *expectations and needs* of the feedback tool are about the timing and content parameters that should be respected when issuing a feedback form. The user's *context, personality, and current state* are reflected in the user's expertise of using the product. We could not identify any other factors in the presented study, including cultural background, which would affect the QoE of the feedback tool.

A feedback request that is disturbing causes negative emotions such as anger (Solomon 2008; Scherer 2005). Such emotions are visible in bad QoE ratings (Antons et al. 2014). The disturbances may also hinder sustained adoption of a product. A user may resist incorporating a product into his daily routines where usefulness and long-term usability are important (Karapanos 2013). Even though the software product may evoke positive emotions in a user, the negative emotions caused

by the disturbance may prevent or delay development of emotional attachment to the product. Hence, in addition to offering an attractive product, it is important to present feedback requests satisfactorily or to offer the possibility to disable the feedback tool.

While feedback may disturb a product's users, our study showed that this disturbance has a negligible impact on the users' reported Quality of Experience for the software product. The users differentiated between a feedback tool they were providing feedback with and the software they were providing feedback for. The disturbance of a user was hardly reflected in that user's QoE ratings for the product. As we could not find any prior study that investigated this perceived separation between product and feedback tool, we believe that this is an interesting new result. The negligible impact implies that software product vendors may trust the collected feedback even if the feedback requests disturb the users to some extent.

In contrast to the perceived separation of a feedback tool and a software product, users blurred the boundary between the software product and the device on which the product was running. The user feedback mixed product and device factors. Perhaps the users could not distinguish the device and the product, or they considered the device to be a part of the product. Thus, a software vendor can receive informative feedback not only about the software product but also about the devices the customers are using to run the product on.

Although disturbing feedback requests did not show any significant impact on QoE of the studied software product, the disturbances might affect how well feedback requests are answered. Disturbances may demotivate users to provide rich feedback since the users would ignore disturbing feedback requests. This reaction was evident in that many study participants canceled feedback requests or switched the feedback tool off. The design of a feedback mechanism is possible through configuring the parameters of the feedback requests model.

The above findings were achieved in a case study that was set up the environment close to reality with less pressure and control on the participants. A pressurized and controlled environment, on one side, could increase the sensitivity of users in response to the environment that might impact on users' perception. On the other hand, such controlled situation could not affect the ability of users to evaluate the software or feedback. Putting users on a regime such as time pressure could amplify the anxiety leading to different judgment (Maule et al. 2000).

Like any other study, also the presented study has its limitations. For example, we did not research when users decide to decline feedback requests (e.g., canceling feedback forms). The research could be interesting to investigate the consequence of being disturbed by the feedback requests in a future study. However, this limitation did not affect the presented result in Fig. 3 that was achieved based on the post-questionnaire. Furthermore, approaches need to be evaluated for including the identified parameters of the user task, feedback request timing, expertise-phase, feedback request frequency, and feedback request content in the design of a feedback mechanism. Finally, users may have different thresholds for feeling affected by disturbance; depending on the situation, some are rapidly disturbed, while others can accept a lot of annoyance (Van der Ham et al. 2014). Therefore, categories of users, contexts, and products may need to be identified to allow investigation of feedback request parameters in each cluster separately. Such research will be future work.

6 Summary and conclusion

Quality of Experience (QoE) is a measurement that is widely used to assess users' perceptions when experiencing a software product. With knowledge about QoE, companies hope to make appropriate decisions to win and retain customers by evolving their products in meaningful ways. Collecting users' QoEs requires automatic and frequent requests for feedback. However, automated requests for feedback may disturb users and perhaps degrade their QoE ratings.

The current study investigated the candidate relationship between the characteristics of automatic feedback requests and the QoE of a software product. The study followed a mixed qualitative-quantitative research method with 35 software engineering participants. We integrated a feedback tool into a mobile software product to prompt participants for feedback randomly in the middle of their experiences. At the end of the users' experiences, we collected their perceptions about the feedback requests and their experiences of using the application through a post-questionnaire.

We offer two contributions to the researcher and practitioner communities. First, we propose a feedback request model that parameterizes the characteristics of feedback requests. The parameters outline the task, timing of the task for issuing the feedback requests, user's expertise-phase with the product, the frequency of feedback requests about the task, and the content of the feedback request. The findings may inform researchers of the parameters that disrupt users' experiences, which may help them develop suitable feedback mechanisms to control users' disturbance. The findings may also help practitioners design the feedback tool and the corresponding feedback mechanisms by adjusting the parameters.

Second, the study showed that feedback requests have negligible impacts on users' QoEs of a software product. Specifically, the quality of the software product has a greater impact on the QoE than the characteristics of the feedback request. For practitioners, this finding implies an ability to trust feedback collected from users, even when the requests for feedback are considered disturbing. The results also imply that the quality of a software product is the most important aspect for practitioners to focus on when examining user feedback. However, the design of suitable feedback mechanisms should not be neglected, since feedback mechanisms are useful for collecting informative user feedback about both software products and any disturbances caused by feedback requests. The informative user feedback assists in enhancing software engineering activities. An informative user feedback assists requirement engineers to elicit new requirements and revise the current requirements for next releases of the software product (Carreño and Winbladh 2013). Such rich feedback also contains valuable information for developers to redevelop a functionality and validate the software product idea (Kujala 2008) toward the software evolution (Pagano and Brügge 2013).

The result was achieved based on constructing one situation. However, case variations in practice might stimulate users' emotions differently and lead to new achievements. Therefore, it would be interesting to replicate the study considering several varieties of contextual and system factors in future. The materials for replication are available in <http://bit.ly/2o89rO4>.

Appendix: Post-questionnaire

Table 2 shows the questions that were used to collect user feedback after the participants performed modeling of the requirements.

Table 3 Post-questionnaire

=== About Yourself ===
Q1: Your Student ID:
Q2: Did you use Flexisketch already before this assignment? [Yes/No]
Q3: Do you have experience with applications like the Drug Supply Manager (DSM)? [Yes (describe your experience)/No]
=== How You Did the Assignment ===
Q4: Did you install the QoE Probe before doing the modeling with Flexisketch? [Yes/No]
Q5: What device did you use for modeling with Flexisketch (e.g., I would have done it on “Sony Z2 Tablet” – you might have used another one)?
Q6: When in the workshop video did you start the modeling (e.g., “12 minutes after start”)?
Q7: When in real-world time did you start the modeling (e.g., “December 14 at 09:04”)?
Q8: How much time did you spend for the modeling?
=== Your Experience of modeling with Flexisketch ===
<u>Note again: no grades are given for this assignment – please be honest</u>
Q9: How good was Flexisketch as a tool for modeling requirements? [Opinion Score Scale, rationale]
Q10: Being a potential requirements engineer, would you use Flexisketch again for requirements modeling? [Yes/No, rationale]
=== QoE Probe ===
Q11: Approximately, how many times did you see any QoE Probe feedback form while you were using Flexisketch?
Q12: How good was the QoE Probe in trying in minimizing the disturbance of your modeling work? [Opinion Score Scale, rationale]
Q13: Any other comment?

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Abelow, D. (1993). Automating feedback on software product use. *CASE Trends December*, 15–17.
- Adamczyk, P. D., & Bailey, B. P. If not now, when?: the effects of interruption at different moments within task execution. In *SIGCHI conference on Human factors in computing systems, Vienna, Austria, 2004*: ACM.
- Ahtinen, A., Mattila, E., Vaatanen, A., Hynninen, L., Salminen, J., Koskinen, E., et al. User experiences of mobile wellness applications in health promotion: user study of Wellness Diary, Mobile Coach and SelfRelax. In *3rd International Conference on Pervasive Computing Technologies for Healthcare, London, UK, 2009* (pp. 1–8): IEEE.
- Ames, M., & Naaman, M. Why we tag: motivations for annotation in mobile and online media. In *SIGCHI conference on Human factors in computing systems, San Jose, California, USA, 2007* (pp. 971–980): ACM.
- Antons, J.-N., Arndt, S., Schleicher, R., & Möller, S. (2014). Brain activity correlates of quality of experience. In *Quality of experience* (pp. 109–119): Springer.

- Bailey, B. P., Konstan, J. A., & Carlis, J. V. The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In *IFIP International Conference on Human Computer Interaction (INTERACT)*, Tokyo, Japan, 2001 (pp. 593–601).
- Barrett, L. F., Mesquita, B., & Gendron, M. (2011). Context in emotion perception. *Current Directions in Psychological Science*, 20(5), 286–290.
- Beyer, J., & Möller, S. (2014). Gaming. In *Quality of experience* (pp. 367–381): Springer.
- Broekens, J., Pommeranz, A., Wiggers, P., & Jonker, C. M. Factors influencing user motivation for giving online preference feedback. In *5th Multidisciplinary Workshop on Advances in Preference Handling (MPREF'10)*, Lisbon, Portugal, 2010: Citeseer.
- Canale, S., Facchini, F., Gambuti, R., Palagi, L., & Suraci, V. User profile based quality of experience. In *18th International Conference on Computers (part of CSCC '14)*, Santorini Island, Greece, 2014 (Recent Advances in Computer Engineering).
- Carreño, L. V. G., & Winblad, K. Analysis of user comments: an approach for software requirements evolution. In *35th International Conference on Software Engineering (ICSE 2013)* San Francisco, USA, 2013 (pp. 582–591): IEEE.
- Carver, J., Jaccheri, L., Morasca, S., & Shull, F. (2003). *Issues in using students in empirical studies in software engineering education*. Paper presented at the Ninth International Software Metrics Symposium.
- Côté, N., & Berger, J. (2014). Speech communication. In *Quality of experience* (pp. 165–177): Springer.
- Elo, S., & Kyngäs, H. (2008). The qualitative content analysis process. *Journal of Advanced Nursing*, 62(1), 107–115.
- Feiten, B., Garcia, M.-N., Svensson, P., & Raake, A. (2014). Audio transmission. In *Quality of experience* (pp. 229–245): Springer.
- Fernández-Dols, J.-M., & Russell, J. A. (2003). Emotions, affects, and mood in social judgements. In T. Millon, M. J. Lerner, & I. B. Weiner (Eds.), *Handbook of Psychology, Personality and Social Psychology* (2 ed., Vol. 5, pp. 283–297). Hoboken New Jersey: John Wiley & Sons.
- Fiedler, M., Hossfeld, T., & Tran-Gia, P. (2010). A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, 24(2), 36–41.
- Fotrousi, F. (2015). QoE-Probe Android. <https://github.com/famazfotrousi/QoE-Probe-Android>. Accessed 22 May 2015.
- Fotrousi, F., Fricker, S. A., & Fiedler, M. (2014). *Quality requirements elicitation based on inquiry of quality-impact relationships*. Paper presented at the 22nd IEEE International Conference on Requirements Engineering, Karlskrona, Sweden.
- Fricker, S. A., Schneider, K., Fotrousi, F., & Thuemmler, C. (2015). Workshop videos for requirements communication. *Requirements engineering*, 1–32, doi:10.1007/s00766-015-0231-5.
- Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., & Landay, J. A. (2007). *MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones*. Paper presented at the 5th international conference on Mobile systems, applications and services (MobiSys2007), San Juan, Puerto Rico.
- García, M.-N., Argyropoulos, S., Staelens, N., Naccari, M., Rios-Quintero, M., & Raake, A. (2014). Video streaming. In *Quality of experience* (pp. 277–297): Springer.
- Garland, R. (1991). The mid-point on a rating scale: is it desirable. *Marketing Bulletin*, 2(1), 66–70.
- Golafshani, N. (2003). Understanding reliability and validity in qualitative research. *The Qualitative Report*, 8, 597–606.
- Golaszewski, S. (2013). Flexisketch. <https://play.google.com/store/apps/details?id=ch.uzh.ifi.reg.flexisketch&hl=en>.
- Hak, T., & Dul, J. (2009). Pattern matching. In A. J. Mills, G. Durepos, & E. Wiebe (Eds.), *Encyclopedia of case study research* (Vol. 2, pp. 663–665). Thousands of Oaks, CA: Sage Publications.
- Herzog, A. R., & Bachman, J. G. (1981). Effects of questionnaire length on response quality. *Public Opinion Quarterly*, 45(4), 549–559.
- Höst, M., Regnell, B., & Wohlin, C. (2000). Using students as subjects—a comparative study of students and professionals in lead-time impact assessment. *Empirical Software Engineering*, 5(3), 201–214.
- Hsieh, H.-F., & Shannon, S. E. (2005). Three approaches to qualitative content analysis. *Qualitative Health Research*, 15(9), 1277–1288.
- Ickin, S., Wac, K., Fiedler, M., Janowski, L., Hong, J.-H., & Dey, A. K. (2012). Factors influencing quality of experience of commonly used mobile applications. *Communications Magazine, IEEE*, 50(4), 48–56.
- ITU-T (2003). ITU-T P.800. in *Mean Opinion Score(MOS) terminology*, ed: Telecommunication Standardization Sector of ITU.
- Ivory, M. Y., & Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33(4), 470–516.
- Jordan, P. W. (1998). Human factors for pleasure in product use. *Applied Ergonomics*, 29(1), 25–33.
- Karapanos, E. (2013). User experience over time. In *Modeling users' experiences with interactive systems* (pp. 57–83): Springer.
- Kennedy, M. M. (1979). Generalizing from single case studies. *Evaluation quarterly*, 3(4), 661–678.
- Khirman, S., & Henriksen, P. Relationship between quality-of-service and quality-of-experience for public internet service. In *3rd Workshop on Passive and Active Measurement, Fort Collins, Colorado, USA*, 2002.

- Kim, J. H., Gunn, D. V., Schuh, E., Phillips, B., Pagulayan, R. J., & Wixon, D. Tracking real-time user experience (TRUE): a comprehensive instrumentation solution for complex systems. In *SIGCHI Conference on Human Factors in Computing Systems, Florence, Italy*, 2008: ACM.
- Kujala, S. (2008). Effective user involvement in product development by improving the analysis of user needs. *Behaviour & Information Technology*, 27(6), 457–473.
- Kujala, S. (2003). User involvement: a review of the benefits and challenges. *Behaviour & Information Technology*, 22(1), 1–16.
- Kujala, S., & Miron-Shatz, T. Emotions, experiences and usability in real-life mobile phone use. In *SIGCHI Conference on Human Factors in Computing Systems, Paris, France*, 2013 (pp. 1061–1070): ACM.
- Le Callet, P., Möller, S., & Perkis, A. (2012). Qualinet white paper on definitions of quality of experience. *European Network on Quality of Experience in Multimedia Systems and Services*.
- Maule, A. J., Hockey, G. R. J., & Bdzola, L. (2000). Effects of time-pressure on decision-making under uncertainty: changes in affective state and information processing strategy. *Acta Psychologica*, 104(3), 283–301.
- Mitra, K., Zaslavsky, A., & Åhlund, C. A probabilistic context-aware approach for quality of experience measurement in pervasive systems. In *26th ACM symposium on applied computing, Taichung, Taiwan* 2011 (pp. 419–424): ACM.
- Pagano, D., & Brügge, B. User involvement in software evolution practice: a case study. In *35th International Conference on Software Engineering (ICSE 2013), San Francisco, CA, USA*, 2013 (pp. 953–962): IEEE Press.
- Potter, W. J., & Levine-Donnerstein, D. (1999). Rethinking validity and reliability in content analysis. *Journal of Applied Communication Research*, 27(3), 258–284.
- Raake, A., & Egger, S. (2014). Quality and quality of experience. In *Quality of experience* (pp. 11–33): Springer.
- Reiter, U., Brunnström, K., De Moor, K., Larabi, M.-C., Pereira, M., Pinheiro, A., et al. (2014). Factors influencing quality of experience. In *Quality of experience* (pp. 55–72): Springer.
- Roto, V., Law, E., Vermeeren, A., & Hoonhout, J. (2011). User experience white paper - bringing clarity to the concept of user experience. Results from Dagstuhl Seminar on Demarcating User Experience. Dagstuhl Seminar Proceedings. Schloss Dagstuhl, Leibniz-Zentrum für Informatik, Germany.
- Scherer, K. R. (2005). What are emotions? And how can they be measured? *Social Science Information*, 44(4), 695–729.
- Schleicher, R., Westermann, T., & Reichmuth, R. (2014). Mobile human–computer interaction. In *Quality of experience* (pp. 339–349): Springer.
- Seyff, N., Ollmann, G., & Bortenschlager, M. iRequire: gathering end-user requirements for new apps. In *19th IEEE International Requirements Engineering Conference (RE'11), Trento, Italy*, 2011: IEEE.
- Sjöberg, D. I., Anda, B., Arisholm, E., Dybå, T., Jørgensen, M., Karahasanović, A., et al. (2003). Challenges and recommendations when increasing the realism of controlled software engineering experiments. In R. Conradi, & A. I. Wang (Eds.), *Empirical methods and studies in software engineering* (pp. 24–38). Berlin Heidelberg: Springer.
- Solomon, R. C. (2008). The philosophy of emotions. In M. Lewis, Haviland-Jones, J. M. & Barrett, L. F. (Eds.), *Handbook of emotions* (3rd ed., pp. 3–16). New York: Guilford Press.
- Strohmeier, D., Egger, S., Raake, A., Hoffeld, T., & Schatz, R. (2014). Web browsing. In *Quality of experience* (pp. 329–338): Springer.
- Szajna, B., & Scamell, R. W. (1993). The effects of information system user expectations on their performance and perceptions. *MIS Quarterly*, 17(4), 493–516. doi:10.2307/249589.
- Tohidi, M., Buxton, W., Baecker, R., & Sellen, A. User sketches: a quick, inexpensive, and effective way to elicit more reflective user feedback. In *4th Nordic Conference on Human-Computer Interaction: Changing Roles, Oslo, Norway*, 2006 (pp. 105–114): ACM.
- Van der Ham, W. F., Broekens, J., & Roelofsma, P. H. (2014). The effect of dominance manipulation on the perception and believability of an emotional expression. In T. Bosse, Broekens, J., Dias J., & Zwaan, J. v. d (Ed.), *Emotion modeling: towards pragmatic computational models of affective processes* (pp. 101–114, *Lecture Notes in Artificial Intelligence*): Springer.
- Varela, M., Skorin-Kapov, L., & Ebrahimi, T. (2014). Quality of service versus quality of experience. In *Quality of experience* (pp. 85–96): Springer.
- Wüest, D., Seyff, N., & Glinz, M. Flexisketch: a mobile sketching tool for software modeling. In *International Conference on Mobile Computing, Applications, and Services*, 2012 (pp. 225–244): Springer.
- Wüest, D., Seyff, N., & Glinz, M. Sketching and notation creation with FlexiSketch Team: evaluating a new means for collaborative requirements elicitation. In *23rd IEEE International Requirements Engineering Conference (RE'15), Ottawa, Canada*, 2015 (pp. 186–195): IEEE.
- Yin, R. K. (2014). *Case study research: Design and methods* (5ed.). Thousands of Oaks, CA: Sage publications.
- Zhang, J., & Ansari, N. (2011). On assuring end-to-end QoE in next generation networks: challenges and a possible solution. *IEEE Communications Magazine*, 49(7), 185–191.
- Zijlstra, F. R., Roe, R. A., Leonora, A. B., & Krediet, I. (1999). Temporal factors in mental work: effects of interrupted activities. *Journal of Occupational and Organizational Psychology*, 72(2), 163–185.



Farnaz Fotrousi is a Ph.D. candidate in Software Engineering at Blekinge Institute of Technology in Sweden and FHNW in Switzerland. Farnaz is interested in applied research, combining her over 15 years of industrial experience in software companies with her research knowledge. Her main areas of research interests are in product management and evolution, user feedback for modeling of the product success, and requirement engineering. Farnaz's research is currently on leveraging users' Quality of Experience (QoE) for validation and maturation of software systems.



Samuel Fricker is a professor at FHNW, Switzerland, and also affiliated with Blekinge Institute of Technology, Sweden. He is interested in small products that help to solve big challenges. With more than 20 years of industry and research experience, he has worked with companies at any scale, from Startups to Fortune500, and with government agencies such as the European Commission.



Markus Fiedler is a professor of teletraffic systems at Blekinge Institute of Technology (BTH), Sweden. His research interests are focused on enabling and disabling factors for Quality of Experience (QoE) in networks and applications; on the validation of future internet solutions in terms of QoE; on performance measurements, modeling, analysis, and control in virtual and wireless environments; and on ecologic and economic sustainability factors in the context of networking. He has been active in European projects (COST, FP6, FP7). In particular, he held several leading positions in a series of Networks of Excellence on future networks (EuroNGI, EuroFGI, and Euro-NF). He has also been co-organizer of a set of Quality of Experience-related Dagstuhl seminars.