mConcAppt – A Method for the Conception of Mobile Business Applications

Steffen Hess, Felix Kiefer, Ralf Carbon, and Andreas Maier

Fraunhofer IESE
Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
{steffen.hess,felix.kiefer,ralf.carbon,andreas.maier}
@iese.fraunhofer.de

Abstract. Mobile business applications (mobile business apps) bear huge potentials for increased work productivity, work comfort, and even sales if they are of high quality. Usability and user experience, in particular, are among the key quality attributes. The high quality requirements of mobile business apps require them to be thoroughly engineered. Unfortunately, today's software engineering approaches are often too heavy-weight to allow developing high-quality mobile business apps in the context of mobile projects, which often face small budgets, extremely limited effort, and short time-to-market requirements. This paper presents mConcAppt, a user-centered and lightweight conception method for mobile business apps. It provides guidance for requirements engineering and interaction design for mobile business apps and provides interfaces to other activities, such as visual design, architectural design, implementation, and testing. The adoption of mConcAppt in various industrial contexts indicates that it enables organizations to elaborate a concept for a high-quality mobile business app in 2-4 weeks.

Keywords: Mobile Business Applications, User Experience, Interaction Design Method, Requirements Engineering, User-Centered Design.

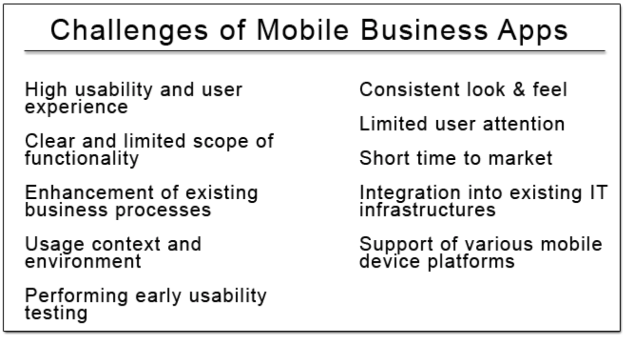
1 Introduction

More and more often, organizations try to exploit the potential of mobile applications (mobile apps) for their business. They do so to increase the work efficiency and work comfort of their employees (B2B scenario) and enhance sales by offering mobile services to their customers (B2C scenario) by means of mobile business apps. There is a huge potential in using not only mobile apps for personal information management (email, calendar, tasks) or general purpose apps, for instance to take notes, but also highly specialized mobile business apps built to support employees' specific work tasks or mobile customers. Both in the B2B and the B2C scenario, mobile apps must guarantee high quality, as they are used in potentially business-critical situations.

D. Uhler, K. Mehta, and J.L. Wong (Eds.): MobiCase 2012, LNICST 110, pp. 1–20, 2013.

Organizations aiming at developing mobile business apps and especially the involved requirements engineers and interaction designers are faced with the following challenges (see Fig.1).

High usability and user experience - Usability and user experience, in particular, are essential quality attributes of mobile business apps [11]. Users of mobile business apps expect to immediately put them to productive use. Typically, users will neither read manuals explaining the usage of a mobile app nor frequently use a help system or call support hotlines. They expect a mobile business app with a unique user experience that is tailored to their device and the underlying mobile device platform. This means, for instance, that available sensor data is leveraged wherever it makes sense, that multi-touch gestures can be used, that the mobile business app adheres to the user interface (UI) guidelines of the device platform, and that the app is well integrated with other apps of the respective ecosystem. In the absence of high usability and high user experience, mobile business apps will not be used frequently and will not generate the expected effects; they will get de-installed again quickly, and if they are publicly available, bad reviews will be posted in app stores.



Challenges of Mobile Business Apps

- High usability and user experience
- Clear and limited scope of functionality
- Enhancement of existing business processes
- Usage context and environment
- Performing early usability testing
- Consistent look & feel
- Limited user attention
- Short time to market
- Integration into existing IT infrastructures
- Support of various mobile device platforms

Fig. 1. Challenges of Mobile Business Apps

Clear and limited scope of functionality - Another key success factor of mobile apps we observed in practice is a clear and limited scope of functionality. Mobile users only perform specific tasks on the go. Therefore, a mobile business app should support its user in a small set of clearly defined tasks [16] and should not overwhelm the user with features supporting an overall workflow the way many stationary information systems do [3]. Hence, re-building complete feature sets of desktop applications on mobile devices is not appropriate. A limited and well-grounded set of functionalities of mobile business apps is crucial for their success and acceptance by users.

Enhancement of existing business processes – In organizational environments, existing business processes that bear the potential of being supported by mobile devices have to be identified first [23]. This potential identification must be directly interwoven with the conception of interaction designs to indicate possible mobile support opportunities efficiently. Identifying potentials does not only mean adding the mobile device to the existing business process; one must also focus on improving the existing process in an intelligent way, which might also change the existing process in general.

Usage context and environment – Environmental challenges are very broad. Besides temperature, light conditions, noise, distraction, mobility of the user, cognitive and physiological constraints of the user [19],[21], the actual usage context, in particular, has to be defined for mobile business apps. In contrast to traditional mobile apps, mobile business apps rely on a concrete organizational task, respectively business process, and therefore the actual usage context can be specified more precisely than "almost anywhere", as it is for traditional apps. In addition, the location awareness of the device and its consideration during conception constitutes a major challenge.

Performing early usability testing – Regarding usability testing, a major challenge for mobile business apps is related to performing tests in the actual usage context with real end users [8]. Usually, access to end users (e.g., pilots) is limited and the usage context often involves customers of the end user (e.g., passengers) who cannot be involved in the process for various reasons (e.g., confidentiality, availability).

Consistent look and feel – Regarding the overall portfolio of business applications used, the mobile business app has to fit into that portfolio to create a consistent look & feel for the end user. This also needs to address legacy software systems and gets even more challenging if more devices and operating systems are added to the portfolio (e.g., smartphone apps, tablet apps, desktop applications, applications for customized systems, etc.).

Limited user attention – Users of mobile business apps often have limited attention for the actual interaction they are performing with the device. In general, this leads to the challenge that the conception of mobile business apps has to consider whether the app will be used during primary tasks (e.g., access information on the tablet while traveling on the train) or secondary tasks (e.g., access information on the smartphone while talking to a customer).

Besides such specific challenges, there also exist more general challenges for the overall development organization (including requirements engineers and interaction designers). These challenges comprise, for instance:

Short time to market – Once it has been decided to bring a mobile app into the market, it has to be realized and delivered quickly. Otherwise, business opportunities can get lost, for instance, if customers buy competitor products that are available in the respective app stores earlier (specifically relevant for the B2C scenario).

Integration into existing IT infrastructures – Mobile business apps rely on data and services from backend systems, which are typically part of the already existing IT infrastructure of an organization [21]. This often requires modifications of backend services or even new backend services that specifically fulfill the requirements of mobile clients.

Support of various mobile device platforms – Particularly if mobile business apps are built for end customers, various mobile device platforms like Apple's iOS, Google's

Android, or Microsoft's Windows Phone must typically be supported. Even if a technology supporting multiple mobile platforms is used, the users of each platform expect a native user experience, i.e., the mobile business app must feel like a native iOS, Android, or Windows Phone app.

Limited hardware resources - Mobile devices bear some huge constraints, especially regarding the usability of mobile apps (e.g., small screen sizes, limited input facilities, limited device capacity, limited power supply) [1],[15].

These challenges reveal that mobile business apps must be thoroughly engineered to satisfy users, customers, and all other involved stakeholders. Unfortunately, we have observed in various industrial settings that mobile business apps are poorly engineered and do not provide appropriate quality. Organizations developing mobile business apps do not adopt proven software engineering methodologies in the development of mobile business apps, as they appear too heavyweight for developing a piece of software that is relatively small compared to desktop products or overall IT infrastructures. Hence, mobile software engineering has to provide tailored engineering solutions for mobile business apps.

This paper introduces mConcAppt (used as an abbreviation for "method for the conception of mobile business apps"), a method for the conception of mobile business apps. Here, conception subsumes requirements engineering as well as UI & interaction design. This integrated view of the two disciplines of software engineering originates from the TORE approach [2]. The method is user-centered to allow striving for high usability and user experience and to develop apps that fit the needs of its users regarding their scope of functionality. It is lightweight to support organizations in efficiently developing mobile business apps and meet their effort and time-to-market constraints. The conception method is interrelated with other activities in the development cycle. It has interfaces to business analysis, visual design, architectural design, implementation, and testing.

The remainder of this paper is structured as follows. Section 2 provides an overview of related work. In Section 3, we introduce the mConcAppt method illustrated on an example project. Section 4 describes lessons learned during the application of the method in industry projects. Section 5 concludes the paper and gives an outlook on future work.

2 Related Work

Without a doubt, usability and user experience are one of the major challenges for mobile business apps. High usability and user experience is required to gain acceptance by the targeted end user group. Thus, usability and user experience are mandatory success factors for a mobile business app. To guarantee these success factors, user-centered design processes for software design and development are common practice. However, only few approaches exist today that shift their focus from a general user-centered approach towards a user-centered approach for mobile apps. A number of scientists also highlight the need for specific mobile software development methods due to the specific challenges in the mobile context (e.g., usage of integrated sensors, native, hybrid, and web application development, fast changing surroundings, hard-to-perform evaluations)[1], [7], [24].

Mayhew describes a holistic approach to a user-centered iterative design process, the "Usability Engineering Lifecycle" [20]. It is divided into three phases: requirements analysis, design/testing/development, and installation. The requirements analysis includes user research, task analysis, usability goal settings, and general design descriptions. In the second phase (design/testing/development), the previously elicited requirements are refined and afterwards designed and evaluated in three different states of maturity. The third and final phase (installation) completes the approach with the installation of the product and uses the user feedback occurring after some time to enhance the design, which will then be incorporated into the next releases of the product. The usability lifecycle is a very detailed description of a general approach to usability engineering activities during software development. The mConcAppt approach is oriented towards the usability lifecycle but with a narrower focus on the challenges of mobile business app development. To address these, the single phases of mConcAppt and the resulting artifacts are tailored to the needs of mobile app development. Similar to the usability lifecycle, several other approaches exist (e.g.,[10]) that cover the complete lifecycle of the product with a start-to-finish method. We found that these methods constitute a common ground for interaction design but are too general to be applicable, especially regarding the challenges of mobile business apps. mConcAppt overcomes this gap by tailoring existing concepts from those methods in an applicable manner to allow them to be used under the given challenges of mobile business apps.

Grill et al. describe a pattern approach to mobile interaction design [12]. The approach starts with the elicitation of requirements with a focus on the analysis of the mobile environment. In this case, the term mobile environment means the environment where a mobile scenario takes place. The mobile technology, the users, as well as data and information are parameters for the mobile environment. After being elicited, the parameters are used in a second step as the basis for the upcoming interaction design process (IxD-Process). The IxD-Process is iterative and consists of building numerous design drafts followed by a formative evaluation. Beneath common tools for interaction design like personas and scenarios, the interaction designer has tool-based access to a library of mobile interaction pattern. Thus, the interaction designer is able to use common practices in the field of mobile interaction design to generate a design solution. The evaluation of the interaction designs created in this tool has shown that especially expert users of mobile devices have some trouble with the interaction because of their concrete experiences and expectations regarding interaction with a mobile device. Our experience shows that the use of patterns for mobile interaction design may lead to underestimating the importance of the actual usage context because of the use of general solutions for known design problems. To avoid this, the mConcAppt method aims for direct communication between the interaction designer and the user of the app in order to address the context of use during the whole process and at every step where design decisions are made. This aspect is also not addressed sufficiently in the pattern-based interaction design method.

De Sá and Carrico describe in [7] lessons learned from the design processes of several mobile apps. They emphasize three stages of app development that lead to particular challenges in terms of mobile devices, namely data gathering, prototyping, and evaluation. In [8], they describe their own design methodology for mobile devices. This methodology contains three phases, pervasive data gathering, early stage prototyping, and mobile evaluation. The first phase is used to gather data and elicit requirements in order to sufficiently address the context of use. The early stage prototyping phase is used to develop prototypes based on the gathered information. These prototypes are developed for evaluation in-situ to address the challenge of context of use again. Such evaluations take place during the last phase of this methodology. To keep the conditions as realistic as possible, the evaluation is conducted in the previously elicited context of use. The authors do not provide techniques or a start-to-end method in their approach. Furthermore, they complete the description of every phase with resulting guidelines from their experience with various development processes. These guidelines focus on the challenges of mobile apps. Thus, this methodology and these guidelines are promising, but they lack concrete guidance and templates, which are provided by mConcAppt. Thus, the mConcAppt approach can also be applied by mobile interaction designers without much experience. Another aspect is that the main focus of this methodology is the in-situ design, which tackles the context of use challenge. In general, it is a very good practice to elicit requirements for mobile apps and evaluate them in the actual context of use. But we found that limited budgets for mobile apps often hinder such field studies. In addition, mConcAppt could be easily enhanced with requirements elicitation techniques such as contextual inquiries to gather information prior to the requirements workshop. However, this is not mandatory for the achievement of successful interaction design by using the mConcAppt approach.

Newer approaches (e.g., [5],[9],[22]) often provide detailed guidance on how to build an app for a certain mobile device or operating system. Unfortunately, they are often not tailored to the specific needs of mobile business apps and do not provide any concrete method for developing an interaction design.

3 mConcAppt Method and Example Project

The mConcAppt method combines requirements engineering and UI & interaction design activities for mobile business apps and produces a so-called interaction concept as a basis for implementation and further activities. This section starts with an overview of the mConcAppt method and then describes each phase of the method in detail.

3.1 Overview

Regarding our user-centered approach, the mConcAppt method is in the center of the overall app development process, acting as a mediator between all activities. Requirements engineering and UI & interaction design activities must be closely interwoven with other engineering activities like architectural design to ensure that the required usability and user experience can be supported by the architecture and to address trade-offs with other quality attributes besides usability and user experience, which are of course important as well. By defining clear communication interfaces between the different software engineering activities, mConcAppt acts as driver for engineering mobile business apps with high overall quality [13].

Upstream activities (e.g., generating the general app idea, resolving organizational issues, and clarifying responsibilities) comprise all activities that take place prior to the actual app development activities. Downstream activities (e.g., final app development, app distribution) comprise activities that take place after the final interaction concept is delivered. This paper focuses on the description of the mConcAppt method and its phases as shown in Fig. 2. The app conception as a whole is an iterative process. We describe in detail every activity to be performed in one complete iteration of the method.

The initial phase of the mConcAppt method, called *Elicit Requirements,* comprises the gathering of requirements that are especially relevant for the interaction design. The phase *Specify Interaction Design* comprises the actual development of the interaction concept in tight collaboration with responsible persons from other software engineering disciplines. To support this collaboration, the method comprises a *Communicate Interaction Design* phase dedicated to showing how the evolution steps of the interaction concept should be communicated to the roles involved in the app development. The rationale for this communication is to get early feedback from other involved roles and to enable the project stakeholders to perform their activities as early as possible. In this case, the architect and the business analyst are the most important information sources regarding the interaction concept because they provide much information and many constraints that might be considered in the concept. Only after this internal communication and the approval by the project management is the interaction design validated with actual end users in the *Validate Interaction Design* phase.

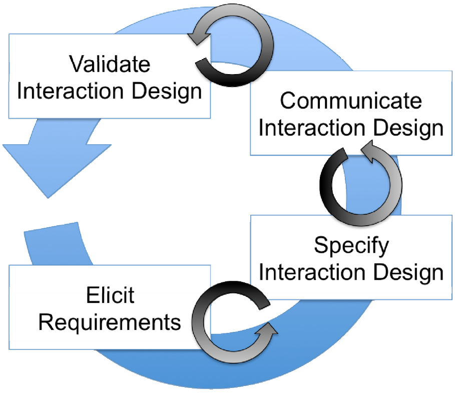


Fig. 2. mConcAppt Overview

The mConcAppt method has been applied in several industry settings and refined based on the practical experiences made there. In the following, the method is described using examples or excerpts of one of our in-house app engineering projects. The mobile business app developed in this project aims to assist employees in tracking information required for travel expense reports while on a business trip and in creating travel expense reports after a business trip. It has been developed for the iPhone. The app is intended to be used as a showcase project in exhibitions to illustrate mobile software engineering.

3.2 Elicit Requirements

The *Elicit Requirements* phase comprises all requirements engineering activities that have to be performed after the first management decision has been made regarding a specific app. Fig. 3 shows the sequence of activities in the first iteration of this phase and the time typically required for each activity based on our experiences gained when applying the method in industry projects. The aim of the activities of this phase is primarily to prepare and conduct a requirements elicitation workshop and to document the elicited results.

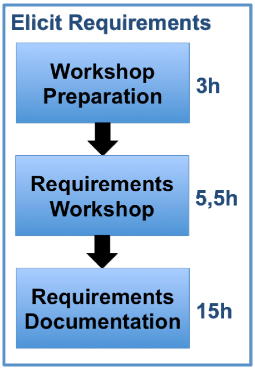


Fig. 3. mConcAppt Elicit Requirements Phase

The workshop preparation subsumes all activities that are necessary prior to conducting the workshop. These activities include creating a workshop agenda, eliciting initial information, and selecting and inviting the participants. The workshop agenda should be aligned with the intended outcomes (see Fig. 4) and allocate enough time for each step. Each step shown in Fig. 4 should be an explicit item on the agenda.

The participants should be selected following Table 1. In particular, the selection of appropriate (lead) users is essential for a successful workshop. The participants should be invited at least one week in advance.

In addition, the workshop organizer has to make sure that the required material for the workshop is prepared (flip charts, beamer, pencil, cards) and that the workshop room is set up to give participants a pleasant environment. Fig. 4 shows an overview of the steps performed during the workshop and the resulting artifacts. The documentation of the workshop results in the form of artifacts is supported by templates provided by mConcAppt.

In our example project, we identified six different stakeholder groups (company employee, developer, travel management, exhibition visitor, company steering committee, and business area manager). We documented their roles and goals and described them using a template. After identifying the company employee as the main target user group, a user persona [6] was developed. The chosen persona should be an instantiation of the user group that represents the most common user (e.g., the 80% user) in the best possible way. The persona describes various goals and behavior patterns of potential users. It also encapsulates and explains critical behavior data in a way that other stakeholders can understand, remember, and relate to. The persona is created during the workshop directly in collaboration with the participants of the workshop, usually in combination with factors that describe the general usage context, especially existing hardware and software experiences. During the remainder of the workshop and throughout the entire development process, this persona will represent the user in upcoming discussions. In our example, the persona describes a team leader in our company. The role of a team leader was selected because the estimated time a team leader spends on business trips is about 40% of his/her total work time. This makes the team leader one of the most important and representative users of the app.

Table 1. Workshop Participants

Participant (Quantity)	**Description and role in the workshop**
Interaction Designer/ Requirements Expert (1-2)	Workshop conception and preparation; moderator of the workshop
(Lead) User (2-4)	Gives input from a user's perspective. The lead user approach should be chosen if no real users are available or if the system offers high innovation potential. "Lead users face needs that will be general in a marketplace – but face them months or years before the bulk of that marketplace encounters them, and lead users are positioned to benefit significantly by obtaining a solution to those needs." [14]
Project Manager (1)	Gives input from the project management perspective. This role is filled by the internal decision maker and the person responsible for the project.
Business Analyst	Takes the minutes of the workshop (the interaction designer is usually not able to make complete notes about the given input because of his moderating tasks); input from business perspective with regard to feasibility can be integrated directly into the minutes to prevent interrupting idea creativity during the workshop.
Customer (1)	Gives input and constraints from a customer perspective. This role is occupied by the decision maker of the customer. This is usually the responsible project management person on the customer's side. Usually this person is not a representative user of the system, although he or she often thinks that he/she knows what the users want.

After the elicitation of the persona and the stakeholders, the as-is-situation including current problems has to be elicited for the tasks that are to be supported by the mobile business app to be developed. In our example, we analyzed the as-is situation of tracking information while on a business trip and creating travel expense reports after returning from the trip. The analysis of the elicited as-is situations revealed several problems that could be tackled with the app:

- The travel expense report requires a lot of information (e.g. locations, dates, distances driven, and mileage) that is not captured during the business trip. In most cases, the traveler forgets to write it down or cannot do so because of a lack of time.
- Often it is not possible to fill out a travel expense report immediately after returning (daily business). This results in a high cognitive load because all information needs to be remembered until the report is created.
- It takes a lot of time to fill out travel expense reports.
- Even business trips without any costs need to be reported.
- Receipts and vouchers from costs incurred during a trip have to be glued on a piece of paper and need to be scanned to digitalize them for further organizational work.
- Each employee fills out travel expense reports in their own way (e.g., crossing out unused parts of the report). This causes additional organizational work in the administration department.

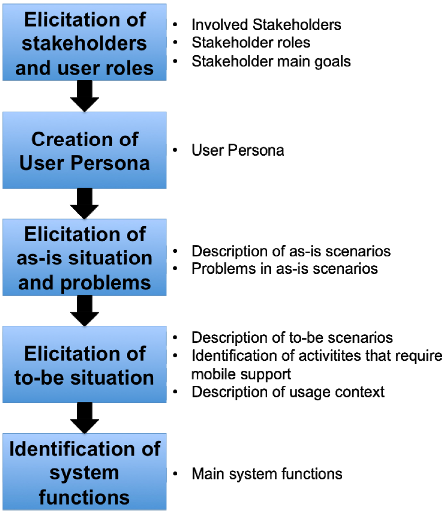

Elicitation of stakeholders and user roles
• Involved Stakeholders
• Stakeholder roles
• Stakeholder main goals
Creation of User Persona
• User Persona
Elicitation of as-is situation and problems
• Description of as-is scenarios
• Problems in as-is scenarios
Elicitation of to-be situation
• Description of to-be scenarios
• Identification of activitites that require mobile support
• Description of usage context
Identification of system functions
• Main system functions

Fig. 4. Steps performed during Requirements Workshop

In order to solve these problems, the to-be-situation was elicited in the workshop. Typically, employees are carrying a mobile device with them during a business trip, which makes it the ideal platform to support precise tracking of business trips. It is, for instance, required to track the exact time when and where (home or office) the business trip starts, when the business appointment starts, etc. The user can track such events of a business trip in real-time, for instance, by interacting with the mobile device when the trip starts. But even better, a mobile business app could recognize such events including locations and start times automatically and propose them to the user. This is possible by aggregating sensor data from the device and up-front knowledge about the respective business trip, for instance from the itinerary. The proposals of the app can be accepted or corrected by the user at any time during the business trip or even afterwards.

Table 2. As-Is and To-Be Scenario Template

Item	Description
Context	Context that leads to the actual task.
Precondition	Precondition for task performance
Step 1-N	Steps that are performed including problems occurring per step.
Postcondition	State that is achieved after the scenario.

In addition to tracking events like the start time of a trip, the mobile app can enable the user to collect artifacts (e.g., tickets and vouchers) during the business trip that are required afterwards for travel expense reporting. If the user uses public transportation and buys a ticket, a picture of the ticket can be taken and added to the travel documentation.

Both the as-is situation and the to-be situation are described using the template shown in Table 2. In addition, steps of the to-be situation that need mobile assistance as well as steps that do not necessarily need to be performed with the support of a mobile business app are identified to get a first indication regarding the mobility potential of the overall process.

The identification of the main system functions completes the workshop. These system functions are based on the previously elicited to-be situations. In addition, exchanged data is identified with the support of the business analyst, who can provide exact information about exchanged data and data formats based on this initial elicitation. The main system functions (e.g., time tracking and collection of travel artifacts) represent the core functionality of the system that has to be designed in the first iteration. During the workshop, it is sufficient to simply name the basic system functions to define the scope of the app.

The requirements documentation processes information gathered during the workshop in a way that allows other involved project stakeholders (see Table 1) to understand the elicited information and design decisions. The workshop documentation is part of the interaction concept description and the basis for all upcoming activities. It can be seen as a first draft version of the concept and documents all elicited information in a structure similar to the workshop agenda. It is a lightweight documentation, focusing on the information that is needed for further steps. mConcAppt also provides templates for the structure of the document. This temporary document can be seen as the first version of the interaction concept. During further iteration cycles of mConcAppt, requirements are elicited as output from the phases *Communicate Interaction Design* and *Validate Interaction Design* and integrated directly into the interaction concept. This workshop is only performed once in the initial phase.

Performing this initial interaction design requirements elicitation phase took 23.5 hours in total. Detailed time estimations for particular phases are shown in Fig. 3.

3.3 Specify Interaction Design

The specification of the interaction design (see Fig. 5) comprises the construction of the actual interaction concept, without consideration of the underlying backend system in the first iteration. The mobile business app is specified based on information elicited during the initial *Elicit Requirements* phase. First, the key functionalities are derived from the to-be situations. These key functionalities form the starting point of the app and should be represented by the first specified interaction cases. During the specification of the interaction cases, the flow of interaction cases is assembled step by step. After this specification, wireframes are created based on the interaction cases. Analogous to the flow of interaction cases, the actual screen flow is assembled.

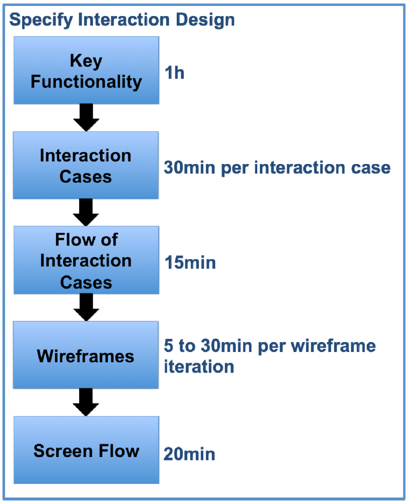

Specify Interaction Design
Key Functionality
1h
Interaction Cases
30min per interaction case
Flow of Interaction Cases
15min
Wireframes
5 to 30min per wireframe iteration
Screen Flow
20min

Fig. 5. mConcAppt Specify Interaction Design Phase

In our example project, we took the to-be situation and the main system functions derived from the workshop and identified key functionalities. Key functionalities are tasks that really have to be performed in a mobile way in the given context. There is always a huge risk to come up with too many functionalities based on existing desktop or legacy systems. Ask yourself: “What must really be done on the mobile device?” In our particular case, the key functionalities were limited to time tracking, artifact tracking, and information support for the traveler during a business trip. We explicitly decided not to provide functionality to edit the travel expense report on the mobile device, as the user cannot easily accomplish this on a smartphone. Based on the given information, seven main interaction cases were identified and described, similar to the example shown in Table 3, which is very similar to use case descriptions (e.g., [17]) but focuses only on information needed to specify the interaction concept. Additionally, the usage context is described in detail for each interaction case. According to agile development processes, one interaction case comprises several user stories that are related to a certain task. Usually, exactly one human action and one system action (see Table 3) should form a user story.

When creating the interaction cases, no concrete design solutions are developed; rather, the concept of user interaction is shown. Fig. 6 shows an excerpt of the flow of interaction cases for this particular example project. Arrows are used to indicate the user flow between different interaction cases and swim lanes might be used to indicate logically divided app contents. All interaction cases are shown in a graphical representation, which allows getting a quick overview of the general structure and function range of the app. Based on the flow of interaction cases, the final decision about the scope of the app is made by the interaction designer in conjunction with the project management.

Table 3. Interaction Case Description Example

Item	Description
ID	IC2: Track Time
Usage Context	A business trip for multiple days. Meeting an industry partner in Leipzig. The user is on his way to the hotel and uses the device while walking.
System Action 1	The system recognizes that the user arrives at his destination and notifies him that the time of arrival is tracked (via notification center).
Human Action 1	The user taps the notification to directly open the app.
System Action 2	The system opens the app and immediately shows the current trip itinerary and proposed time.
Human Action 2	The user confirms the proposed time.
System Action 3	The system provides feedback about the confirmation to the user.
Human Action 3	The user closes the app.
Postcondition	The arrival time is persistently stored.

Using the interaction cases and the flow of interaction cases as a basis, the actual wireframes are created. In first iterations, wireframes are created using paper and pencil. We found that this is the quickest way to create several versions of a screen and to try out different solutions. To guarantee traceability, we use a wireframe template that shows screen identifier, name, version of the wireframe, and date of production. Based on these first paper prototypes, prototypes in other sketching tools (e.g., Balsamiq, PowerPoint, Photoshop) might be created depending on the designers' preference, experience, and time constraints. Fig. 7 shows several versions of the same screen to demonstrate how the prototype evolves in iterations from initial wireframe sketches to the final app design.

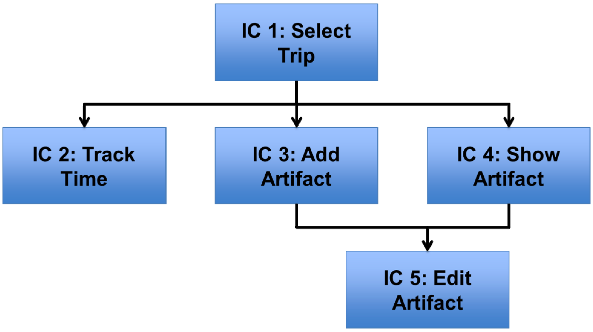


Fig. 6. Flow of Interaction Cases (excerpt)

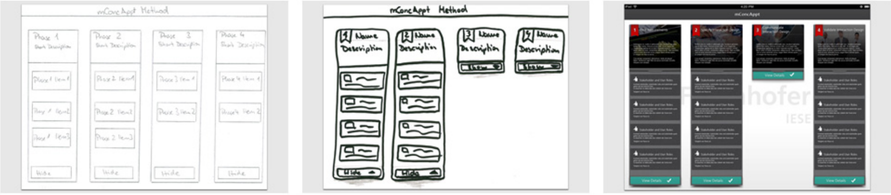


Fig. 7. Evolution stages from first paper mockup to final app

While creating wireframes, the screen flow is assembled. In the interaction case example of "track time" (see Table 3), the simple screen flow only consists of two screens. Typically, the complete screen flow is more complex and represents a holistic view of the complete navigation tree of the interaction concept. Arrows show concrete navigation possibilities, while text along the arrows provides a description of the intended screen transitions. The results of the steps *Elicit Requirements* and *Specify Interaction Design* represent the first final version of the interaction concept. Performing this phase in the given example project took ten hours in the first iteration; time estimates for individual steps are given in Fig. 5.

3.4 Communicate Interaction Design

In this phase, the interaction design is communicated to the other project stakeholders (see Fig. 9). We found that a video explaining the interaction concept is a helpful instrument for communicating it to the other stakeholders and propose this as a best practice for sharing the interaction concept in distributed project teams. Nevertheless, producing a video is an optional step in mConcAppt, especially if the team is not locally distributed. The focus is on exchanging information between all stakeholders and gaining their feedback, which might be transferred to requirements for the interaction design.

To produce the video, the interaction designer must first write a presentation script. Based on this script, snippets are recorded for each interaction case and a complete movie is assembled using these snippets. Once the complete movie is rendered, the interaction designer can distribute the video to the whole project team.

The presentation script is assembled based on the interaction cases and wireframes. The presentation script supports especially inexperienced interaction designers by creating a structured video and is an important part of the interaction design itself, as it documents ideas and reasons for design decisions in a way that allows other project stakeholders to get the story behind the concept and make decisions based on the script, the video, and the overall interaction concept description. It is also important to describe design decisions that are not seen on the actual wireframe at first glance (e.g., screen transitions, gestures, required input feedback, and fancy stuff). Experienced interaction designers might skip this step by performing the activities described above on the fly while recording. Nevertheless, documentation is important for further steps during the development process.

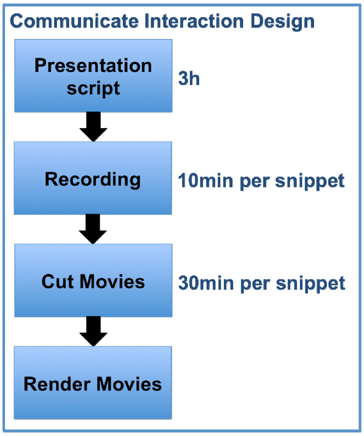


Fig. 8. mConcAppt Communicate Interaction Design Phase

In our example project, we recorded seven snippets according to the given interaction cases. We used a fixed-mounted camera and a pencil to point to elements. The overall length of a snippet was no longer than five minutes and the total video was twenty minutes in length. To prepare for the cutting and assembling of the final video, a template for headlines and overlays that is reusable in other projects is used. Sketches are scanned and/or imported into the authoring software (e.g., iMovie) to show the corresponding wireframe in full screen mode before explaining it. Using this practice, other stakeholder groups are able to easily discuss the actual version of the interaction concept and give feedback on it as well as make decisions based on it. Creating the interaction video for the example project took eight hours in total.

3.5 Validate Interaction Design

As shown in Fig. 2, a real user test is done after the first complete iteration of the method. Fig. 10 shows an overview of the activities during the Validate Interaction Design phase. Based on the interaction cases, concrete usage scenarios [4] form the basis for conducting the test. In addition, a clickable prototype can be created easily using presentation software for non-linear presentations on the mobile device (e.g., Presentation Link). This approach requires marginally more time than applying Wizard-of-Oz testing [18] with paper prototypes, but enables us to perform a user review on the actual end device in the concrete usage environment, which we think is mandatory especially for mobile business apps.

Since the focus was on a small set of features in our example project, user testing was possible very early. We used an interactive on-device prototype with clickable scanned wireframes. The user review was done using a scenario-based approach and we abstained from recording the review. During the next two weeks, the prototype was improved based on the findings and evaluated two more times. For each user review, we chose three to four users.

In the following section, major lessons learned from adopting the mConcAppt method in projects with various industrial customers are presented.

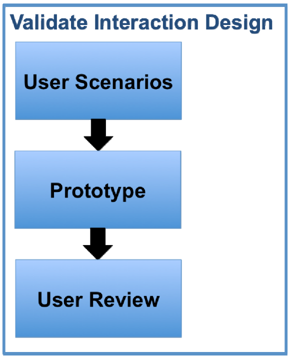


Fig. 9. mConcAppt Validate Interaction Design Phase

4 Lessons Learned

4.1 Elicit Requirements

One of the major lessons learned while using the mConcAppt method is that only one requirements workshop is needed to execute the further activities. However, it is mandatory that all roles described in Table 1 take part in the workshop. Leaving out one of the roles would lead to time-consuming rework and result in lower return on investment. Furthermore, the workshop has to be prepared optimally – information elicited in advance (e.g., observing the users in their as-is situation) increases the quality of the intended outcome. During the workshop, the competence of the moderator significantly influences the quality of the results. The moderator has to adapt to the customer problems and should have a basic understanding of the technical capabilities. Users participating in the workshop often do not discover the potential for mobile support in general, and especially the technical capabilities of the devices are unknown. Potential users of mobile business apps are often constricted by existing workflows, which indeed represent an obstacle to better mobile support.

4.2 Specify Interaction Design

Regarding the specification of the interaction design we learned that it is mandatory that the interaction designer himself is a user of the addressed device class and operating system. To provide high-quality results, it is not sufficient to know operating system programming guidelines. When starting to design the mobile business app, it is a key influencing factor to decide which functionality has to be realized in the backend system and which functionality has to be on the mobile device. Early communication of ideas and sketches across colleagues increases the quality of the results. We found that the interaction designer, who is usually just one person in the case of mobile business apps, needs to step out of his sandbox and obtain quick informal feedback from colleagues or other available persons. While communicating iterations to other project stakeholders, we observed that there is a large potential conflict area between interaction design and architecture. Software architects tend to refuse ideas and concepts due to technical reasons (e.g., enabling live search, which might be a large boost to user experience but requires significant resources in the backend). Especially user experience enabler and architecture are influencing each other significantly. Regarding mobile business apps, the frontend, which is usually represented by an app, strongly depends on the performance of the backend system. This dependency may have a major impact on the interaction design and creative solutions might be necessary. Project management should be aware of these special circumstances and solve conflicts regarding the overall goals of the project as well as provide prioritization between system capabilities and financial budget.

4.3 Communicate Interaction Design

In the interaction design communications phase, we introduced video and textual communication in combination as the preferred approach, since product teams are often distributed both spatially and temporally. While face-to-face communication should be preferred whenever possible, insisting on lightweight documentation as described will increase the quality of the product.

4.4 Validate Interaction Design

A user review should be performed with four users. For further iterations, the user review team should be composed of two users who took part in previous activities and two users who were not part of previous activities. This composition allowed us to trace whether previously discussed issues had been addressed in an applicable way and discover new issues that might not have been discovered by the users who had taken part in the creation of the concept. Performing the user review on the actual device in the given usage context leads to more reliable feedback than performing tests in a clinical environment using paper mockups.

4.5 General Lessons Learned

It is not beneficial if one person takes over different roles during the workshop because conflicting interests cannot be identified and elaborated. Furthermore, it is often laborious to convince users that they do not have to be able to perform every activity that can be performed with a legacy stationary device while being mobile. The mobile device usually serves as an additional device that supports especially those activities that need to be performed while being mobile.

The complete development of an interaction concept using mConcAppt ordinarily takes two to four weeks, and three iterations were usually found to be sufficient. Concerning the effort and times estimated in the description of mConcAppt, we have to admit that times correlate with the experience of the interaction designer, creative steps cannot really be projected, coordination and communication between the different development activities still require too much time, and mobile business apps have varying scopes and challenges.

5 Conclusion and Future Work

The mConcAppt method guides its users in performing requirements engineering and UI & interaction design activities for mobile business apps. It aims at addressing the challenges for the engineering of mobile business apps mentioned in the introduction. Thus, *high usability and user experience* are addressed by the overall engineering approach of mConcAppt. A *clear and limited scope of functionality* is explicitly considered during the interaction design specification. *Enhancement of existing business processes, usage context and environment*, as well as *consistent look and feel* are ensured by the systematic evaluation of the as-is and to-be situations in combination with prospective end users. *Performing early usability testing* is ensured by conducting concrete user tests after each iteration cycle of the concept similar to functional testing in agile development processes.

Dealing with *limited user attention* is still not explicitly addressed by mConcAppt, but should be considered in future work. The general challenges as mentioned in Section 1 are implicitly covered by the complete mConcAppt method with the essential approach followed during the design of the method being user-centered and lightweight. Even if our practical experience with mConcAppt is mainly based on the development of mobile business apps, the circumstance of addressing general challenges enables mConcAppt to be applicable in mobile app development in general. The main requirements engineering work is performed in the context of a one-day workshop, respectively its pre- and post-processing. The workshop strongly focuses on the requirements of (lead) users of the mobile business app and elicits as-is and to-be situations, but also involves, for instance, customers, business analysts, or architects. The results of the workshop depend on its participants, its moderator, and the planning performed up-front (see up-front activities of mConcAppt description). Our experience from various applications in industry shows that by conducting the workshop according to the fixed agenda and adhering to the additional guidelines proposed in the mConcAppt method; at least 80% of the user requirements can be elicited in the one-day workshop and feedback from other involved stakeholders can be gathered directly. If required, the elicitation of the remaining requirements and the discussion of open issues can be done after the workshop, typically in short phone conferences.

Interaction design can already start in the requirements engineering workshop, for instance if examples of screens are quickly sketched to illustrate certain ideas. But the main interaction design work is performed by the interaction designer in the back-office in close collaboration with the lead users and other involved stakeholders. Feedback on micro-iterations during interaction design can be discussed with a colleague of the responsible interaction designer. This often takes only some minutes but can be extremely helpful. Lead users and other stakeholders are also often asked for feedback. One week after the requirements engineering workshop at the latest, first sketches of the interaction design should be provided to them for feedback.

Our experience has shown that the mConcAppt method supports organizations in delivering an interaction concept ready to be implemented within two to four weeks. Based on the interaction concept, development teams were able to produce a first release of the mobile business app fulfilling the required core functionality and the usability and user experience requirements in two more weeks. Our customers were always very much satisfied with the duration of four to six weeks until a first release of a new mobile business app could be delivered.

The mConcAppt method has various interfaces to other activities like visual design, architectural design, implementation, or testing. As part of our future work, such interfaces need to be defined and elaborated in more detail. The artifacts exchanged between interaction designers, architects, and other stakeholders in the development process must be specified more precisely as well as the process of collaboration, for instance when to exchange which kinds of artifacts, provide feedback, etc. The communication interface to architectural design, in particular, needs to be addressed. As integration into existing IT infrastructures was mentioned before as a major challenge, interaction designers and architects should align solution ideas early on to assure that the IT infrastructure can fulfill the requirements arising from such solution ideas.

Producing great mobile business apps requires creativity. During the requirements engineering workshop and the elaboration of the interaction concept, creativity is required to define innovative solutions. In the future, the mConcAppt method will be extended by means of specific creativity techniques that can be adopted during the requirements engineering workshop to facilitate the elaboration of innovative solutions for mobile business apps.

As mentioned in the introduction, *support of various mobile device platforms* is another challenge for mobile business apps. The mConcAppt method will be extended to allow coping with the specifics of various mobile device platforms. In the project UID4Mobile, solutions are being developed to address the scalability of interaction concepts to different mobile device platforms. Scalability means the instantiation or tailoring of interaction concepts or designs to different mobile device platforms like Apple's iOS, Google's Android, or Microsoft's Windows Phone, but also to different device types for each platform, such as smartphones and tablets.

References

1. Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme, T., Jäälinoja, J., Korkala, M., Koskela, J., Kyllönen, P., Salo, O.: Mobile-D: an agile approach for mobile application development. In: Proc. OOPSLA 2004, pp. 174–175. ACM Press (2004)
2. Adam, S., Doerr, J., Eisenbarth, M., Groß, A.: Using Task-oriented Requirements Engineering. In: Proc. RE 2009, pp. 267–272 (2009)
3. Andersson, B., Henningsson, S.: Developing Mobile Information Systems: Managing Additional Aspects. In: Proc. ECIS 2010 (2010)
4. Carroll, J.M.: Making use: Scenario-based design of human-computer interactions. The MIT Press, Cambridge (2000)
5. Clark, J.: Tapworthy: Designing Great iPhone Apps. O'Reilly Media, Inc. (2010)
6. Cooper, A.: The Inmates Are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity, Indianapolis, USA (1999)

7. de Sa, M., Carrico, L.: Lessons from early stages design of mobile applications. In: Proc. MobileHCI 2008, pp. 127–136. ACM Press (2008)
8. de Sa, M., Carrico, L., Duarte, C.: Mobile Interaction Design: Techniques for Early Stage In-Situ Design. In: Asai, K. (ed.) Human Computer Interaction: New Developments. InTech (2008)
9. Ginsburg, S.: Designing the iPhone User Experience: A User-Centered Approach to Sketching and Prototyping iPhone Apps. Addison-Wesley Professional (2010)
10. Goodwin, K.: Designing for the Digital Age: How to Create Human-Centered Products and Services, 1st edn. Wiley (2009)
11. Gorlenko, L., Merrick, R.: No wires attached: Usability challenges in the connected mobile world. IBM Systems Journal 42(4), 639–651 (2009)
12. Grill, T., Biel, B., Gruhn, V.: A Pattern Approach to Mobile Interaction Design (Die Verwendung von Patterns beim Entwerfen mobiler Applikationen). IT - Information Technology 51(2), 93–101 (2009)
13. Hess, S., Kiefer, F.: Quality by Construction through mConcAppt – Towards Using UI-Construction as Driver for High Quality Mobile App Engineering. In: Proc. of QUATIC 2012 (2012)
14. Hippel, E.: Lead Users: A Source of Novel Product Concepts. Management Science 32(7), 791–806 (1986)
15. Huang, K.-Y.: Challenges in human–computer interaction design for mobile devices. In: Proc. of the World Congress on Engineering and Computer Science 2009, San Francisco, USA, October 20-22, vol. 1 (2009)
16. iOS Human Interface Guidelines. Technical report, Apple Inc. (January 2011)
17. Jacobson, I.: Use cases - Yesterday, today, and tomorrow. In: Software and Systems Modeling, pp. 210–220 (2004)
18. Kelley, J.F.: An empirical methodology for writing user-friendly natural language computer applications. In: Proc. of ACM SIG-CHI 1983 Human Factors in Computing Systems, Boston, pp. 193–196. ACM, New York (1983)
19. Looije, R., te Brake, G., Neerincx, M.: Usability engineering for mobile maps. In: Proc. of Mobility 2007, pp. 532–539. ACM, New York (2007)
20. Mayhew, D.: The Usability Engineering Lifecylce: A Practitioner's Handbook for User Interface Design, 1st edn. Interactive Technologies. Morgan Kaufmann (1999)
21. Perry, M., O'hara, K., Sellen, A., Brown, B., Harper, R.: Dealing with mobility: understanding access anytime, anywhere. ACM Transactions on Computer-Human Interaction 8(4), 323–347 (2001)
22. Yarmosh, K.: App Savvy: Turning Ideas into iPad and iPhone Apps Customers Really Want. O'Reilly Media, Inc. (2010)
23. Yuan, Y., Zheng, W.: From Stationary Work Support to Mobile Work Support: A Theoretical Framework. In: Proc. of the International Conference on Mobile Business (ICMB 2005), pp. 315–321. IEEE Computer Society, Washington, DC (2005)
24. Wassermann, A.: Software engineering issues for mobile application development. In: Proc. FoSER 2010, pp. 397–400. ACM Press (2010)