Scenario-Driven Continuous Mobility Requirements Analysis in Mobile App Maintenance

Xiaozhou Li[1], Zheying Zhang[2] and Timo Poranen[3]

[1] University of Tampere, Kalevantie 4, 33014, Tampere, Finland
{xiaozhou.li; zheying.zhang; timo.t.poranen}@uta.fi

Abstract. As mobile devices have become an indispensable part of people's lives, the growth of mobile app market is inevitably rapid. Accordingly, the mobile app users have increasing demands on the quality of the apps. Their tolerance towards bugs and poor usability of the apps is growing low. In addition, as the mobile devices and apps have enabled users to use them in various contexts, the mobility of the apps, referred to as their capability to provide ubiquitous services with quality, is also required. In this study, we propose the scenario-driven approach of mobility requirements analysis with context and ways of interaction. It supports the continuous requirements analysis in the maintenance of mobile apps where their mobility is constantly maintained despite the changes in the features.

Keywords: Mobile App, Mobility, Requirements, Scenario, Situational Contexts, Interaction, Maintenance.

1 Introduction

The mobility of mobile apps, as their most significant feature, is defined as the ability to access services ubiquitously, on the move, though wireless networks and various mobile devices [1]. It enables the users to access the services with the independence from time and space [2]. The mobility of mobile apps is closely related to their context of use, especially the situational context, which largely influences the post-adoption behaviors of the users, including use continuance, word-of-mouth, and complaints [3]. Therefore, the quality of the mobile apps that enables the user to use them comfortably and respond with positive post-adoption behaviors in various contexts determines the apps' overall success [4].

The different digital distribution platforms for mobile apps enable and promote the easy communication among users and developers. Therefore, it is a common strategy for the companies to update their apps in short cycles continuously [5]. Despite the proposed continuous requirements engineering methods for mobile app development [6], it does not cover the mobile app maintenance and the continuous requirements engineering on mobile app mobility. Abrupt changes in a feature or design might affect the mobility attribute of an app. When change requests are proposed, the impact analysis shall consider the various contexts and users' way of interaction with the affected features. Therefore, in addition to the common practices on impact analysis, the app maintainers must pay attention to: 1) the identification of contexts where users most often interact with the affected feature and 2) the analysis of ways by which users interact with this feature in identified contexts.

Context is defined as "any information that characterizes a situation related to the interaction between humans, applications, and the surrounding environment" [7]. In addition, many studies categorize context [8-10], among which Belk [8] emphasizes the situational characteristics of context that have an impact on the customer behaviors, and categorizes them into physical context, social context, temporal context, task definition, and antecedent states. Such impact on the customer behavior of mobile services is also studied [11,12]. On the other hand, the different ways of interaction between users and apps are also identified [4].

Hence, to analyze the mobility attribute of an app, we shall obtain information on the context in which the app is often used. Scenario is an efficient tool to facilitate context information acquisition and its mapping with the system functions from the users' viewpoint [13] it has been used widely in requirements engineering related activities [14-17]. In this paper, we tackle the issue of how to maintain the mobility of mobile apps when a new feature is requested, or a change request is proposed. We focus on the integration of the use of scenarios into the identification of situational contexts and the analysis of ways of interaction and present a process model for mobility requirements analysis to maintain the app mobility.

The remainder of the paper is organized as follows. Section 2 overviews understanding of mobility as the quality of mobile apps, and the situational contexts of using an app. Section 3 presents the use of scenarios to elicit contexts information for mobility requirements analysis. Section 4 explains the use of conflicts in ways of interaction representing mobility loss and the process model of analyzing a change request towards mobility maintenance. Section 5 demonstrates how our approach is applied for requirements analysis in a real-life mobile app case. Section 6 concludes the paper with the limitation and future work covered.

2 Mobility and Situational Context

Mobility is the ability to access services ubiquitously, on the move, though wireless networks and various mobile devices [1]. Closely related to ubiquitous computing [18] and nomadic computing [2,19], the concept of mobility is seen as the capability of mobile apps providing services with the independence from time and space or the accessibility whenever and wherever [2,19]. From the perspective of app development, mobility is a unique quality attribute of mobile apps that enables not only operations on functionalities, but also the comfortable interaction throughout various contexts [4]. Hence, identifying what the contexts are and how users use the app in those contexts is the prerequisite to maintain and enhance the mobility of mobile apps.

The context of information systems and mobile devices have been studied widely. Most of the given definitions of context claim that context encompasses the aspects including, location, the physical environment, people nearby the user, the time of the day, the computing environment, user's emotional states, the focus of attention, and so on [8,9,20,21]. Amongst the wide spectrum of mobile context in general, the situational characteristics of context, referred to as the situational context, has been found closely influential to the users' continuous use and behaviors after use [3,4,8].

The temporality, spatiality, sociality are the three critical perspectives to illustrate the situational context [4,8,23]. The temporal perspective illustrates originally the time of day or season of the year when the users' interaction occurs [8,38]. To investigate the situational influence of temporality towards the users' behavior, we see the temporal perspective as the users' sense of time when starting the use of the app as well as their sense of time for the interaction of the app itself which influences the contextual activities [4]. Therefore, to provide a straightforward understanding of the temporal context influence on users' behavior, we categorize temporality into intensive and allocative. The second important perspective of situational context is the spatiality. It indicates the influence of physical surroundings towards users' use of the apps [8,38]. Concerning only the situational variables, the environmental influence of the light, wind, air, and etc. are excluded. We see the spatial context as the one influences the users' physical movement and gestures, and the indicator of the users' physical availability relevant to the use of apps [4]. We specify the spatial context by adopting the value categorization based on the concept of mobile modality [24], including, visiting, traveling and wandering. The sociality is the perspective describing the interpersonal interaction between the user and the others [8,21,25,26]. We interpret the social context based as the social norms that constrain users from or encourage users into the interaction with the app [4,26]. Thus, it contains the value of either constraining or encouraging. The description and example of each situational context perspective are shown in Table 1.

Table 1. Context Perspective Specification

Temporal	Intensive (I)	in urgent need of achieving other goals with a limited spare time of interacting with the app (*e.g., catching a train which leaves in 5 min*)
	Allocative (A)	no urgent goal to achieve and is temporally available (*e.g., idle at home*)
Spatial	Visiting (V)	in a physically and geographically stationary status (e.g., sitting in class)
	Traveling (T)	in a passively moving status (e.g., *sitting in a car*)
	Wandering (W)	in an actively moving status (e.g., *walking in a building*)
Social	Constraining (C)	the social norms require the user not to interact with the app (e.g., *speaking to the president*)
	Encouraging (E)	the social norms allow the user to interact with the app (e.g., *drinking coffee alone*)

Based on our categorization of the situational context of the mobile app, for each user, we consider he/she must be situated in one combination of temporary, spatial and social context. Thus, by cross-combining the different perspective values of the situational contexts, we have then 12 unique situational contexts including IVC, AVC, IVE, AVE, ITC, ATC, ITE, ATE, IWC, AWC, IWE, AWE [4]. A description of a situational context example is referred to as a scene. It can be used in the mobility analysis to ease the understanding of each situational context. For example, a typical situation of "*a person is sitting alone in the classroom doing nothing*", falls into the situational context AVE (i.e., Allocative, Visiting, Encouraging). It indicates that the user is in a physically stationary status (i.e., Visiting) and temporally available (i.e., Allocative) and is able to interact with the app (i.e., Encouraging). Herein, such a situation is one scene of the AVE context.

Furthermore, the situational contexts are not equally important to take into account for each app. Based on the vision and scope of each app, it is originally planned and developed to fit specific situational contexts. For example, the game *Vainglory*[1] (a real-time player-vs-player multiplayer online battle arena) is certainly designed for the AVE context, as one gameplay session of it will take quite long time with users' full concentration. It is unsafe to play the game when walking or running and is also inappropriate to play in a social event too. However, the game *2048*[2] (a casual game without time limit) for not only AVE context but also other contexts, such as ITE, and AWE, as the user can make a quick move even when the time is limited. We define such more important context(s) as the primary context(s) of the app. The suitable context for individual features can be inherited from the primary contexts, however, it can also differ from feature to feature. Because the different features serve the different courses of interaction, which occur more preferably, in a different context. Taking *Pokémon Go*[3]as an example, the feature of "*searching for a Pokémon nearby*" requires a player to walk around the streets and encourages users to play in an AWE context, while "*catching a Pokémon*" requires the player to stop walking for a short while and mostly fits an AVE context. Therefore, the prioritization of situational contexts is also necessary for each individual feature.

3 Scenarios in Requirements Analysis

Scenarios are stories or examples of events as grounded narrative taken from real-world experience, in which the design rationale or a problem situation is anchored for requirements analysis and design decisions [13,27,28]. Also, they can be sequences of behavior and possible contextual description used for guiding software design and providing the basis of test scripts [27,28]. According to the roles they take in software system development, their representation varies from rich and informal narrative description of real-world experience on using a system to formatted texts and more formal models of an event sequence or system behavior. Although the content and the formality of representation of scenarios may vary, they have an essential element in common, which is the event sequences.

1 https://5v5.vaingloryga me.com/
2 https://gabrielecirulli.github.io/2048/
3 https://www.pokemongo.com/

An event sequence is a description of what people do and how people use a software system in a particular instance. It is a sequence of activities or a more or less richly branched structure of such sequences [29]. In addition to the event sequence, a scenario may also include the description of people who are engaged in the event, the goals the people hope to achieve, and the condition or situational context where the event occurs [28,30]. These are useful elements to support the analysis of the use of a target system in requirements elicitation and analysis, to gather stories, search for generalities, identify and analyze the needed behavior of software [14-16, 28]. For example, the sentence *"… After my morning class, I sit at my table and feel bored. I then opened my WeChat to send a message to Vicky to have a quick chat."* is extracted from a large textual scenario paragraph for requirements analysis for an app *WeChat*[4]. It contains the mentioned elements, which describes a "*sending a message*" feature expected by a user for achieving his/her goal to "*have a quick chat*". In addition, the user is in a particular context, i.e., "*after my morning class*" and "*sits at my table and feel bored*", when interacting with the application. To apply the definition of context given in the prior section, we get the context as AVE (allocative, visiting, encouraging).

The intrinsic mobility of mobile devices and the apps within have enabled users to use them ubiquitously [2,4,31], which largely changed the ways of people perceiving the use of mobile apps [32]. The use of mobile apps in various contexts shall be analyzed when an app's feature is designed for implementation. Scenarios form a means of situating discussion about the design so that the mobility attribute can be elicited for fulfillment by reasoning about an event sequence posed by scenarios describing a context of use [27,28].

Many studies propose methods and processes to construct scenarios for requirements analysis in various forms [15,16,27], as well as the formal methods translating scenarios into a set of executable state machines for simulation and validation [33]. Other studies cover scenario-based requirements analysis methods for mobile app development as well [34]. In this study, we focus on using scenarios as the driver to facilitate the situational context extraction. Different from the scenario-based requirements engineering approach, in app maintenance, previous requirements documents can be used as reference to identify the concerned feature.

4 Continuous Mobility Requirements Analysis

To maintain the mobility attribute of the mobile apps in the maintenance and evolution phase, the mobility loss resulting from changes in features must be detected and controlled. It shall prevent the changed features cause the users' uncomfortable interactions in some important situational contexts. Such practice shall continuously detect and address the potential mobility loss within the proposed change requests.

[4] https://web.wechat.com/

4.1 Context Extraction from Scenarios

The first critical step of proceeding the approach is to obtain the context information concerning the proposed change requests. Here we use scenarios to elicit the required context information.

When a change request is proposed in the maintenance phase, referencing the documented requirements, the features affected by the change request could be identified. Presumably, when similar mobility requirements analysis has been executed in the development phase, the previous context analysis results can be inherited.

A change request records an adjustment for an app which includes a narrative abstract of the change. If the change is about specific features, the scene could be extracted from the abstract. In addition, scenes and scenarios can be identified and specified by interviewing the originator who submitted the change request, as well as the stakeholders who are related to the affected features. Therein, the requirements analyst shall pay attention to the edge of the complete use session of the feature and the scene switching. For ambiguities, the stakeholder shall be further probed. On the other hand, the event sequence of scenarios can be recorded in various forms, including textual descriptions, use cases, activity sequence diagrams, etc. [27,28].

With the created scenarios, the requirements analyst shall be able to obtain a list of scenes where the change occurs. For each scene, the situational context (one out of the 12) will be identified. When there are multiple scenes describing the same situational context, the number will be recorded to facilitate the prioritization of the situational contexts. When the list of situational contexts is extracted from the scenarios, they shall be distinguished from the primary contexts for the app and assigned uniquely to each feature.

4.2 Detection of Ways of Interaction Conflicts

The way of interaction describes the users' ways of switching from their activity in the context to the interaction with the apps until the goal of app use is accomplished [4]. Ideally, the user shall be able to start the interaction with the app whenever possible, while presumably, the context has zero influence on the app use. However, when the context has influence, the switching from context interaction to the app interaction becomes a trade-off. Subjectively, the users can choose not to use any feature of any app, especially when the context activity cannot be interrupted or even slightly influenced, temporally, spatially, or socially. For example, when the user is taking an exam at school, although he/she is able to use apps physically or spatially, the time limits (temporally intensive) and social norms (socially constraining) will influence his decision on the app use. Therefore, the matching between the ideal way of interaction from the context and the expected way of interaction with the app feature determines how the users feel about the mobility of the app. Hence, the mismatching between such two ways of interaction results in mobility loss.

There are four different ways of interaction, including accompanying, interrupting, intermittent, and ignoring (shown in Table 2) [4].

Table 2. Specification on Ways of Interaction

Ways of Interaction	Description
Accompanying	The paralleling engagement in both user's interaction in context and the interaction with app. (e.g., *Listen to music when reading*)
Interrupting	The user converts full concentration on the interaction with app and ceases the original activity. (e.g., *Stop reading and start browsing friends' status updates*)
Intermittent	The interlaced engagement in both the user's interaction in context and the interaction with app, with the whole process of several short interactions, which are neither consistent nor interfering the proceeding of the original task. (e.g., *Watch TV and send messages*)
Ignoring	The interaction with app is ignored by the user in order to maintain the proceeding of interaction in context. (e.g., *Ignore all messages when taking exams*)

According to the definition of the ways of interaction, it represents the users' interaction switching from the context to the app. Therefore, for every context as categorized in the previous section, one or more ways of interaction is encouraged, and defined as the ideal way of interaction (IWoI), as shown in Table 3. To identify the IWoIs for a typical situational context, the three perspectives of the situational context, temporality, spatiality, and sociality must be taken into account. For example, Table 3 shows an example of determining the possible ideal ways of interaction for each context perspective.

Table 3. Ideal Ways of Interaction for Context Perspectives

Temporality	Spatiality	Sociality
Intensive: *Accompanying, Intermittent* **Allocative**: *Accompanying, Intermittent, Interrupting*	**Visiting**: *Accompanying, Intermittent, Interrupting* **Traveling**: *Accompanying, Intermittent, Interrupting* **Wandering**: *Accompanying, Intermittent*	**Constraining**: *Intermittent* **Encouraging**: *Accompanying, Intermittent, Interrupting*

Based on the table of IWoIs for context perspectives, the IWoI for each situational context is thus the intersection of ways of interaction set of all three perspectives. For example, the IWoI of IVC situational context (Intensive, Visiting, Constraining) is Intermittent, which is the intersection of (Accompanying, Intermittent), (Accompanying, Intermittent, Interrupting), and (Intermittent).

Similarly, for a feature offered by an app, some ways of interaction are also required, defined as the expected way of interaction (EWoI). The EWoI of the feature is determined by the persistence and obtrusiveness of the feature [36,37]. The persistence of a feature indicates the duration for a user to complete a use session with it.

Therein, a persistent feature requires long interaction when an ephemeral feature requires a short duration. The obtrusiveness of a feature indicates whether the feature urges the user to start using immediately.

Table 4. Expected Ways of Interaction for Features

	Obtrusive	Unobtrusive
Persistent	*Interrupting, Accompanying*	*Accompanying, Interrupting*
Ephemeral	*Intermittent, Accompanying*	*Intermittent*

Table 4 shows one example of determining the EWoI of one app feature. For example, the “sending instant message” feature of WeChat is ephemeral (i.e., it takes several seconds to send a message) and obtrusive (i.e., the user will be notified when receiving a message). Therefore, the possible EWoI of this feature is intermittent or accompanying.

Therefore, the conflicts between such ways of interaction choices and the ones for the contexts will then cause the users’ uncomfortable use of the app or frequent ignoring the app when such contexts occur often. Thus, detecting and addressing such conflicts between IWoI and EWoI is the key to improve users’ comfortable use of the app in primary contexts, and further the sense of mobility overall.

4.3 The Mobility Requirements Analysis Process Model

The mobility requirements analysis process model is shown in Fig. 1.

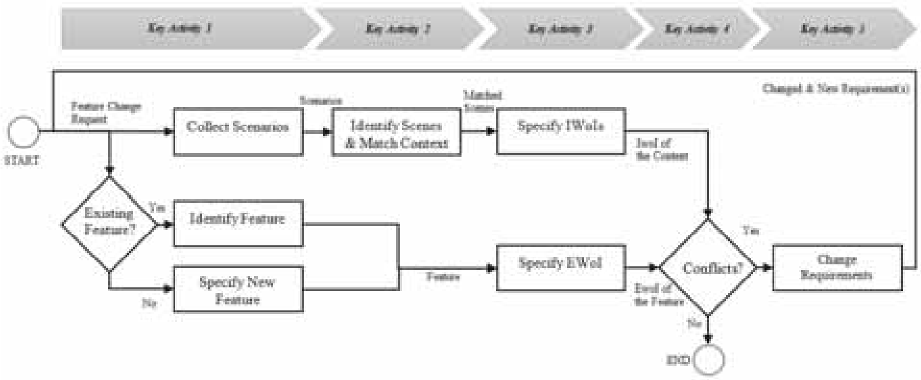

Fig. 1. Mobility Requirements Analysis Process Model

Key Activity 1. Collect Scenarios & Identify Feature

Before the maintenance phase starts, the development team shall possess the project vision and scope, as well as the requirements documents in a certain form. Therefore, when a feature change request is proposed (e.g., based on the reviews of the end users) the team shall start collecting the scenarios from the stakeholders who propose the change based on the team's experiences and overall understanding of the app. On the other hand, the affected features will be identified. The feature is associated with information such as the context and the EWoI which has been accumulatively specified during the development and maintenance process.

Key Activity 2. Identify Scenes & Match Context

Based on the scenarios collected, the team shall then be able to specify the situational context of the feature with the reference of the situational context classification. If the primary contexts of the app are available in the requirements document, the context information shall be compared with the ones obtained from the scenarios. When the predefined primary contexts are not covered in the scenarios, they shall also be taken into account. On the other hand, when no previous primary contexts are defined, the situational contexts of the feature will then be only based on the scenes from the collected scenarios.

Key Activity 3. Specify Ways of Interaction

If the identified feature is a new one, the team shall specify this new feature as well as its EWoI. When the feature is identified from the previous requirements document, the EWoI, as well as the related primary situational contexts will be inherited. On the other hand, the identified situational contexts from the collected scenarios will be matched with primary contexts. When a significant number of new situational contexts are identified from the scenarios, the new situational contexts will be seen as a primary context. Then the IWoIs for all primary contexts will be specified.

Key Activity 4. Identify Conflicts

Conflicts of interactions will be identified when both EWoI and IWoI are specified. For example, the EWoI of "playing a combat" feature of Vainglory (interrupting) has conflict with the IWoI of IVE situational context (intermittent). It indicates the users will mostly not have a good experience playing the game while he/she is watching an intensive ice hockey game (a scene for IVE).

Key Activity 5. Propose & Validate Change Requests

When conflicts of interaction occur, the changes shall be made to the feature in terms of the functional and quality requirements related in order to solve such conflicts. Each requirement with conflicting ways of interaction will be further analyzed. Changes are made, or new requirements are added, based on the collective understanding towards the nature of the features to comply those requirements with the IWoIs of the primary contexts. For example, to address the conflict in the previous example, one option is to change the gameplay towards the intermittent way of interaction, which provides ephemeral play session not interrupting the original activity.

The process shall continue through the whole maintenance phase. It continuously detects and addresses the potential conflicts in ways of interaction and maintain the mobility of the target apps towards the users' primary situational contexts.

5 An Example with FireStatus Mobile App

In this section, we apply the mobility requirements analysis approach into a change control process in the maintenance of a mobile app, called FireStatus. FireStatus is an

Android app designed to facilitate the emergent message transmission between the Emergency Response Centre and the Volunteer Fire Department. The client of the project, Perhon Palomiesyhdistys ry[5], is one of the Finnish Volunteer Fire Departments. In the volunteer fire department, firefighters are not present at the fire station, but at home, at work, etc. The FireStatus app enables the volunteer firefighters, who are in various contexts, to receive messages of emergency calls and respond quickly.

The app was developed in a student project[6]. Its main features include receiving and responding to emergency messages (Screenshot 1 in Fig. 2.), checking the responding situation of the incident calls (Screenshot 2), and various settings of the alarm sound, the keywords, incident dispatching number, etc. (Screenshot 3).

Fig. 2. Screenshots of the FireStatus App

When a volunteer receives an alarm message, he/she can respond with coming immediately, later or not coming by clicking one of the three buttons. The participation information is given in the Participant tab, which shows three firemen will come immediately (in the green bar) with their names and their skills shown in the gray area, another three comes in five minutes (in the yellow bar), one is unable to come (in the red bar) and nine are without any response (in the white bar). The settings tab shows the functions of changing alarm sound, changing vibration and so on.

Key Activity 1. Collect Scenarios & Identify Feature

We started an online interview with the end-users of the FireStatus app for their feedback on the application. Despite of the overall satisfaction on the app from the users, they did mention that the main feature of the app can be certainly improved to be more convenient. Therefore, tracing back to the original requirements document, we confirm the core feature of the app was recorded as Requirement 170057, which states *a volunteer firefighter shall be able to get information about the accident when receiving the emergency alarm and make decision on whether to participate and when*. Concerning this specific feature of the app, further questions concerning the scenarios of use are also forwarded to the users.

Key Activity 2. Identify Scenes & Match Context

---

[5] http://www.perhonpalomiehet.fi/

[6] http://www.uta.fi/sis/tie/pw/previous-projects.html

When a set of scenarios are collected with the situational contexts are easily specified (shown in Table 5). As the context analysis was not included in the development phase nor recorded in requirements document, the situational contexts acquired from the users' scenarios will be seen as the primary contexts. A typical scene is assigned to each situational context to ease the understanding of such contexts.

Table 5. The Scenes and Context Information Specified from Scenarios

ID	**Requirements**	**EWoI**
Requirement 170057	A volunteer firefighter shall be able to get information about the accident when receiving the emergency alarm and make decision on whether to participate and when.	Interrupting
Scenes		**Context**
Idle at home		AVE
Visiting a friend with family and having dinner		AVC
Watching an intense hockey game		IVE
Morning exercising with a friend		AWC
Receiving treatment from a doctor		IVC
Driving		ITC

Key Activity 3. Specify Ways of Interaction

Furthermore, we prioritize the acquired situational contexts by the occurrence of such context in the scenarios. For each situational context, one or more IWoIs are assigned based on the previous reference tables in Section 3 (shown in Table 6). The EWoI is defined as Interrupting.

Table 6. Conflicts of Interaction for Requirements 170057

Context	IWoIs	Conflicts
AVE	Interrupting, Intermittent, Accompanying	
IVE	Intermittent, Accompanying	Yes
AVC	Intermittent	Yes
ITC	Accompanying	Yes
AWC	Intermittent, Accompanying	Yes
IVC	Accompanying	Yes

Key Activity 4. Identify Conflicts

Therefore, by comparing the IWoIs of the situational contexts and the EWoI of the feature, we can easily identify the conflicts. As shown in Table 6, the interrupting nature of this core feature of the app only suits the situational context of AVE, which means that the user shall have a pleasant use of the app only when he/she is in a situation similar to "*being idle at home*". In other situations, for example, when the users are engaged in a social interaction (socially constraining, e.g., AVC, IVC), or when they have limited time to put attention and focus on reading the information (temporally intensive, e.g., IVE, ITC), intermittent or accompanying ways of interaction shall be more preferable by the users, towards which the feature shall be updated.

Key Activity 5. Propose & Validate Change Requests

Currently, considering the current design of the app, information of the occurred incident and the participants are separately displayed (Screenshot 1 & 2). This shall result in long concentration and interruption towards users' original action and should be improved. Furthermore, ways of acquiring information other than reading from the screen is also a good way to avoid interrupting. A set of change requests are proposed as new requirements shown in Table 7.

Table 7. Changes in Requirements 170057

Requirement ID	New Requirements
170057.1	The volunteer firefighter shall be able to view the incident information (e.g., location, severity, keywords, responded participants) and respond on one screen.
170057.2	The volunteer firefighter shall be able to listen to the read-out of the incident information.
170057.3	The volunteer firefighter shall be able to respond and change the response via voice control (e.g., Siri, OK Google)
170057.4	From opening the app to the response is sent, the duration shall be no longer than 3 seconds.

Among the new requirements, Requirement 170057.1 aims to reduce the interaction duration of the targeting feature and enable retractable action. In this way, the feature shall then provide more intermittent ways of interaction, which suits more for socially constraining situations and temporally intensive situations. Furthermore, to specify Requirement 170057.1 towards shortened concentration time and effective information acquiring, a set of sub-requirements can also be proposed, such as the notification sound varies according to the severity of the incident, the theme color varies according to the severity of the incident, the keywords related to the firefighter's skills shall be highlighted and shown at front, etc. On the other hand, Requirements 170057.2 and 170057.3 aim to create an alternative accompanying way of interaction for the feature by introducing information read-out and voice control, so that the users are then able to continue focusing on their original activity while acquiring the information and making decisions. In addition, non-functional Requirement 170057.4 adds further constraints on and emphasize the short interaction duration.

With the help of the app development team and the end users, we further validated the new requirements. The development team agree that the six scenes shown in Table 3 are indeed the most common situations. Furthermore, despite not recognizing any enhancement possibilities at the beginning, the developers do believe Requirement 170057.1, 170057.1.3 and 170057.4 are necessary considering the situational context of use. However, they also emphasize that "simplicity" is the key attribute of the app, as the app targets to provide the pure functional service for the users from a special domain, who might be confused with more features added. Therefore, they consider reading-out information and voice control features unnecessary. Concerning the end users, a majority of them are satisfied with the current version of the app when the rest express rooms for improvement but not dissatisfaction. Concerning the proposed new features, the end users also like the way to reduce interaction duration by condensing key information on single screen and using easily detectable visualizations. Moreover, like the indication from the developers, only a minority of the end users like the idea of voice-based info acquiring and responding.

6 Conclusion

As the key quality attribute of mobile apps, their mobility should be taken into account in both development and maintenance phase. When one change request is proposed, whether it will cause mobility loss is a question to be addressed before the app is updated. By using scenarios as a supporting tool, we propose the continuous mobility requirement analysis approach for app maintenance with the process model and relevant techniques, illustrating the way of maintaining the mobility of the app. Via the demonstration of the example of proposing change request to FireStatus app, we validate the usefulness of this approach in detecting the potential mobility loss hazard and addressing the issue by updating requirements. This approach will facilitate the mobile app maintenance practice by enhancing the understanding of the relation between mobility-impacting factors, e.g., the situational context and ways of interactions.

On the other hand, this study can be improved by a more thorough classification in each of the perspectives of situational context. Furthermore, a more explicit and unified reference model for detecting the conflicts in ways of interaction is also needed. A predefined scenario modeling approach shall also facilitate the process of acquiring context scenes and identifying features. A collaborating analysis tool can support this approach well. Further validation and evaluation of the approach are required with the support of empirical data. The relevant future empirical study shall also investigate the relation between the feature changes in maintenance and the mobility loss, as well as the factors to it. A prospective case study with real-life mobile companies shall be planned to address the aspects mentioned above.

Acknowledgments

We thank the development team of the FireStatus app, Perhon Palomiehet association, and Ossi Puustinen for their help in this study.

References

1. Mallat, N., Rossi, M., Tuunainen, V.K. and Öörni, A.: An empirical investigation of mobile ticketing service adoption in public transportation. *Personal and Ubiquitous Computing*, *12*(1), pp.57-65. (2008)
2. Kleinrock, L.: Nomadicity: anytime, anywhere in a disconnected world. *Mobile networks and applications*, *1*(4), pp.351-357. (1996)
3. Salo, M. and Frank, L.: User behaviours after critical mobile application incidents: the relationship with situational context. *Information Systems Journal*, *27*(1), pp.5-30. (2017)
4. Li, X. and Zhang, Z.: A User-App Interaction Reference Model for Mobility Requirements Analysis. The Tenth International Conference on Software Engineering Advances, Barcelona, Spain. *ICSEA 2015*, pp. 170 – 177 (2015)
5. Li, X., Zhang, Z. and Nummenmaa, J.: Models for mobile application maintenance based on update history. In *Evaluation of Novel Approaches to Software Engineering (ENASE), 2014 International Conference on* (pp. 1-6). IEEE. (2014)
6. Stepanova, E. and Kirikova, M.: Continuous Requirements Engineering for Mobile Application Development. In *REFSQ Workshops*. (2017)
7. Dey, A.K., Abowd, G.D. and Salber, D.: A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-computer interaction*, *16*(2), pp.97-166. (2001)
8. Belk, R.W.: Situational variables and consumer behavior. *Journal of Consumer research*, *2*(3), pp.157-164. (1975)
9. Schmidt, A., Beigl, M. and Gellersen, H.W.: There is more to context than location. *Computers & Graphics*, *23*(6), pp.893-901. (1999)
10. Lee, I., Kim, J. and Kim, J.: Use contexts for the mobile internet: a longitudinal study monitoring actual use of mobile internet services. *International Journal of Human-Computer Interaction*, *18*(3), pp.269-292. (2005)
11. Liu, Y. and Li, H.: Exploring the impact of use context on mobile hedonic services adoption: An empirical study on mobile gaming in China. *Computers in Human Behavior*, *27*(2), pp.890-898. (2011)
12. Liang, T.P. and Yeh, Y.H.: Effect of use contexts on the continuous use of mobile services: the case of mobile games. *Personal and Ubiquitous Computing*, *15*(2), pp.187-196. (2011)
13. Glinz, M.: Improving the quality of requirements with scenarios. In *Proceedings of the second world congress on software quality* (Vol. 9, pp. 55-60) (2000).
14. Hickey, A.M. and Davis, A.M.: A unified model of requirements elicitation. *Journal of Management Information Systems, 20*(4), pp.65-84 (2004).
15. Sutcliffe, A.G. and Ryan, M.: Experience with SCRAM, a scenario requirements analysis method. In *Proceedings of 1998 Third International Conference on Requirements Engineering, 1998.* (pp. 164-171). IEEE (1998).
16. Kaindl, H., Brinkkemper, S., Bubenko Jr, J.A., Farbey, B., Greenspan, S.J., Heitmeyer, C.L., do Prado Leite, J.C.S., Mead, N.R., Mylopoulos, J. and Siddiqi, J.: Requirements engineering and technology transfer: obstacles, incentives and improvement agenda. *Requirements Engineering*, 7(3), pp.113-123 (2002).
17. Sutcliffe, A., Scenario-based requirements engineering. In Proceedings of the *11th IEEE international Requirements engineering conference, 2003.* (pp. 320-329). IEEE (2003).
18. Weiser, M.: The computer for the 21st century. *Scientific american*, *265*(3), pp.94-104. (1991)

19. Lyytinen, K. and Yoo, Y.: Research commentary: the next wave of nomadic computing. *Information systems research*, *13*(4), pp.377-388. (2002)
20. Abowd, G., Dey, A., Brown, P., Davies, N., Smith, M. and Steggles, P.: Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing* (pp. 304-307). Springer Berlin/Heidelberg. (1999)
21. Jumisko-Pyykkö, S. and Vainio, T.: Framing the context of use for mobile HCI. Social and Organizational Impacts of Emerging Mobile Devices: Evaluating Use: Evaluating Use, 217. (2012)
22. Heijden, H.V.D. and Ogertschnig, M.: Effects of context relevance and perceived risk on user acceptance of mobile information services. *ECIS 2005 Proceedings*, p.7. (2005)
23. Bradley, N.A. and Dunlop, M.D.: Toward a multidisciplinary model of context to support context-aware computing. *Human-Computer Interaction*, *20*(4), pp.403-446. (2005)
24. Kristoffersen, S. and Ljungberg, F.: Mobile use of IT. In *the Proceedings of IRIS22, Jyväskylä, Finland*. (1999)
25. Eshet, E. and Bouwman, H.: Addressing the Context of Use in Mobile Computing: a Survey on the State of the Practice. *Interacting with Computers*, *27*(4), pp.392-412. (2014)
26. Chittaro, L.: Distinctive aspects of mobile interaction and their implications for the design of multimodal interfaces. *Journal on Multimodal User Interfaces*, *3*(3), pp.157-165. (2010)
27. Sutcliffe, A.: Scenario-based requirements analysis. *Requirements engineering*, *3*(1), pp.48-65. (1998)
28. Sutcliffe, A.: September. Scenario-based requirements engineering. In *Requirements engineering conference, 2003. Proceedings. 11th IEEE international* (pp. 320-329). IEEE. (2003)
29. Alexander, I.F. and Maiden, N. eds.: Scenarios,? Stories, Use Cases: Through the Systems Development Life-Cycle. John Wiley & Sons. (2005)
30. do Prado Leite, J.C.S., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G. and Oliveros, A.: Enhancing a requirements baseline with scenarios. *Requirements Engineering*, *2*(4), pp.184-198 (1997).
31. Coursaris, C. and Hassanein, K.: Understanding m-commerce: a consumer-centric model. *Quarterly Journal of Electronic Commerce*, 3, pp.247-272 (2002).
32. Skelley, T., Namoun, A. and Mehandjiev, N.: The impact of a mobile information system on changing travel behaviour and improving travel experience. In *Mobile Web Information Systems* (pp. 233-247). Springer Berlin Heidelberg (2013).
33. Hsia, P., Samuel, J., Gao, J., Kung, D., Toyoshima, Y. and Chen, C.: Formal approach to scenario analysis. *IEEE Software*, 11(2), p.33 (1994)
34. Di Giovanni, P., Romano, M., Sebillo, M., Tortora, G., Vitiello, G., Ginige, T., De Silva, L., Goonethilaka, J., Wikramanayake, G. and Ginige, A.: User centered scenario-based approach for developing mobile interfaces for social life networks. In *First International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE), 2012* (pp. 18-24). IEEE (2012)
35. Pohl, K.: Requirements engineering: fundamentals, principles, and techniques. Springer Publishing Company, Incorporated. pp.408-450 (2010)
36. Schmidt, K. and Simonee, C.: Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. *Computer Supported Cooperative Work (CSCW)*, *5*(2), pp.155-200. (1996)
37. Ljungberg, F. and Sørensen, C.: Overload: From transaction to interaction. *Planet Internet*, pp.113-136. (2000)
38. Kakihara, M. and Sørensen, C.: Expanding the 'mobility' concept. *ACM SIGGroup bulletin*, *22*(3), pp.33-37. (2001)