Software Qual J (2015) 23:509–536
DOI 10.1007/s11219-014-9239-1

The fine art of user-centered software development

Bernhard Peischl · Michaela Ferk · Andreas Holzinger

Published online: 27 May 2014

Abstract In this article, we report on the user-centered development of a mobile medical app under limited resources. We discuss (non-functional) quality attributes that we used to choose the platform for development of the medical app. As the major contribution, we show how to integrate user-centered design in an early stage of mobile app development under the presence of limited resources. Moreover, we present empirical results gained from our two-stage testing procedure including recommendations to provide both a useful and useable business app.

Keywords User-centered design · Software engineering process · Usability · Mobile software quality · Mobile usability

1 Motivation

Many studies have reported the benefits of a strong commitment to usability in the software development life cycle, from the early classic work of Norman and Darper (1986), Mayhew (1991), Karat (1997), Vredenburg et al. (2001), Holzinger (2003) to most recent research (Flood et al. 2012; Harrison et al. 2013; Iacob et al. 2014) to mention only a few. Usability is increasingly recognized as an important quality factor for interactive software

B. Peischl (✉)
Void, Softnet Austria, Inffeldgasse 16b/II, 8010 Graz, Austria
e-mail: bernhard.peischl@soft-net.at

M. Ferk
FERK Systems, Raiffeisenstrasse 46, 8010 Graz, Austria
e-mail: michaela.ferk@ferk-systems.com

A. Holzinger
Medical University Graz, Auenbruggerplatz 2/V, 8036 Graz, Austria
e-mail: andreas.holzinger@medunigraz.at

systems including software running on mobile devices (Holzinger et al. 2008; Seffah et al. 2006). Very recent research (Harrison et al. 2013) in particular addresses the limitations of existing usability models when applied to mobile devices. Among the observable benefits of usable user interfaces, one can mention human productivity and performance, safety, and commercial viability. Usability is important, not only to increase the speed and accuracy of the range of tasks carried out by the users of a system, but also to ensure the safety of the user (e.g., repetitive strain injury in medicine). Usability is also imperative where software is used to control critical processes as for example in medicine and healthcare (Holzinger et al. 2012a; Thimbleby 2007). Attitudes can be influenced by abstract factors such as the look and feel of a product, and how the software can be customized by the user (e.g., colors, fonts, and commands).

However, usability is only a part of the whole. There are a lot of other aspects to consider when speaking about quality of mobile apps. This includes quality-related aspects in terms of non-functional attributes. Further, the usefulness of an app needs to be taken into account from the very beginning.

This article is organized as follows. In Sect. 2, we discuss the quality of mobile medical apps. Section 3 analyses and compares mobile platforms with a particular focus on the mobile Health (mHealth) market. Our decision to focus primarily on the Android platform (and in a second step on the iOS platform) is motivated in terms of assessing seven criteria: platform openness, means of accessing hardware, integration of third-party apps, reliability and offline functionality, multithreading, mobile device management, as well as security and privacy.

Business apps in general have high requirements with respect to reliability and security. In our case, in particular, the interaction between mobile clients and back-end server needs to fulfill the highest standards with respect to reliability and security. Therefore, in Sect. 4, we briefly discuss the interaction with the back-end server and the specific requirements of our app with respect to security and privacy.

Afterward we turn to the major contribution of this article showing how to integrate user-centered design into mobile medical app development. Section 5 outlines the procedure of integrating user-centered design into the mobile software development life cycle from the very beginning. In Sect. 6, we report on the qualitative and quantitative results obtained from our two-staged usability test. Section 7 summarizes our recommendations and Sect. 8 reviews related work afterward Sect. 9 concludes this article.

2 Mobile medical apps in healthcare

Mobile computing in general has followed in the footsteps of the Web, and quality assurance of mobile apps has followed quality assurance of Web applications. For example, requirements review, software test planning, test strategy, test case creation, test case execution, test results reporting, software defect management, and recording of software quality metrics follow a nearly identical process flow. Although these two areas share many drivers, there are some aspects of mobile quality that are distinctly more challenging and require particular attention (Flood et al. 2013; Garcia et al. 2011). Platform fragmentation, the physical characteristics of the mobile device, user experience, integration of third-party apps, performance, security, system integration, and the deployment of mobile apps are pressing concerns that need to be addressed.

Whereas the aspects mentioned previously hold for mobile apps in general, business apps for professional usage pose more specific challenges. Particularly due to the higher quality requirements, the development of mobile apps for professional usage in the healthcare field represents a challenge for mobile software development and quality assurance.

Some of the benefits that mobile computing offers are essential for the healthcare industry as the major part of nurses' and doctors' work require them to interact with patients. Their work style cannot be changed so that they work from a desk using a stationary computer. IT systems in health care that are located at a fixed terminal are inefficient because the required information is not available at the time of query and this interrupts the workflow. Healthcare providers would like to enter information into the patient record and process information at the point of care to save time and minimize input errors.

The inherent properties of mobile devices are the root cause of their security issues. As the term "mobile devices" indicates, they are portable. Due to the form factor, they have limited resources. Unfortunately, the current practice when addressing resource limitations is to ignore well-known security concepts. For instance, to empower WML (Wireless Markup Language) scripts, implementations lack the established sandbox model; thus, downloaded scripts can access all local resources without restriction (Greamo and Ghosh 2011).

Since mobile devices are portable, they are used at different places and within different environments. Thus, mobile devices are used to connect to various networks and some of these networks might not be trustworthy (e.g., not offering state of the practice encryption or hiding the legal owner of the network). In addition, mobile devices are usually connected to wireless networks that are often easier to compromise than their wired counterparts. Moreover, the devices are usually personal devices ("personal digital assistant", PDA), and therefore, people store a lot of personal information on them even though the operating systems usually offer little protection for the privacy of data. Together these issues impose requirements with respect to secure and trustworthy interoperability (Holzinger et al. 2012b).

Their portability clearly makes mobile devices subject to loss or theft. Once a mobile device has been stolen or lost, the thief or other unauthorized individuals are likely to gain direct access to the data stored on the device (Weippl et al. 2006).

2.1 A useful app for acquisition of a patient's health status

Our goal was to develop a software system, including an application running on mobile devices that supports the acquisition and documentation of the health state of a patient. For this purpose, we employ the ICF and the ICD standards (WHO 2010b, c). Internationally established classifications including ICF and ICD facilitate the storage, retrieval, analysis, and interpretation of health-related data. For example, this allows one to compare data within populations over time as well as the compilation of nationally or regionally consistent data.

ICF is a comprehensive classification of the health state of a person. "Comprehensive" in this sense means that the health state of a person is not exclusively assessed by relying on physical functions or disorders but also by taking into account the possibility of an active lifestyle and factors regarding the environment of the patient. The ICF consists of several components (a hierarchical structure grouped into physical functions, activities, environmental factors, and factors regarding the patient itself), where each component is assigned a set of ICF codes. The assessment for each ICF code is done with the help of a five-valued scale, where the value increases with increasing limitation of the patient.

Details regarding the assessment are explained in the ICF application and training tools (WHO 2010a).

The ICD is a classification system for medical diagnoses. Within the ICD, each medical diagnosis is assigned to a three or four digit code. ICD also supports hierarchies in terms of chapters, groups, and categories. In general, the relevance of ICF codes for a specific patient primarily depends on the patient's ICD diagnoses. Therefore, ICF-coresets have been developed. ICF-coresets contain ICF codes that are of particular relevance for groups of ICD diagnoses.

2.2 Project setting and challenges

In the course of a collaborative research project with an Austrian SME developing software solutions for the medical domain, we identified the following pivotal issues regarding the quality of mobile apps.

First, there is the question of the platform being supported (see Sect. 3). As the current landscape for mobile operating systems is rather heterogeneous, we needed to trade off advantages and drawbacks particularly for the healthcare sector.

Second, and in contrast to the mainstream consumer market, mobile business applications must fulfill business-critical requirements like a certain degree of quality of service and requirements with respect to privacy. Moreover, the communication of the mobile device and the data synchronization mechanism must be secure, robust, and reliable independent from place, time, or usage context. Within a heterogeneous environment, these issues are related to the Web service architecture (see Sect. 4).

Third, as the application being developed is highly interactive, the acceptance among the staff members is of uttermost importance. Unclear user requirements pose a further challenge in developing a user-centered and useful application. User-centricity and usefulness become major quality attributes as the exposure to occupational stress in healthcare organisations typically is rather high (see Sects. 5–7).

3 Analysis and comparison of mobile platforms

In deciding upon the platform for of our app, besides of the technological aspects, we particularly considered the main trends within today's healthcare sector in order to reach a high degree of acceptance among the users from the very beginning. In healthcare, the staff itself and the vocational training of the staff are rather diverse. Various user groups have to feel comfortable with the mobile device and the platform.

3.1 Market trends in mHealth

According to a survey (MH 2012) considering the US market, medical doctors prefer Apple's iPhone followed by RIM's Blackberry and devices powered by Google's Android OS. As the main reason for high proliferation of the iPhone, medical doctors mention the high degree of usability (Dolan 2011; Turisco and Garzone 2013). According to of Liu et al. (2011), there are far more healthcare applications in Apple's App Store than on the markets for other platforms. Particularly in the recent past, the healthcare market appears to be increasingly focused on iOS. However, this is quickly changing, most recently, Apple has lost market shares and Android is catching up.

In summer 2010, a study among healthcare institutions (R2G 2010) anticipated that Android will overtake the current market leader iOS by 2015 in terms of market penetration in the healthcare sector. Currently, Android and iOS are the prevailing platforms for mobile healthcare applications and Android has better prospects for the near future.

The relevance of Android becomes even bigger considering the fact that the mentioned survey (MH 2012) exclusively focuses on the (1) US healthcare market and (2) that this survey focuses exclusively on medical doctors. On a global scale, without limiting the view on medical doctors only, the market share of iOS is already considerably smaller than that of Android devices. In the second quarter of 2012, statistics from the International Data Corporation (2012) highlighted that the market share of iOS (with a share of 16.9 per cent worldwide) is considerably behind that of Android (amounting to a market share of 61.1 %).

Furthermore, well-off medical doctors use the premium products from Apple more likely than therapists or nursing staff. For our application, medical doctors, therapists, and nursing staff are potential users whereby therapists and general nursing staff will play a major role. This makes the Android platform even more interesting.

Apart from the dominance of iOS and the Android platform, the Global mHealth Developer Survey (R2G 2010) revealed that HTML5 will become the technology of choice in the healthcare market in the near future. With this technology, an application needs to be developed only once and supports all platforms that can run within an HTML5 capable browser. Moreover, there is no need for installing an app and thus no need for updating the app on the device.

3.2 Technological characteristics

Each of the platforms, respectively, technologies mentioned above—Android, iOS, and HTML5—have certain advantages and drawbacks with respect to the technological characteristics relevant for our application. We briefly discuss the characteristics that we have identified prior to design and development (Ferk 2013).

3.2.1 Openness of the platform

For development and distribution of our app, the platform should not impose too stringent restrictions (e.g., conformance to design rules). In this respect, Google's Android turned out to pursue a rather open approach: There is a proposal for design rules, however, these rules are not binding and are not enforced. Distribution of the app is possible without using Google's market place (Google Playstore), e.g., e-Mail or a Webserver could be used for deployment as well.

In distributing apps, Apple is rather restrictive as the distribution must be done via Apple's appstore. Prior to releasing an app for download, the fulfillment of certain guidelines is enforced. Details on the checks being conducted are rather fuzzy and Apple moreover can refuse acceptance of an app. For enterprises, there is an alternative distribution channel (iOS Enterprise Program) that allows an enterprise to deploy apps to the devices of employees. Deployment can thus be performed via a simple Webserver or a mobile device management solution.

HTML5 Webapps run within the browser of the mobile device. Thus, ideally, there are no restrictions with respect to platform openness. However, HTML5 widely differs in what is implemented on which platform, and thus, the "write-once, run anywhere" approach is not fully covered considering the current state of practice.

3.2.2 Integration of hardware/other apps

Mobile apps provide value to the users by smoothly integrating device-specific hardware, e.g., the camera. Furthermore, the means of communication with other apps are an important issue to support the various integration scenarios in the medical field (e.g., apps for a barcode scanner for medications).

Both platforms, Android and iOS, support the integration of specific hardware components and third-party applications. Whereas Android offers an access scheme for hardware, iOS allows one to access every hardware component with the exception of notifications and location services. Integration of third-party applications is supported on both platforms.

HTML5 allows one to access hardware components as long as the browser on the mobile device supports the Media Capture API that supports, e.g., access to camera and audio. However, at time of carrying out this study, Bluetooth interfaces have not been supported in terms of an API. This has been a severe restriction for our application, as external medical devices often use Bluetooth. Furthermore, the communication with other applications on the device has not been possible, which further restricts the scope of applicability. However, at time of writing, this article the W3C systems applications working group provides APIs for interacting with the life cycle of apps and a low-level Bluetooth API (Raggett 2014).

3.2.3 Offline functionality

In order to ensure that users can continue their work in the case of network outages or low bandwidth, our app needs to provide critical functionalities offline as well. Therefore, data need to be stored on the device. In this respect, iOS and Android can make use of the lightweight database SQLite. In order to provide these functionalities, Web applications need to have permanent Internet access. Furthermore, the HTML5 standard supports offline Webapps. Therefore, HTML5 allows one to define resources that need to be available offline (Pilgrim 2011) and enables local and persistent data storage within the Web browser.

3.2.4 Multithreading

Our application needs to synchronize the local data with the back-end server. Therefore, multithreading needs to be supported by the platform. Without this ability, the user interface of our app might be blocked during synchronization of the data. Modern operating systems such as Android or iOS support multithreading, however, up to now, Web applications did not provide a practical solution to this problem. In this respect, HTML5 introduces the concept of "Webworkers" which operate independently from the thread of the user interface (Rousset 2013). Thus, HTML5 applications fulfill the requirements with respect to multithreading.

3.2.5 Mobile device management

For integration of mobile devices and apps into the business processes, configuration and monitoring via a mobile device management are indispensable. Neither Android nor iOS offers native solutions for mobile device management. However, there are third-party applications over which, for example, the usage of a password or locking can be enforced. HTML5 does not support the configuration of mobile devices at all.

3.2.6 Security aspects

Healthcare applications process and store sensible and personal data. Therefore, the underlying platform has to support foundational security and privacy mechanisms to protect the data.

On the iOS platform, secure storage of passwords and private keys is done via keychains. Secure storage of authentication data is thus supported. On the Android platform, private keys can be stored in a secure way via the credential storage. Username and password are solely stored within the internal memory of an application; thus, on non-rooted devices, these data are only visible to the application itself. Further, iOS enforces a device encryption and Android versions after version 3.0 support the encryption of the file system (Apple 2012; Google 2012a, b). The HTML5 standard does not provide encryption of data being stored locally.

3.3 Platform decision

Table 1 outlines a simplified overview of the platforms and requirements with respect to the technological characteristics discussed previously. The iOS and the Android platform fulfill most of the requirements for our application. As both platforms are relevant for the healthcare market, it might be necessary to support iOS and Android. However, this decision finally has to be aligned with the overall business strategy of our cooperation partner. For the given application, HTML5 is not a serious option due to limited means for integration of third-party applications and the low level of security mechanisms being provided. As mentioned in Sect. 2, means of integration of third-party apps and security require particular attention when it comes to quality assurance of mobile apps. Moreover, the authors of Bloice et al. (2011) recommend developing a native application, particularly when hardware access or a notification mechanism is required. As the market trends indicate the leading position of the Android platform and the mHealth study (R2G 2010) anticipates a higher impact of the Android platform for the future, we decided to focus the work presented herein on the Android platform. However, in a later stage of product development, our collaboration partner intends to support the iOS platform as well.

Table 1 Summary on platforms and desired properties: fully supported (*green/dark grey*), partially supported (*yellow/light grey*), and not supported (*red/black*) (Color table online)

	iOS	**Android**	**HTML5**
Platform openness			
Hardware access			
Integr. 3rd party apps			
Offline functionality			
Multithreading			
Mobile device mgmt.			
Security and privacy			

4 Service architecture

A key factor of any mobile business application is the interaction between the mobile app and the back-end server. In contrast to apps for the consumer market, mobile business applications impose stringent requirements with respect to security and privacy of the data being exchanged and the quality of service in interacting with the back-end server (Ferk 2013).

4.1 Interaction with the back-end server

The core functionality of our app must be independent from the availability of wireless networks. The development of such an occasionally connected application is more complex than that of an application that relies on an available network connection. For example, if no network connection is available, data must be stored locally and synchronized with the server when a reliable connection becomes available. For example, mobile healthcare workers in rural areas cannot rely on a reliable broadband connection. In such cases, it is important to provide the core of the functionality although there is no reliable network connection in place. Further, local storage of the acquired data reduces the data traffic and the risk of loss of data and counteracts a blocking user interface. Furthermore, mobile business applications are accompanied from various requirements with respect to communication and data synchronization with the back-end server. For example, the delivery of messages has to be guaranteed and performing transactions must be possible to ensure the consistency of data. The fact that we store data locally to enable the usage of the app even without having access to the Internet has a severe impact on the software architecture. To provide a high degree of reliability for our business-critical application, we decided to rely on a software architecture supporting the paradigm of "occasionally connected computing," "Occasionally connected computing" is a software architecture based on the idea that an end user should be able to continue working with an Internet application even when temporarily disconnected or when a wireless connection fails or is otherwise unavailable (Bahrami et al. 2006).

4.2 Security and privacy

Mobile business apps often process sensitive data, and thus, high security requirements to protect the data from unauthorized access are in place. In this respect, healthcare applications are liable to particular European guidelines and national laws thereon, e.g., Data Protection Act (in Austria: Datenschutzgesetz DSG) and law on healthcare telematics (in Austria: Gesundheitstelematikgesetz GTelG). Requirements with respect to security are mostly non-functional—and alongside with other non-functional quality attributes—suffer from fuzzy specification. Irrespective of this, security is a quality attribute that is gaining more and more importance, particularly if sensitive and personal data are subject of transmission. To protect sensitive data from unauthorized access, appropriate authorization mechanisms have to be in place. This ranges from simple authorization via the Hypertext Transfer Protocol (HTTP) to the usage of qualified digital signatures.

Furthermore, stored and transmitted data must be encrypted. For example, when transmitting data, one can make use of encryption mechanisms like Transport Layer Security (TLS) or similar techniques. To reliably ensure the integrity of the transmitted data, an electronic signature can be used. Alternatively, Message Authentication Codes (MACs) can be employed.

5 Early use of user-centered design

Today, it is well known that usability engineering methods have a huge impact on the perceived quality of a software. However, in practice, high costs and stringent time planning prevent the use of these well-known methods (Boivie et al. 2003; Cooke and Mings 2005; Larusdottir 2011). In order to address user-centered design in spite of limited resources, we propose to integrate usability evaluation in a very early stage of software design by using paper prototypes. Particularly within a fast-paced and innovation-driven setting, under presence of rather fuzzy requirements, the early feedback of potential users may considerably improve the quality of the obtained design and product. Due to this, we prohibit development efforts that do not match market and user requirements. Faily and Lyle (2013) point out that the increased interest in the personification of archetypical users has extended to the software engineering communities in the recent past.

The core element of integrating usability engineering techniques into the software development life cycle is the usage of interaction design in combination with paper prototypes to conduct early usability tests. Rubin and Chisnell (2008) emphasize that the usage of usability engineering techniques is appropriate as soon as obvious usability flaws have been removed. Due to this, prior to conducting usability tests, our company partner carried out several walk-throughs. Further, the initial design followed the usability guidelines for mobile applications according to Inostroza et al. (2012) as well as the design guidelines for Android and iOS.

According to Rubin and Chisnell (2008), we conducted an exploratory and formative part for usability testing. The exploratory part of our usability tests aims to clarify whether or not the assumed user profiles match the reality and whether our app provides assistance in the daily work. We emphasized this aspect by conducting interviews questioning the state of practice in collecting health-related data. The formative part of our usability tests aims to verify whether our design supports the user in an optimal way.

5.1 Research questions

As we relate usability tests with interviews about the common working practices, we decided to classify the research questions according to this. The interviews being performed intend to strengthen the exploratory component in our evaluation. In the course of the interviews, we reveal typical working procedures in collecting health-related data in order to acknowledge the assumed user profiles and usage context. Our survey among the potential users of the app intends to answer the following research questions:

- What are typical working procedures regarding documentation of the health state of a patient in care and therapy?
- To what extent are these procedures standardized? For example, are there standardized questionnaires in place and are these used on a regular basis?
- Is the term ICF well known, respectively, is the ICF used in practice?
- To what extent are medical doctors involved in the documentation of a patient's health state?

After having conducted interviews in a very early stage of development (stage 1), we conducted the formative component of the evaluation (stage 2) to evaluate the quality of the user interface design. This part of the evaluation strived to answer the following questions:

- Does the user benefit from using ICF with the planned user interface, that is, does the ICF app provide assistance in the working procedures?
- Is all the relevant information processed?
- Is the overall usability and navigation within the app intuitive?
- Which kind of previous knowledge, respectively, which kind of training is required in order to use the app for the daily work?

This very early evaluation of the usability in the software life cycle mainly serves the purpose of providing feedback regarding the suitability of the user interface design. Taking into account the lack of resources and the current practice in software engineering, this can be done even with a small amount of test participants. It is well known that the number of test participants has an economic as well as a qualitative impact. The more test participants are being considered, the higher the costs are and the more relevant the results of the tests are. Thus, the main challenge in the given context is to reveal as many usability problems as possible with a small number of test participants.

5.2 On the optimal number of participants

Nielsen and Landauer (1993) show that a sample composed of a range of between three and five subjects with appropriate skills is generally enough to assess interactions with a Web interface and to identify 80 % of the problems. Borsci et al. (2013) most recently investigated this controversial assumption and report that it may be a good decision to adopt the five-user assumption but also illustrate where a more nuanced judgment around the minimal number of test participants is required. The decision process outlined in Borsci et al. (2013) considers three main constraints:

- The cost of error identification against the available budget
- The kind of product and the level of safety required for optimal interaction
- The external issues that may require evaluation with more than five users

When none of the above-mentioned constraints affect the evaluation, or when the cost of error identification is considered to be an important issue (as in our case, where only a low evaluation budget is available), the decision process outlined in Borsci et al. suggests to test the product under consideration with only five users or (as in our case) with two or three different groups composed of at least three users of each kind.

According to Bastien (2010), the optimal number of test participants is discussed since the nineties. At that time, about four to five test participants were considered adequate to reveal 80–85 per cent of all usability problems. In the meanwhile, more recent studies recommend a considerably higher number of test participants (Bastien 2010).

The number of persons is further determined by the complexity of the tasks scheduled for the usability test and is still an open research issue at the moment. Up to now, many successful research works with practical relevance (Holzinger and Errath 2007; Horsky et al. 2010) follow the recommendation from the nineties (1994; Short et al. 1991) to conduct usability tests with four to five persons. According to Rubin (Rubin and Chisnell 2008), this number is sufficient to reveal the majority of usability problems.

Furthermore, Rubin and Chisnell (2008) note that the number of test participants can be smaller, if the tests are carried out in an iterative manner. Usability tests that consider the Thinking-Aloud method can be conducted in an adequate manner even with three persons per user group (Holzinger 2005; Holzinger and Brown 2008; Holzinger and Slany 2006).

Table 2 Demographics of the test participants

No. of participants	Age (from/to)
5	18–25
2	26–35
1	36–45

Table 3 Areas and specific topics covered by our questionnaire

Questionnaire items	Covered topics
General issues and demographics	Name, education, and age of participant
Technical background—utilization of desktop computer	Amount of daily computer usage, preferred operating systems, purpose of desktop computer usage in the private and professional field
Technical background—utilization of smartphone	Amount of daily smartphone usage, preferred operating systems, purpose of smartphone usage in the private and professional field
Expertise in the healthcare sector	Education and vocational training in the healthcare field, relevant professional experience in the (a) medicine (b) therapy and (c) care
Familiarity with the terms "International Classification of Diseases" (ICD) and "International Classification of Functioning, Disability and Health" (ICF)	Self-assessment given the categories, "well-known", "roughly known", "not-known"

As we used the Thinking-Aloud method, supported the test in terms of interviews and performed these tests in an iterative way, we considered three persons per user group as lower bound for obtaining meaningful results. This holds for the relevant user groups of nursing staff and therapists. Medical doctors are considered as a fringe group, as they turned out not to be the main users of our application. In summary, we decided to set up an experiment with eight participants: three nurses, three physiotherapists, and two doctors. Table 2 provides an overview on the demographics of our test participants.

5.3 Design of the experiment and test procedure

In order to gain insights into the typical working practices and the usage of ICF/ICD classification, we prepared a questionnaire consisting of various sections. With this questionnaire, we collected data regarding the demographics of our test participants, their technical background, the degree of smartphone usage, the education and expertise in the healthcare field, and the knowledge of ICD and ICF. Table 3 summarizes the relevant areas covered and the most relevant data that we surveyed using this questionnaire.

After conducting the interviews regarding the common practice of collecting health-relevant data, we explained the basic idea of our application and the concept of ICF. Table 4 gives some examples for interview questions regarding the state of practice in using the ICF. This was followed by a demonstration of paper prototype tests and an introduction to the Thinking-Aloud method.

Table 4 Examples of interview questions for capturing the state of practice in ICF usage

No.	Interview questions (state of the practice in using ICF)
1	Describe how you have already come into contact with ICF?
2	Do you find the use of the ICF codes to document the health status of patients to be practical and suitable for everyday use? If not, why not? Do you have suggestions for improvements? Do you see deficiencies?
3	Do you think there is need for retrieving all the information about the health status of a patient on the smart phone?
4	How is the health status of a patient currently captured and documented? How is the exact workflow when documenting a patient's health state? Do you see advantages or disadvantages?
5	Describe if you have come in contact with mobile applications in your professional work and how this happened?

Table 5 Examples of interview questions for the qualitative survey

No.	Interview question
1	What is your overall impression of the prototype?
2	In general, do you feel that the user interface is intuitive? Where did you experience problems and why did you encounter these problems?
3	Is there too much or too less information on the screen?
4	Is the navigation well-structured?
5	Is the screen-layout clearly arranged?
6	Is the font readable without effort?
7	Do you feel the selection of colors is adequate?
8	Were the icon graphics self-explanatory and supportive? Which symbols have been misleading?

For the usability tests, we selected problems that reflect the most common working procedures in the daily business. The sequence of the individual tasks was chosen to represent the typical sequence of tasks in daily business ordered by increasing difficulty. After conducting the usability tests, the critical issues observed during the test have been discussed with the participants. For this purpose, we also prepared a questionnaire (see Table 5). Taking this into account, we provide recommendations for improvements summarized in Sect. 7.

In the following, we briefly summarize the individual tasks that had to be carried out.

5.3.1 Task 1

This task is designed to remember the concepts of ICF and ICD in order to establish optimal prerequisites for the test. Participants were given the design of a patient sheet (see Fig. 1). Based on this design, we asked the participants to describe the supposed functionalities behind the individual ICF- and ICD-related buttons.

Task-end criteria: The participants managed to correctly assign the terms disease pattern ("Krankheitsbild" in German, Fig. 1), diagnosis ("Diagnose" in German), ICF-acquisition ("ICF-Erfassung" in German, Fig. 1) and ICF Evaluation ("ICF-Auswertung" in German, Fig. 1) to the corresponding app functionalities.

Fig. 1 Patient sheet

5.3.2 Task 2

In the second task, we asked the test participants to associate a clinical disease pattern and an ICD diagnoses to a patient. Although this task is not a typical working procedure for our test participants, it fosters the understanding of the relationship between ICD diagnoses and the clinical disease pattern. The term clinical disease pattern was subject of briefing of the participants.

Task-end criteria: The relevant diagnosis as well as the clinical disease pattern has been assigned to the patient.

5.3.3 Task 3

Documenting the health status with ICF. This task deals with the core of our application, the recording of health-relevant data based on ICF codes. The participants were asked to evaluate the code "d450 Gehen". For this purpose, we provided further information (diagnostic findings of nursing staff etc.).

Task-end criteria: Detailed information regarding the values for evaluation has been retrieved by the participants. The relevant code "d450 Gehen" has been evaluated with the values 1, respectively, 2.

5.3.4 Task 4

Analyzing the health status based on ICF data. The final task considered the visualization and analysis of the ICF recording. First, the participants were asked to retrieve those ICF codes that have improved in the recent past (Task 4.1). Second, we asked the participants to retrieve details about the evolution of the ICF code "d450 Gehen" (Task 4.2).

Task-end criteria: This criteria consists of two parts. First, the participant had to correctly mention the code, which exhibits a positive trend in the course of the last three data acquisition points. Second, the participants had to view the detailed diagram for the code "d450 Gehen" over a time period of 6 months.

6 Evaluation

In this section, we report on the results of the conducted usability tests. Before we start to discuss the results for the different tasks, we summarize the results from the interviews that we conducted prior to carrying out the usability tests.

6.1 Results from the interviews

6.1.1 Practice in healthcare documentation

The majority of the test participants considered the typical working procedures regarding the documentation of the health state as tedious and intensive writing work. In outpatient care and in therapy, the test participants use questionnaires and standardized scales (such as the Braden-, Barthel- or Morse-Fall-scale). The standardized tools being used capture a partial aspect of the health status of a patient. Additional information that is relevant to characterize the health state is captured by using free text notes. Questionnaires are filled in regularly although the period of filling out the questionnaires varies from patient to patient. Besides of these questionnaires and the mentioned scales, the majority of documentation in medical care is done using free (i.e., no templates are used) text notes.

6.1.2 Scope of activities for the different user groups

Therapists and nurses consider ICD diagnoses provided by medical doctors as a logical starting point for their professional work. However, during the interview, the medical doctors emphasized that the questionnaires from physiotherapists or nursing staff are in general not used by them. Therefore, the important user groups for our app consist of therapists and nurses.

6.1.3 ICF in practice

Our test participants did not use the ICF classification in practice. However, the ICF classification scheme plays an important role in vocational training and education. This particularly holds for the group of physiotherapists.

6.1.4 Mobile technologies in healthcare

At the time of conducting this study, mobile apps have not been used at the hospitals under consideration. In one of the hospitals ward, rounds are carried out using a laptop computer on a creeper.

Table 6 Comparison of registered nurses/care assistants and physiotherapists

Task	Registered nurses		Physiotherapists	
	with assist. (%)	wo. assist. (%)	with assist. (%)	wo. assist. (%)
1	50	0	75	0
2	75	25	100	0
3	25	25	75	0
4.1	0	75	0	100
4.2	100	0	100	0

Table 7 Comparison of smartphone users and unexperienced users

Task	Smartphone users		Unexperienced users	
	with assist. (%)	wo. assist. (%)	with assist. (%)	wo. assist. (%)
1	75	0	0	0
2	100	0	67	33
3	20	60	0	0
4.1	0	100	0	67
4.2	100	0	67	33

6.2 Results from the usability tests

We start with a discussion from a quantitative point of view; however, the focus of our experiments is on the qualitative evaluation. On average, six out of eight participants were able to fulfill the given task. Table 6 outlines a comparison of the results between the major user groups, physiotherapists, and nurses with and without assistance. Table 7 groups the test participants, into smartphone users and those who have no experience with smartphones. This specific grouping is a result of the answers to the general questionnaire outlined in Table 2.

Notably, in our experiments, physiotherapists achieved considerably better results than nurses. The same observation holds for smartphone users, who consistently gained better results than unexperienced users. However, one has to bear in mind that 100 % of the physiotherapists use a smartphone, whereas only 50 % of the nurses claim to be experienced with smartphones. This relationship needs to be kept in mind in interpreting the results from Tables 6 and 7.

Besides the quantitative results, we obtained qualitative feedback from Task 1 to Task 4. In the following, we discuss these results in detail.

6.2.1 Task 1

Only 50 % (physiotherapists 75 %) of the nursing staff was able to associate all three functions with the corresponding working task. The reason for this is the term disease pattern (in German: “Krankheitsbild”). Some of our test users were not able to associate a functionality of the app with this specific term. The term refers to a particular set of ICF codes in the context of a medical diagnoses or a group of diagnoses. However, most of our test participants associated the term disease pattern with the term diagnosis. During the preparation for the interviews, the majority of our test participants clearly stated that the term disease pattern is not adequate. However, our test participants were not able to come up with a proposal for designating the specific functionality. Most of the test participants found that ICF diagnoses group (in German: "ICF Diagnosegruppen") is more suitable for designating this kind of functionality.

6.2.2 Tasks 2

The UI designs for acquisition of diagnoses and ICF diagnoses groups are sketched in Figs. 2 and 3. All test participants were able to complete this task successfully. At the very beginning of the design process, the relationship between acquisition of ICD diagnoses and ICF-coresets in everyday practice has been an open issue. Our interviews revealed that nursing staff and physiotherapists partially carry out the documentation of a patient's health status by using medical diagnoses. However, in general, they do not fully rely on this. Therefore, the acquisition of ICF-coresets together with medical diagnoses appears to be suitable. The good results obtained from this task emphasize that the user interface design is able to convey the relationship between ICD and ICF in an intuitive way.

6.2.3 Task 3

Figure 4 shows the dialog for acquisition of ICF documentation. The rather low success rate for this task is rooted in the fact that around half of our test participants did not pay attention to supplemental information regarding the ICF codes. Retrieving this supplemental information has been considered as being a part of the task-end criterion. Without

Fig. 2 Acquisition of ICD diagnosis

Fig. 3 Cross reference to acquisition of ICF disease pattern

this criterion, the success rate of this task would have been 100 %. Figure 5 shows the UI displaying the supplemental information for evaluating ICF codes. Each value is associated with specific criteria that the patient has to fulfill. The supplemental information is retrieved via the central buttons showing the relevant numbers. In summary, we were able to identify four different reasons for disregarding this information:

1. *Meaning of the values for ICF evaluation*: Several of our test participants did not recognize the meaning of the value displayed on the button. In Fig. 4, this value for the code "d450, Gehen" is rendered as the green number 1 twice. Although we briefed the participants about the five-valued scale for all ICF codes, the test participants could not associate the value on the buttons with the value for the ICF codes.
2. *Relationship between values and buttons*: Several test participants did not recognize the relationship between the buttons and the values being displayed. Although our test participants understood that the red/yellow/green buttons denote decreasing/constant/increasing health state compared with the previous evaluation, they did not recognize that hitting one of these buttons has an effect on the value being displayed. We knew this problem from a previous walk-through cycle, where we had positioned the value on the left-hand side. As a consequence of the previous walk-through cycle, we decided to position the value in the center. The majority of our test participants had preference to position this button on the left-hand side.
3. *Double meaning of colors*: This task further revealed a weakness of our color concept. Several test participants were unsure about the (double) meaning of our color concept. On the one hand, we use the colors red, yellow, and green for the assessment buttons to depict degradation of the health state, stable health state, and improvement of the state.

Fig. 4 Dialog for acquisition of the ICF evaluation

On the other hand, we associated colors to the ICF values. However, these colors do not denote the evolvement of the health state but rather a static assessment. For example, values 0 and 1 are displayed in green as these values indicate a small limitation of the patient, whereas the values 3 and 4 are displayed in red correspondingly indicating a severe limitation of the patient.

4. *Discovering information regarding the ICF values*: Our test participants had problems to retrieve the supplemental information regarding the ICF values. Often the information for the ICF codes itself or actions and goal planning (Fig. 4) have been retrieved.

6.2.4 Task 4

This task deals with the assessments of ICF codes. Notably, the learning curve strongly increased (see Tables 2, 3) when moving from Task 4.1 to Task 4.2. While none of our test participants managed to solve Task 4.1 without assistance, 87.5 per cent of our test users were able to perform Task 4.2. This indicates that our user interface enables simple guidance of the user through the menu and our test users experienced some learning effect.

6.2.5 Task 4.1

Figure 6 shows the alternation of certain ICF codes over a certain period of time. In this figure, the current ICF assessment is compared to an old assessment that was carried out

Fig. 5 Supplemental information for code “d450 Gehen”

several months ago. Our test participants have not been able to correctly interpret this alternation. The following issues have been discovered due to this test.

6.3 Information visualization

Our test users did not perceive that the values being displayed refer to a time period of 100 acquisitions in between. Intuitively, our test users assumed that the current assessment is compared with the previous assessment. Our users did not pay attention to the information that is indicating the longer time period for acquisition (“100 Erfassungen”). Further, similar to Task 3, the double meaning of colors was mentioned as a problem.

6.4 User expectations

Another problem regarding this task is the menu outlined in Fig. 7. This menu offers options to switch between various views for evaluation and allows the configuration of these views. After the options are set, one hits the back button and returns to the previous view. Most users looked for a button to acknowledge the options being set and only a minority used the back button prior to searching for a button to acknowledge the new settings.

6.4.1 Task 4.2

As mentioned previously, the learning effect from Task 4.1 was vast. Almost no one of our test users had problems in navigating through the menu. Figure 8 outlines the detailed view

Fig. 6 List view on the ICF trend

for the ICF code "d450 Gehen" which lots of our test participants managed to retrieve without problems.

7 Discussion

In the following, we summarize the specific recommendations that follow from the evaluation:

7.1 Recommendation Task 1

The term "Krankheitsbild" is not well known and not suitable to designate the specific functionality. A unique term should be used, so that even staff with low knowledge of ICF can associate the appropriate functionality with this term. This term should be used consistently in trainings and in the everyday working practice.

7.2 Recommendations Task 3

The recommendations resulting from this task can be split into four different aspects.

1. *Meaning of the values for ICF evaluation*: Basically, there are two ways to solve this problem. One way is to conduct vocational training on the ICF. Once the users have a clear understanding of the ICF classification scheme, associating the numbers on the

Fig. 7 Menu for the ICF evaluation

buttons with the values for the ICF code should not pose a severe problem. On the other hand, we could hide the evaluation of an ICF code with a specific number and instead of that introduce a color scheme so that the used color uniquely characterizes the ICF evaluation.

2. *Relationship between values and buttons*: The value should again be placed to the left of the button. Further, a short flashing of the value when hitting a button could be used to emphasize the relationship between hitting a button and the change of the value.
3. *Double meaning of colors*: To solve this issue, set aside the coloring for the ICF values. The colors green, yellow, and red should solely be used to indicate the variation (degradation, stable state or improvement) of ICF values.
4. *Discovering information regarding the ICF values*: To remove this problem, use visual means to emphasize that there is some relevant supplemental information regarding the ICF values.

7.3 Recommendations Task 4

The recommendations resulting from this task can be split into two different aspects:

1. *Information visualization*: The time period over which the trend of the assessment (degradation, constant state, and improvement) is taking place has to be emphasized. A reduction of the amount of information displayed to the user will ease interpreting the data. The colors red, yellow, and green should consistently and exclusively be used to depict improvement, degradation, or a constant health status.

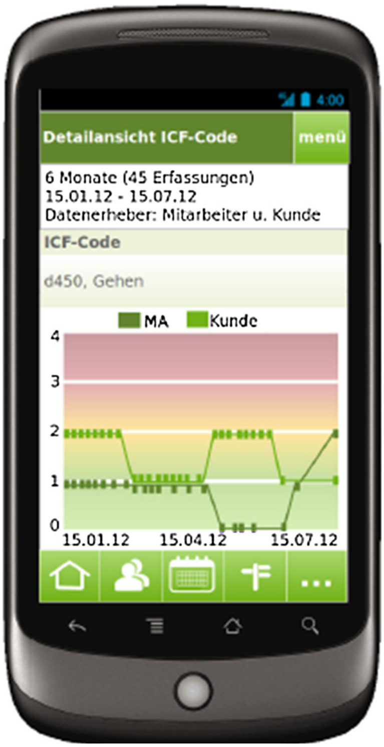

Fig. 8 Details view for ICF code "d450 Gehen"

2. *User expectation*: The easiest solution is to provide an acknowledge button as the majority of the test participants expects this button. Notably, the same issue has been raised in the walk-through cycle before conducting the experiment.

8 Related work

Nielsen and Landauer (1993) applied their ROI (Return on Investment) model to analyze datasets from verbal protocol techniques and expert-based evaluations and showed that a sample composed of a range of between three and five subjects with appropriate skills was generally enough to assess an interaction with a Web interface and identify at least 80 % of the interface problems. This result has been confirmed by several studies (Nielsen 1995, 2000, 2012; Virzi 1990, 1992) and is widely known as the five-user assumption (Nielsen and Landauer 1993). Spool and Schroeder (2001) provided one of the first studies that reflected on the five-user assumption, reporting on an experiment in which they found that five users were far too few.

User-centered design integrates both, design and evaluation, by focusing on the interface needed to meet the specific needs of the users. Rather than being considered as an ideal approach, user-centered design turned out to be a pragmatic or even necessary way forward. This behavior was justified on the grounds that developers were focused on controlling the costs of design, and where therefore looking on low-cost techniques instead of what might be an ideal approach (Borsci et al. 2013). In particular, this raised the question about the cost of design in comparison with the overall lifecycle costs. Kurosu (2007) proposed a "big usability approach," in which the evaluation is fully integrated into the product development cycle in the context of the user needs. Spool and Schroeder (2001) provided one of the first studies that reflected on Nielsen's (2000) five-user assumption, reporting an experiment in which they found that five users were far too few. They described an evaluation of four Web sites by 49 subjects and reported that to identify 85 % of the problems required considerably more than five subjects. Alshamari et al. (2009) contrasted Nielson's insights that five users would identify 80–85 % of the issues with a study from Lindgaard and Chattrartichart that managed only to identify 35 % of the usability problems. Borsci et al. (2013) argue that fifty post-2001 citations of Nielsen and Landauer (1993) demonstrate a continued interest in the diverging views on the five-user assumption.

Recent research (Harrison et al. 2013) clearly points out that the most prominent usability models overlook important aspects of mobility such as cognitive load. Cognitive load refers to the amount of processing required by the user to use the app. In traditional usability studies, a common assumption is that the user is performing only a single task. In our specific setting, the nurses and physiotherapists are entering health-related information while paying attention to the patient, i.e., the users are performing a second task and thus cannot pay full attention to the app. This aspect has not been addressed by our study; however, for future research, it is important to consider both dimensions when assessing the usability of mobile apps. For this reason, Harrison et al. (2013) present the PACMAD (People At the Centre of Mobile Application Development) usability model, which is designed to address the limitations of the prominent usability models when it comes to software development for mobile devices.

Our approach follows the idea of integrating user-centered design as early as possible. In this respect, paper prototypes turned out to be well suited as they can be used for clarifying the requirements without any major investment in technology or/and development. Our experience shows that supporting the usability evaluation by interviews allows one for discovering usability flaws in the design even with a couple of test participants per user group. This corresponds with earlier experiences regarding the application of the Thinking-Aloud method when conducted in an adequate manner (Holzinger 2005; Holzinger and Brown 2008; Holzinger and Slany 2006). This procedure allows us to test as whether we had the final product, but has the advantage that we focus on usefulness and usability at a very early point in the product life cycle.

Kangas and Kinnunen (2005) discuss the user-centered design process for mobile applications pointing out the need to provide the real usage context. Like our research, Kangas and Kinnunen (2005) provide real-world case studies but rather address the consumer market than professional business applications. Although the authors point out the need for cheap methods to make quick UI prototypes, they do not address the economic benefits from early integration of UI tests within a resource-constrained setting (e.g., limited number of participants per user group).

When designing a mobile software in the medical domain, no feature should be added to the software only because it is easy or cheap to implement. Well-defined practices for development of useful and usable apps are of uttermost importance as employees in healthcare have a high exposure to occupational stress, and therefore, these quality attributes of mobile apps become crucial for acceptance among the staff.

Besides usefulness and usability, the quality of apps depends on a lengthy list of non-functional quality attributes. Among those attributes that are most relevant to mobile applications are platform openness, means of accessing hardware, integration of third-party apps, reliability and offline functionality, multithreading, mobile device management, and security and privacy. Many of these issues have been addressed for Web applications and that knowledge provides an excellent starting point for studying quality of mobile applications. In our specific case, understanding the implications of Web service architecture and the impact on the desired quality attributes turned out to be crucial. The REST-based service architecture fulfills the requirements for our business application with respect to integrity of the recorded data. However, in general, it is an open issue, whether new techniques are needed for assuring data integrity, or whether the synchronization techniques from traditional client–server computing are sufficient in the near future. Wasserman (2010) also points out the need for further research in this direction and—besides of user experience—proposes to incorporate research on non-functional quality attributes into the research agenda for mobile software engineering.

Users are willing to tolerate quite much when it comes to quality of mobile apps as long as the apps are useful.[1] In our context, being useful means that the usage of ICF and ICD classification scheme has to provide assistance during every day work.

Bender (2010) points out that numerous health organizations experience problems when applying the ICF classification in practice. According to Bender (2010), the reason for this is the lack of guidelines for practical application of the complex ICF classification scheme. Further information regarding ICF, the available software products and the efforts to make these products available on mobile devices can be found in Peischl et al. (2013).

9 Conclusion

Numerous studies in the past have reported the benefits of a strong commitment to usability in the software development life cycle. The observable benefits of user-centered software engineering in particular hold for mobile apps. In this article, we report on the fine art of user-centered software development in the field of mobile medical apps. Our collaboration partner—an Austrian SME—provides integrated solutions for hospitals and nursing homes including mobile apps. Medical apps turn out to be very demanding with respect to a couple of quality attributes.

First, the platform being addressed influences non-functional key quality attributes. On the example of our medical app, we revisit the process of selecting the adequate platform taking into account issues such as the openness of the platform, ease of hardware access, integration capabilities with respect to third-party apps, offline availability, multithreading, mobile device management, and the crucial issue of security and privacy in the medical domain.

Second, this article contributes to the advancement of user-centered design in the field of mobile app development. In practice, high costs and stringent time planning prevent the use of usability engineering techniques. In order to address user-centered design in spite of limited resources, we propose to integrate usability evaluation in a very early stage of the mobile software life cycle. In this context, we outline how we took into account the usefulness (using the ICF and ICD classification scheme) and the usability of the mobile medical app. This article provides qualitative and quantitative insights on the experimental

[1] Personal communication at the Workshop@BritishHCI '13, Brunel University, London, September 9, 2013. There are two observations motivating this behavior of users. First, as the consumer market provides apps for free, users are willing to tolerate faults in apps. Second, continuous deployment via internet allows for fast removal of bugs, once they have been identified and reported.

evaluation. In detail, we report on the detected usability flaws in a very early state of the development cycle. The related work surveys the various points of view regarding the minimal number of participants for meaningful usability testing and refers to similar studies.

References

Alshamari, M., Mayhew, P., et al. (2009). Technical review: Current issues of usability testing. *IETE Technical Review, 26*(6), 402.

Apple. (2012), 'Ios security'. Accessed on December 18, 2013. http://www.apple.com/ipad/business/docs/iOS_Security_Oct12.pdf.

Bahrami, A., Wang, C., Yuan, J. & Hunt, A. (2006). The workflow based architecture for mobile information access in occasionally connected computing. In *Services computing*, 2006. SCC'06. IEEE International Conference on', IEEE, pp. 406–413.

Bastien, J. (2010). Usability testing: A review of some methodological and technical aspects of the method. *International Journal of Medical Informatics, 79*(4), e18–e23.

Bender, D. (2010). *Voraussetzungen für die nachhaltige Anwendung der internationalen Klassifikation der Funktionsfähigkeit, Behinderung und Gesundheit (ICF) in der Rehabilitationspraxis: Ergebnisse einer Analyse im Spannungsfeld von globaler Konzeption und lokaler Umsetzung*. Marburg: Tectum.

Bloice, M., Simonic, K.-M., Kreuzthaler, M. & Holzinger, A. (2011). Development of an interactive application for learning medical procedures and clinical decision making. In *Information quality in e-Health*. Springer, pp. 211–224.

Boivie, I., Åborg, C., Persson, J., & Löfberg, M. (2003). Why usability gets lost or usability in in-house software development. *Interacting with Computers, 15*(4), 623–639.

Borsci, S., Macredie, R. D., Barnett, J., Martin, J., Kuljis, J. & Young, T. (2013). Reviewing and extending the five-user assumption: A grounded procedure for interaction evaluation. *ACM Transactions on Computer-Human Interaction (TOCHI), 20*(5), 29:1–29:23.

Cooke, L., & Mings, S. (2005). Connecting usability education and research with industry needs and practices. *IEEE Transactions on Professional Communication, 48*(3), 296–312.

Dolan, P. L. (2011). Doctors cite ease of use in rapid adoption of tablet computers. http://www.amednews.com/article/20110418/business/304189961/7/. Last accessed on October 2013.

Faily, S., & Lyle, J. (2013). Guidelines for integrating personas into software engineering tools. In *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*, ACM, pp. 69–74.

Ferk, M. (2013). Konzeption und Entwicklung einer gelegentlich verbundenen mobile Applikation für die Datenerfassung in Gesundheitsorganisationen. Master's thesis, Graz University of Technology, Technische Universität Graz.

Flood, D., Harrison, R., Iacob, C., Nosseir, A., & McDaid, K. (2012). Are mobile spreadsheet applications usable? *Journal of Organizational and End User Computing (JOEUC), 25*(1), 43–63.

Flood, D., McCaffery, F., Casey, V., & Regan, G. (2013). Med ud a process reference model for usability design in medical devices. In *South CHI 2013 international conference on human factors in computing and informatics*, Maribor, Slovenia.

Garcia, E., Martin, C., Garcia, A., Harrison, R. & Flood, D. (2011). Systematic analysis of mobile diabetes management applications on different platforms. In *Proceedings of the 7th conference on workgroup human-computer interaction and usability engineering of the Austrian computer society: information quality in e-Health*, USAB'11, Springer, Berlin, Heidelberg, pp. 379–396.

Google. (2012a). Notes on the implementation of encryption in android 3.0. Last accessed on December 2012. http://source.android.com/tech/encryption/android_crypto_implementation.html.

Google. (2012b). *Storage options*. Last accessed December 2013. http://developer.android.com/guide/topics/data/data-storage.html.

Greamo, C., & Ghosh, A. (2011). Sandboxing and virtualization: Modern tools for combating malware. *IEEE Security Privacy, 9*(2), 79–82.

Harrison, R., Flood, D., & Duce, D. (2013). Usability of mobile applications: Literature review and rationale for a new usability model. *Journal of Interaction Science, 1*(1), 1–16.

Holzinger, A. (2003). Experiences with User Centered Development (UCD) for the front end of the Virtual Medical Campus Graz. In J. A. Jacko & C. Stephanidis (Eds.), *Human-Computer Interaction (HCI) theory and practice* (Vol. 1, pp. 123–127). Mahwah, NJ: Lawrence Erlbaum.

Holzinger, A. (2005). Usability engineering methods for software developers. *Communications of the ACM, 48*(1), 71–74.
Holzinger, A., & Slany, W. (2006). Xp + ue xu praktische erfahrungen mit extreme usability. *Informatik-Spektrum, 29*(2), 91–97.
Holzinger, A., & Errath, M. (2007). Mobile computer web-application design in medicine: Some research based guidelines. *Universal Access in the Information Society, 6*(1), 31–41.
Holzinger, A., & Brown, S. (2008). Low cost prototyping: Part 2, or how to apply the thinking-aloud method efficiently. In *Proceedings of the 22nd British HCI group annual conference on people and computers: Culture, creativity, interaction-volume 2*. British Computer Society, pp. 217–218.
Holzinger, A., Searle, G., Kleinberger, T., Seffah, A., & Javahery, H. (2008). Investigating usability metrics for the design and development of applications for the elderly. In *Computers helping people with special needs*. Springer, pp. 98–105.
Holzinger, A., Popova, E., Peischl, B., & Ziefle, M. (2012a). On complexity reduction of user interfaces for safety-critical systems. In G. Quirchmayr, J. Basl, I. You, L. Xu, & E. Weippl (Eds.), *Multidisciplinary research and practice for information systems*. Lecture Notes in Computer Science (Vol. 7465, pp. 108–122). Springer Berlin Heidelberg.
Holzinger, A., Treitler, P., & Slany, W. (2012b). Making apps useable on multiple different mobile platforms: On interoperability for business application development on smartphones. In G. Quirchmayr, J. Basl, I. You, L. Xu, & E. Weippl (Eds.), *Multidisciplinary research and practice for information systems*. Lecture Notes in Computer Science (Vol. 7465, pp. 176–189). Springer Berlin Heidelberg.
Horsky, J., McColgan, K., Pang, J. E., Melnikas, A. J., Linder, J. A., Schnipper, J. L., et al. (2010). Complementary methods of system usability evaluation: Surveys and observations during software design and development cycles. *Journal of Biomedical Informatics, 43*(5), 782–790.
Iacob, C., Harrison, R., & Faily, S. (2014). Online reviews as first class artifacts in mobile app development. In G. Memmi & U. Blanke (Eds.), *Mobile computing, applications, and services*. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (Vol. 130, pp. 47–53). Springer International Publishing.
IDC. (2012). Android and iOS surge to new smartphone OS record in second quarter, according to IDC. Last accessed on October 2013. International Data Corporation, http://www.idc.com/getdoc.jsp?containerId=prUS23638712.
Inostroza, R., Rusu, C., Roncagliolo, S., Jiménez, C., & Rusu, V. (2012). Usability heuristics for touch-screen-based mobile devices. In *Ninth international conference on information technology: new generations (ITNG), 2012*, IEEE, pp. 662–667.
Kangas, E., & Kinnunen, T. (2005). Applying user-centered design to mobile application development. *Communications of the ACM, 48*(7), 55–59.
Karat, J. (1997). Evolving the scope of user-centered design. *Communications of the ACM, 40*(7), 33–38.
Kurosu, M. (2007). Concept of usability revisited. In *Human-computer interaction. Interaction design and usability*. Springer, pp. 579–586.
Larusdottir, M. K. (2011). Usability evaluation in software development practice. In *Human-computer interaction–INTERACT 2011*. Springer, pp. 430–433.
Liu, C., Zhu, Q., Holroyd, K. A., & Seng, E. K. (2011). Status and trends of mobile-health applications for iOS devices: A developer's perspective. *Journal of Systems and Software, 84*(11), 2022–2033.
Mayhew, D. J. (1991). *Principles and guidelines in software user interface design*. Upper Saddle River: Prentice-Hall Inc.
MH. (2012). Last accessed on October 2013. Manhattan Research. http://manhattanresearch.com/.
Nielsen, J. (1994). Estimating the number of subjects needed for a thinking aloud test. *International Journal of Human-Computer Studies, 41*(3), 385–397.
Nielsen, J. (1995). *Severity ratings for usability problems*. April 2014. http://useit.com/papers/heuristic/severityrating.html.
Nielsen, J. (2000). Why you only need to test with 5 users'. Last accessed on April 2014. http://www.useit.com/alertbox/20000319.html.
Nielsen, J. (2012). How many test users in a usability study?. Last accessed on April 2014. http://www.useit.com/alertbox/number-of-test-users.html.
Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on human factors in computing systems*, CHI'93, ACM, New York, NY, USA, pp. 206–213. http://doi.acm.org/10.1145/169059.169166.
Norman, D. A., & Draper, S. W. (1986). *User centered system design; New perspectives on human-computer interaction*. Hillsdale: L. Erlbaum Associates Inc.

Peischl, B., Ferk, M., & Holzinger, A. (2013). Integrating user-centred design in an early stage of mobile medical application prototyping—A case study on data acquistion in health organisations. In *DCNET/ICE-B/OPTICS*, pp. 185–195.
Pilgrim, M. (2011). Dive into html5: No 7. the past, present and future of local storage for web applications. Last accessed on October 2013. http://diveintohtml5.info/storage.html.
R2G. (2010). Global mhealth developer survey. Last accessed on October 2013. http://www.research2guidance.com.
Raggett, D. (2014). System applications working group—w3c. Last accessed on April 2014. http://www.w3.org/2012/sysapps/.
Rousset, D. (2013). Introduction to html5 web workers: The javascript multi-threading approach. Last accessed on October 2013. http://msdn.microsoft.com/en-us/hh549259.aspx.
Rubin, J., & Chisnell, D. (2008). *Handbook of usability testing: How to plan, design, and conduct effective tests.* Wiley.com.
Seffah, A., Donyaee, M., Kline, R. B., & Padda, H. K. (2006). Usability measurement and metrics: A consolidated model. *Software Quality Journal, 14*(2), 159–178.
Short, E. J., Evans, S. W., Friebert, S. E., & Schatschneider, C. W. (1991). Thinking aloud during problem solving: Facilitation effects. *Learning and Individual Differences, 3*(2), 109–122.
Spool, J., & Schroeder, W. (2001). Testing web sites: Five users is nowhere near enough. In *CHI'01 Extended abstracts on human factors in computing systems*, CHI EA'01, ACM, New York, NY, USA, pp. 285–286.
Thimbleby, H. (2007). User-centered methods are insufficient for safety critical systems. In A. Holzinger (Ed.), *HCI and usability for medicine and health care*. Lecture Notes in Computer Science (Vol. 4799, pp. 1–20). Springer Berlin Heidelberg.
Turisco, F., & Garzone, M. (2013). *Harnessing the value of mhealth for your organization*. Last accessed on October 2013.
Virzi, R. A. (1990). Streamlining the design process: Running fewer subjects. In *Proceedings of the human factors and ergonomics society annual meeting* (Vol. 34). SAGE Publications, pp. 291–294.
Virzi, R. A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors: The Journal of the Human Factors and Ergonomics Society, 34*(4), 457–468.
Vredenburg, K., Isensee, S., Righi, C., & Design, U.-C. (2001). *An integrated approach*. Englewood Cliffs: Prentice Hall.
Wasserman, A. I. (2010). Software engineering issues for mobile application development. In *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, FoSER'10* (pp. 397–400), Santa Fe, NM. New York, NY: ACM
Weippl, E., Holzinger, A., & Tjoa, A. M. (2006). Security aspects of ubiquitous computing in health care. *e & i Elektrotechnik und Informationstechnik, 123*(4), 156–161.
WHO. (2010a). Icf application and training tools. Last accessed on November 2013. World Health Orgranization. http://www.who.int/classifications/icf/icfapptraining/en/index.html.
WHO. (2010b). International classification of diseases (icd). Last accessed on November 2013. World Health Organization. http://www.whoint/classifications/icd/en/.
WHO. (2010c). International classification of functioning, disability and health. Last accessed on November 2013. World Health Organization. http://www.who.int/classifications/icf/en/.

Bernhard Peischl is the coordinator for the competence network Softnet Austria (managing the innovation programs K-net Softnet Austria and COMET K-Projekt Softnet Austria II). He received a MS in telecommunications engineering (2001) and a Ph.D in computer science (2004) from the Technische Universität Graz, Austria. He is responsible for managing the network's R&D activities and a number of applied research projects dealing with software test, verification and debugging and user-centered software engineering. He has co-authored over 60 scientific articles on peer-reviewed workshops, conferences and in scientific journals.

Michaela Ferk finished her master's studies in Software Development and Business Management at Graz University of Technology in 2013 with distinction. She has been developing innovative mobile applications for over two years now. Currently, she works at the Personal Ubiquitous Computing department at the Know Center in Graz. Michaela's second field of interest is law. Currently, she works toward her second diploma examination in her law studies at Karl-Franzens university Graz.

Andreas Holzinger is head of the Research Unit HCI, Institute for Medical Informatics, Medical University Graz, and Associate Professor at Graz University of Technology, where he teaches Biomedical Informatics at the Institute of Bioinformatics and Genomics, and supervising engineering students at the Institute of Information Systems and Computer Media. He started as apprentice in Information Technology, and after his second chance education and IT industry experience, he began his academic studies at the age of 26, studied Information Technology and Physics, Psychology and Media at Graz University, obtained his PhD in Cognitive Science from Graz University in 1997 and his Habilitation in Computer Science from Graz University of Technology in 2003. Andreas was Visiting Professor in Berlin, Innsbruck, Vienna, London and Aachen. Since 1999 he has participated in leading positions in 30+ R&D multinational projects, budget 3+ MEUR, 300+ publications, 3300+ citations, h-index = 26, g-index = 68. He is founder and leader of the Expert Network HCI-KDD.