NordDesign 2016
August 10 – 12, 2016
Trondheim, Norway

Enhancing Early Mobile Application Prototype Testing: Requirements analysis through interviews and a System Design

Sebastian Meyer, Franziska Häger, Matthias Uflacker

Hasso Plattner Institute
sebastian.meyer@student.hpi.de, franziska.haeger@hpi.de, matthias.uflacker@hpi.de

Abstract

During the last three decades the number of active mobile devices went from zero to 7.2 billion devices. And research predicts the usage of mobile devices will replace desktop systems in many areas i.e. mobile devices are becoming the primary way for people to access the internet. Thus, the mobile market has become extremely competitive. In this market delivering high quality software which addresses the user's needs and has an appealing interface becomes a competitive advantage.

To achieve a good user experience for their products, companies are aiming to involve end-user into their development processes by adopting user-centred approaches. These approaches rely heavily on repeated user-research and validating prototypes with end users early on and throughout the development process. Research agrees that testing mobile application prototypes should be in the right context, at best in the field on the mobile device. However, there is still little experience on mobile application testing and standardized approaches or tools are not yet established. Especially for early non-technical prototypes such as wireframes or clickable user interface dummies testing on the actual device is challenging.

This paper provides a glimpse into current challenges of testing for mobile application prototypes and proposes a system to solve them. Current challenges derived from interviews with three major software companies on current approaches to mobile testing with regards to tooling and processes are presented. Based on these challenges a list of requirements for a system that allows testing of early mobile prototypes on the actual device and collecting data around the tests such as gestures used, environmental data and so on is composed. A system architecture for a software system which can fulfil these requirements is presented and its possibilities and limitations are discussed.

Keywords: Mobile Prototype Testing, System Design, Software Engineering

1 Introduction

Software systems have existed for many decades and underwent enormous changes since they first appeared. They evolved from being mainly functional systems operated by few expert users to reaching into nearly every aspect of our lives and making almost everyone a software user. Consequently, the software industry has evolved to one of the most influential industries since it is integrated into every other industry. Therefore, todays software needs to be more than just functional: it has to be usable for novices - hence clear and easily learnable (Nielsen, 1993).

Additionally, mobile devices and applications became increasingly important in our globally connected world. In only three decades the number of active mobile devices went from zero to 7.2 billion devices (Cisco, 2014) just exceeding the number of humans recently (Gunelius, 2014). Research predicts the usage of mobile devices will replace desktop systems in many areas (Cisco, 2014; Pepitone, 2013), e.g., as the primary way to access the internet (Bosomworth, 2014). The attention and usage of mobile devices in the business context is increasing rapidly as well (IDC, 2012). Thus, the market for mobile applications has become extremely competitive. In this market, delivering high quality software which addresses the user's needs and has appealing user interfaces becomes a competitive advantage.

With rising awareness for a good user experience, companies are incorporating user-centred approaches to involve end users and other stakeholders into the development. This is also suggested by current research to provide high-quality software with satisfying usability (Hornbaek, 2006; Rasmusson, 2010; Coursaris & Kim, 2011). The main focus of user-centred approaches is to shift from traditional design that is driven by technology, to an ongoing focus on customer needs. Most of these user-centred approaches are based on early validations with end users and prototypes with different levels of fidelity. Thus, Prototype testing is a crucial part of end user involvement, because typical end users do not have a technical background therefore it is not possible to talk directly about technical requirements (Nielsen, 1993). Over time, researchers have developed different approaches and tools to support end user testing for desktop software. However, experience in testing mobile application prototypes and standards are not established yet (Hornbaek, 2006; Coursaris & Kim, 2011).

This paper presents a summary of explorative research on testing mobile application prototypes at SAP SE, Microsoft Corp. and Nokia HERE focusing on challenges the companies are currently facing and shortcomings of the existing tools and methods they apply. Based on these challenges a list of requirements for a system that allows testing of early mobile prototypes on the actual device and collecting data around the tests such as gestures used, environmental data and so on is composed. A system architecture for a software system which can fulfil these requirements is presented and its possibilities and limitations are discussed.

2 Related Work and Existing Tools

Compared to usability studies for desktop software, mobile applications have different characteristics (Nielsen & Budiu, 2013). Two main aspects have to be taken into consideration when preparing mobile usability studies: the context in which the mobile application is used and the technical differences of mobile devices compared to desktop systems, e.g. smaller screens. Both lead to challenges and opportunities in the execution of mobile usability studies.

Dey and Abowd define context as "any information that can be used to characterize the situation of entities […] that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity, and state of people, groups, and computational and physical objects” (Dey & Abowd, 2001). The term context in this paper adheres to this definition. Related literature such as the reviews of usability studies by Krannich or Coursaris and Kim or the work of Tamminen et. Al. emphasize the importance of a real test context when it comes to testing for mobile applications, instead of testing in a controlled lab environment (Krannich, 2010; Coursaris & Kim, 2011). Additionally, both works try to specify context more deeply. Krannich sees two dimensions relevant to the mobile context: the cognitive dimension, and the situational dimension. Coursaris and Kim base context on four variables: user (knowledge, experience, perception), environment (physical influences), technology (device type, interface, input mode), and task (open vs. closed). Additionally, to his review of mobile usability studies Krannich proposes a tool-based model for rapid prototype testing in a real context, which makes his work by far the most similar to this paper. In contrast to his general technical tool support for field testing based on a review of usability studies, this paper derives requirements for a tool to support testing of web-based prototypes from interviews in the software industry.

The most obvious challenge for mobile usability studies in the real context is the small device and screen size, which makes it difficult to observe and record interactions on the screen during testing. Even lab-settings have to be adjusted to accommodate testing on such small devices. Other challenges include natural influences on mobile devices or restrictions from the mobile operating systems. For example, limited network coverage or memory capacity can be a problem for screen recording software. Another technical difference is the variety of sensors mobile devices are equipped with, e.g. GPS, light sensors, gyroscope, microphones or cameras. Literature already mentions the use of such sensors to provide additional value to the users (Coursaris & Kim, 2011; Wood & Romero, 2010) and to utilize them to study the context of a user (Arase et al., 2010).

Literature mentions adaptions to the hardware setup of usability studies made to accommodate for mobile testing in the field. For example, it is common to equip participants with additional cameras to record user interactions. One example is a video camera mounted on the mobile device to capture the screen and the interactions (Kiljander, 2004; Krannich, 2010). However, this setup has a different haptic, weight, and additional cables, which make the interactions with the device unnatural (Krannich, 2010). Other camera settings include glasses with a built-in camera or sport cameras worn on the user’s head. Usability tests conducted in a lab environment often utilize screen mirroring software to stream the test screen to an observer's screen. In general, this is also possible for mobile devices.

Many providers of prototyping tools have adapted their tools to the need of creating and testing mobile prototypes. Most of these prototype editors are either web-based or have standalone desktop software programs. Either way, nearly all of them offer to export the prototype to a format that can be presented in a browser. Additionally, some tools, allow prototypes to be shared or prototyped on the mobile device itself. Another kind of software tool that supports mobile testing are software development kits (SDK), which can be included into native applications. The main disadvantages are the additional effort needed and the need for a prototype that is already built as native software, e.g. as an android app.

3 Industry Interviews

In order to create a software system that truly enhances early mobile prototype testing we interviewed employees of three software companies: SAP SE, one of the largest software companies in the world, focusing on business software; Microsoft Corp., the world's biggest software company, offering consumer as well as enterprise software; and Nokia HERE, specialised in mobile applications with a focus on location based services.

A total of 25 interviews with employees of these companies helped us to investigate how software companies use and especially test prototypes in their development processes and to identify existing challenges. This section will summarize the most important challenges the companies face with regard to early mobile prototype testing. A more detailed description of the case studies can be found in (Meyer et. Al., 2016), discussing prototyping and testing in the case companies in general and (Meyer et. Al., 2016) focusing on mobile prototyping and testing in the case companies.

3.1 Wishes and Challenges

In all companies, mobile applications are commonly tested in lab-settings but the existing settings and tools are adapted to accommodate mobile application testing. It was pointed out that, as a main difference to desktop application testing, people interact in a more natural way with a mobile device, than with a mouse and keyboard setup. Moreover, mobile applications have a more interactive application design with more content changes and scrolling than desktop applications due to the smaller screen. All three case companies agree on the importance of testing mobile applications in the right context, which means at least on a mobile device. However, they experienced significant challenges due to the form factor of mobile devices which introduces technical challenges as well as challenges related to the execution context, the test organization and organizational resources.

First and foremost, industry experts mentioned the limited observation ability of user interaction on a mobile device as a challenge. In contrast to a traditional desktop environment, it is hard to physically see the screen even when being next to a test participant. Moreover, users often cover the view of observers with their hands while interacting with a touch screen or even experience the observer as obtrusive while being observed very closely.

To achieve a user behavior as natural as possible during prototype testing it is crucial to have a test setting and tools in place that allow for indirect observation. The interviewees mentioned three different settings to record the screen of mobile devices during testing:

1. The first variant aims to mirror the screen of the device on a second screen with the help of standard screen sharing apps, which were not specifically build for the purpose of prototype testing. The main downside of this approach is that touch points of the user's interactions with a mobile prototype are not visible to the researcher when the prototype runs on a mobile device, especially if the mirrored screen is recorded for later replay. This important due to differences of interaction with mobile prototypes as opposed to desktop prototypes, e.g. the missing mouse pointer as indicator of the users focus and different movement patterns.
2. Another approach to recording the screen is the use of a camera mounted onto the device with a special bracket. This makes it possible to see the interaction on the screen. The unnatural feel of the device and interaction was pointed out as main

disadvantage. According to statements, this setting greatly interferes and influences the user behavior, which makes it problematic to use for cases where the context is critical to the experience.

3. A less obtrusive version of filming the interaction on the screen of a mobile device is using the existing camera settings in a laboratory. The cameras have to follow the screen of the device; this is however prone to failure. Thus, the user is asked to keep the device within a certain box-shaped area marked by tape. A camera is focused on this area to record the interactions between the device and the screen. Disadvantages to this approach are that the user's hand sometimes covers the screen and that it is unnatural to keep the device within the box.

Observing the user interaction on a mobile device screen in the field becomes even more challenging. On one hand, the behavior in is influenced when users or the mobile device are equipped with cameras. On the other hand, screen mirroring has not been applied in field testing situations due to its dependence on connectivity and its high risk of failure. However, only observing without recording aggravates the problem of seeing all interactions properly. Additionally, unrecorded observations cannot be replayed during analysis, which is especially relevant for field research.

Interviewees expressed difficulties to collect data from the mobile devices itself, especially constant video or screen captures. Constant video streaming is often dependent on hardware, other tools that collect this kind of data or developers to integrate the respective code into prototypes. Moreover, low-fidelity prototypes do not allow for the integration of such code. At SAP, prototypes often have to be tested remotely or are just sent out for feedback, due to time and budget constraints, although this is not a preferred test setting. Interviewees expressed it would be beneficial to actually know what the user/customer has done when getting textual feedback and descriptions from such a remote testing. To achieve such information prototypes native to the devices operating system can enable features, i.e. code tracking the user's behavior. However, their setup effort can easily become too high and too costly.

Prototype testing for native applications introduces further technical challenges when using low-fidelity prototypes. Prototyping tools used within the case companies are often generating HTML or PDF files as output. To simulate native apps using such outputs in a browser the prototypes need to be started in full screen mode, and gestures of the host application need to be disabled in order to avoid side effects such as scaling or gestures starting browser features.

A major topic with regards to challenges on mobile application testing for all case companies is the execution context of a mobile research study. The context for mobile applications is seen as very important by the user researcher experts. And our interviews agree with literature that a mobile prototype should be presented on the actual device in a real world scenario instead of a laboratory to get people into the right cognitive context. Additionally, mobile apps are often used in conjunction with other apps, making it essential to understand the whole usage scenario for the app. Testing of mobile applications in a real life context is very complex but essential for some applications, e.g. location-based services. Recordings become more important as data source, because live note taking is limited by various physical influences.

In addition to technical barriers and challenges related to the execution context, there are often resource, time and budget constraints within large enterprises. Due to the previously pointed out challenges, mobile research studies logistically much more effort and technically more challenging, especially when working with low-fidelity prototypes and therefore need more people and resources to be conducted. This is especially the case for studies which are done in the real physical environment and where people need to follow participants around.

Overall, a conflict exists between overcoming these challenges and the return of investment of the prototype testing. Due to limited budget and time constraints companies, often only adjust their lab set- up to accommodate for testing on mobile devices instead of doing what would be technically possible, and best suited for the research questions.

Table 4.1: List of requirements on a tool for testing early mobile prototypes

Type	ID	Prio	Description
Functional requirements	R01	High	The tool should present a prototype on the actual mobile device.
	R02	High	The tool should mirror the content of the test device to multiple other screens (observer screens).
	R03	High	The tool should show all interactions of a test participant with a mobile device on an observer screen.
	R04	High	The tool should highlight all touch points / gestures to the observer.
	R05	High	The tool should support multi-touch gestures.
	R06	High	The tool should deactivate all standard gestures of the test device.
	R07	High	The tool should support a broad range of prototypes.
	R08	Med	The tool should present the prototype in full screen mode without borders or mobile browser elements.
	R09	Med	The tool should record all prototype implemented gestures.
	R10	Med	The tool should record all tried gestures, even if they are not implemented in the prototype.
	R11	Med	The tool should replay recorded test sessions.
	R12	Low	The tool should collect contextual information about the device such as device type, screen resolution, operating system, and browser version.
	R13	Low	The tool should synchronize all recorded data sources for playback.
General requirements	R14	High	Every influence on the device's handling should be avoided, in particular connected cables and cameras mounted onto the device.
	R15	High	The tool should be invisible to the end-user.
	R16	High	The tool should not be platform specific.
	R17	High	Response times should not differ from the original prototype.
	R18	Med	It should not be necessary to change the actual prototype or include additional software to enable testing.
	R19	Med	The tool should be easy to set up.
	R20	Med	The tool should support confidentiality.
	R21	Med	The tool should be available, especially for remote sessions.
Requirements for remote testing	R22	High	The tool should not install the prototype on the user's device.
	R23	High	The tool should support expiration dates for access.
	R24	High	The tool should restrict access to the prototype.
	R25	Med	The tool should send allow to access the prototype remotely.
	R26	Med	The tool should record the user and its interaction with the mobile device as soon as the prototype is used.
	R27	Low	The tool should collect contextual information such as network coverage, GPS, light situation and execution time.

4 Requirements on a Tool for Testing Early Mobile Prototypes

Table 4.1 lists our set of requirements on a tool for early mobile prototype testing, which were derived from the challenges in the previous section. They include functional requirements; general requirements, that cannot directly be translated to a feature and complementary requirements needed to support unmoderated remote testing. The requirements are ordered by type and priority. The term tool refers to a software system, which itself may consist of several components.

4.1 Comparison of Requirements and existing Tools

A huge gap can be outlined in comparison to the existing tools presented in section 2. There are plenty of existing tools that support prototype creation for mobile devices and often feature some way to present the prototype on a mobile device. However, these tools are not designed to support the actual testing. Usually, researchers use a combination of prototyping tool and screen or video recording software. Thus, one of the most important requirements, gesture and touch visualization, is not supported by any of the used tools. Moreover, it is not possible for the researcher to actually see gestures which are not implemented but used by the user.

For testing on the mobile device, the recording technique often influences the user's behaviour. Either the user has to keep the test device within a certain area or an additional camera is mounted onto the device. Furthermore, real life testing of mobile prototypes is often neglected, as there is no suitable tool support available. Another kind of tool for testing mobile applications is the integration of a software development kit (SDK) into software prototypes. However, the integration requires additional effort and does not work for early prototypes, which stands in contrast to requirements R07 and R18.

5 System Proposal for Testing Mobile Prototypes

As pointed out in section 2, there are fundamental differences between testing desktop software and mobile applications. So far, the tools for desktop testing have only been adapted to mobile use. In fact, testing mobile prototypes is more complex and requires additional considerations. Consequently, a gap can be found between requirements for mobile testing and currently available testing tools (see section 4.1). As discussed in sections 2 and 3, the minimal context for most mobile applications testing is to run the prototype on an actual mobile device. Moreover, for many use cases, it is suggested to test prototypes in their original context. Consequently, the following solution proposal focuses on supporting tests on the actual mobile device and in the field. To achieve a broad application of the system, it focuses on supporting tests for web-based prototypes and applications. HTML is the most common technique for interactive prototypes and is the output of many prototyping tools. Furthermore, it is platform independent and more and more applications are based on its modern web technologies. The proposed solution aims to support mobile prototype testing within software development processes of small and large companies.

5.1 General System Architecture

The core of the proposed solution is a wrapper for web-based prototypes which is executed server-sided. In addition, it consists of one hybrid mobile application for each supported platform. An abstracted system architecture is depicted in Figure 5.1. It shows a webserver, which is running a prototype wrapper as a central component. The general principle is that a request to the prototype wrapper is done from within a hybrid mobile application. The wrapper then requests the prototype from its original source and creates a local copy on the same server.

The copy is included in an iframe, which the wrapper returns as view to the mobile device. The prototype is presented as it would be accessed directly from within the hybrid app, thus the end user does not notice this additional layer. Multiple observer devices can register themselves on the server to view a testing session.

The system supports all kinds of web-based prototypes, thus the wrapper itself is also realized with web technologies. The copy of the prototype is necessary to address the highly important R04 - visualize all touch points / gestures to the observer. On web pages every interaction such as clicks or touches triggers a DOM event, which can be caught and visualized with the help of JavaScript. In order to not contradict R07 and R18 the JavaScript code needs to be deployed into the wrapper, which makes it necessary to deploy wrapper and prototype in the same server due to the JavaScript same-origin policy, a security feature restricting access to documents not on the same server.

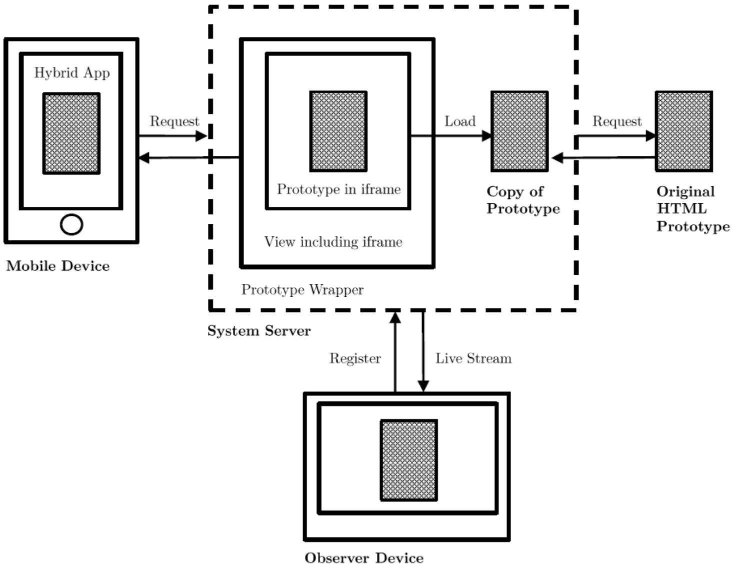

Figure 5.1: General system architecture for testing of prototypes on mobile devices

To avoid any interference between the prototype and the wrapper, the prototype is loaded within a full screen iframe. Thus, it is completely isolated and appears as it would be accessed directly. Touch events and gestures can be recognized for instance by using the touch framework Hammer.js. Once an event is triggered, it provokes its general event handling and is then bubbled up to the body element, where it is caught by the wrapper and its register event listeners. This scenario has been implemented prototypically to ensure the technical feasibility. With access to all events occurring in the prototype, the wrapper can record and analyse these events easily. Additionally, the wrapper collects and records contextual information as defined in R12.
To address the requirement R03 - streaming video signals to an observer screen, WebSockets are proposed as one possible technical solution.

With this approach, the influence on the performance is kept to a minimum because there are no further resources required on the device itself. In a simple form, the wrapper could be used in any mobile web browser. According to requirement R08, the system shall present the prototype in full screen and deactivate standard browser interactions. Thus, it is proposed to have additional hybrid apps for major platforms. The hybrid apps only consist of an app container with a web view for the wrapper. This way, the requirements R06, R08, and R15 are best incorporated in contrast to native applications or using a mobile web browser. However, there is a small trade-off between the requirements R19 to allow easy sharing of the prototype. Furthermore, hybrid apps allow various valuable extensions of the system, which will be described in the following section.

6 Discussion

The proposed solution can be utilized in various scenarios and improve existing user test approaches for mobile devices. As depicted in Figure 6.1, the prototype wrapper can be used for all fake prototypes which are web-based. Moreover, if the final technology stack is web based, it can also be applied for testing of product pre-versions. For these kinds of prototypes, the system can be used for both laboratory tests and field tests. Due to the fact that moderated testing provides more valuable feedback and is preferred by a majority of the interviewed researchers, the system primarily focuses on the support of moderated test sessions. However, the system can be easily extended to additionally support unmoderated remote testing.

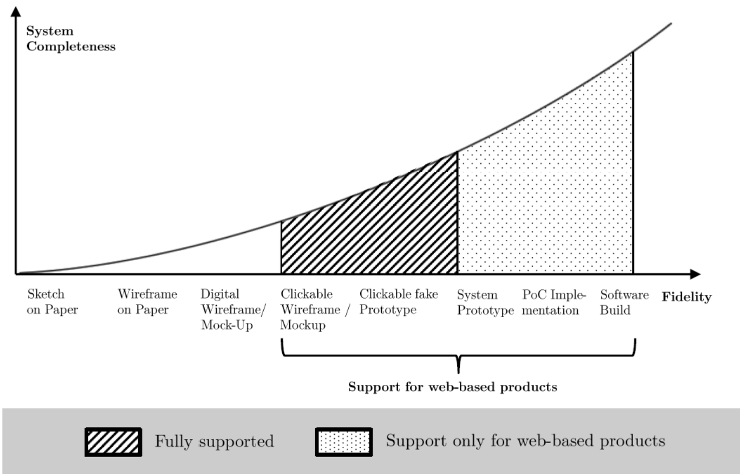

Figure 6.1: Field of application for the proposed system

As the proposed system supports moderated testing in a lab environment, there are no additional cables or cameras mounted on the mobile device. A user can interact with the device in a natural way and does not notice the additional layer. As pointed out in sections 2 and 3, for many cases testing should be conducted in a real context. Even with bad network coverage, field testing is still supported with the proposed system. It streams events and user interactions to a server where they are recorded and analysed while the WebSocket protocol handles the sending of events and manages connection interruptions. In this way test participants can use a mobile device naturally without interference of recording tools and the researcher can concentrate on observing the participants and note taking.

Another interesting use case, mentioned by interviewees of the case studies is the support for long term studies, as some interactions occur very rarely and cannot be observed by moderated field testing. A running web-based software prototype would allow the system to collect data on the general usage behaviour, e.g. analysing execution and foreground times. In addition, it would be possible to record contextual information such as GPS position, device type, screen resolution, operating system version, etc. With extended hybrid apps it would even be possible to access system sensors such as the accelerometer and light sensor to gather information about physical influences or the device camera to capture rarely occurring moments or interactions.

Overall, the proposed system can be applied in a variety of use cases and provides plenty of opportunities for extension. It can be used in addition to existing prototyping tools with an HTML output.

7 Sumary

This paper presents challenges for testing mobile prototype applications as derived from interviews with three major software vendors. Based on these challenges requirements for a software tool supporting the testing of mobile application prototypes were derived and compared to existing tools. As the existing tools could not fulfil all the requirements the paper presents a system design that would. The decisions made on the architecture of the proposed system and its limitations are briefly discussed. The implementation of the system proposal is out of scope of this paper. Nevertheless, the technical feasibility has been proven.

References

Arase, Y.R.F., Xie, X. 2010. User activity understanding from mobile phone sensors. Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing - Ubicomp '10, 391.

Cisco (2014). Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2013-2018.

Coursaris, C.K., Kim, D.J. (2011). A meta-analytical review of empirical mobile usability studies. Journal of usability studies, 16(3), 117-171.

Dey, A.K., Abowd, G.D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction, 16(2), 97-166.

Gunelius, S. (2014). Mobile Devices to Surpass the Number of People on Earth – Infographic. http://aci.info/2014/05/03/mobile-devices-to-surpass-the-number-of-people-on-earth-infographic/ (retrieved March 6, 2016).

Hornbaek, K. (2006). Current practice in measuring usability: Challenges to usability studies and research. International Journal of Human-Computer Studies, 64(2), 79-102.

IDC (2012). IDC Worldwide Mobile Worker Population 2011-2015 Forecast (Doc #232073). Tech. rept. IDC.

Kiljander, H. 2004. Evolution and Usability of Mobile Phone Interaction Styles. Ph.D. thesis, Helsinki University of Technology, Otamedia Oy.

Krannich, D. (2010). Mobile Usability-Testing - Ein toolbasiertes Vorgehensmodell zum Rapid-Prototyping und Usability-Testing von Mobilen Systemen im originären Benutzungskontext. Dissertation Universität Bremen, Fachbereich 3.

Meyer, S., Häger F., Uflacker M. (2016). Finding the Appropriate Test Context for Mobile Application Prototypes - A Decision Model Based on a Case Study. In DS 84: Proceedings of the DESIGN 2016 14th International Design Conference, 843-852.

Meyer, S., Häger F., Uflacker M. (2016). Case Studies on End-User Engagement and Prototyping during Software Development - An Overview of Current Practices in the IT Industry. In Design Thinking Research - Taking Breakthrough Innovation Home. Springer International Publishing.

Nielsen, J. (1993). Usability Engineering, Interactive Technologies, 44(3). Morgan Kaufmann.

Nielsen, J., Budiu, R. (2013). Mobile Usability. New Riders Berkely USA.

Pepitone, J. (2013). Worst PC sales drop in history. http://money.cnn.com/2013/04/10/technology/pc-sales/ (retrieved October 10th, 2014).
Rasmusson, J. (2010). The Agile Samurai - How Agile Masters Deliver Great Software. The Pragmatic Bookshelf.
Tamminen, S., Oulasvirta, A., Toiskallio, K., Kankainen, A. (2004) Understanding mobile contexts. Personal and Ubiquitous Computing 8(2), 135-143.
Wood, S., Romero, P. (2010). User-centred design for a mobile learning application. In Proceedings of the 3rd Mexican Workshop on Human Computer Interaction, 77-84.