

Design and Implementation of a Usability-Framework for Smartwatches

Steffen Zenker
University of Goettingen
steffen.zenker@uni-goettingen.de

Sebastian Hobert
University of Goettingen
shobert@uni-goettingen.de

Abstract

Due to technological developments in the last decade, the class of wearable computers arose which offers innovative access to human-computer interaction. Especially smartwatches attracted attention and are established as a permanently worn computer device on many wrists nowadays. In particular, for new technologies usability is an important success factor. Although usability is a well-known domain with a long research history, unique characteristics of smartwatch applications complicate the utilization of recent usability analysis methods. Therefore, we survey recent techniques for the usability analysis, outline and respectively adapt suited approaches based on the requirements induced by the special characteristics of smartwatches. In addition, we design and implement a usability framework that facilitates the automated usability analysis for smartwatch applications in a design science research approach. Furthermore, we demonstrate the applicability of the developed framework and show the results of a usability analysis for an exemplary case study.

1. Introduction

In the domain of mobile devices, which have been dominated by smartphones in the last decade, a new category of devices evolved due to technological advances and the ongoing miniaturization of computing components: wearable computers. They are worn on the user's body [8, 35], are experiencing an immense upswing and promise an improved human-computer interaction due to ubiquitous and non-disruptive access to information [44]. Examples are clothes integrating digital systems, smartglasses, and smartwatches [38].

The continuous increase in sales of wearable computers is significantly driven by digital wristwatches, which are forecasted to account for 64 % of total sales of wearables in 2022 [20]. One reason for this can be found in the public acceptance of these devices caused by the familiarity of watches and the

experience of well-being while wearing it. Nevertheless, smartwatch applications have to offer additional value and have to fit into a user's everyday life seamlessly.

Thus, the usability of smartwatch applications is an important success factor as it facilitates the efficient and effective use of an application. Typically, consumers obtain their applications from app-stores like *Google Play* or *Apple App Store* and can choose from a broad range of software products that differ in their functionality and design. In many cases, there are multiple providers for an application with similar functionalities. Users tend to prefer applications that provide the best usability, since those applications can solve the particular problem in an easily learnable and effective way, which reduces their cognitive load [4, 5]. Hence, considering usability becomes an economic factor for software developers. In the corporate context employees usually cannot choose their favorite application, since the selection is rather done by the employer. Thus, companies have to make sure, that the provided software-tools are appropriate. It should be easy for employees to learn the operation of an application in order to avoid a first barrier. A key factor is the high efficiency of an application. Employees should have fast access to particular functionality without taking unnecessary thoughts and paths, which makes it possible for them to focus on their proper work and save time. Furthermore, well-designed applications can facilitate to avoid mistakes and support employees within their working tasks. Finally, usability is strongly connected to acceptance as it is proposed by the Technology Acceptance Model [12] and weakly designed software can lead to a lack of motivation, fears, and denial of systems [1]. Since employees are an important economic factor, companies can benefit from investing in the design of their software and taking usability into account.

However, the small form factor and novel operating concepts of smartwatches introduce a series of new challenges and unique requirements. Usability, in particular, poses a challenge, because interaction primarily takes place on the small touch-sensitive screens [21]. The dimensions of a wristwatch, make user input more error-prone and the input of text seems

impracticable [10]. In addition, the heterogeneity of smartwatch-devices including different forms (e.g., round or squared), operating systems and hardware buttons necessitate a holistic view on usability analysis.

Gaining knowledge about the usability of smartwatch applications is of immense importance for research and practice. For research, it forms the theoretical foundation for the design of future concepts and possible solutions. For practice, it is possible to create applications and devices with a high level of satisfaction and to conquer market shares.

In order to develop a usability-framework for smartwatches, we apply a design science approach [27] in this paper. We propose a research design strongly inspired by Peffers et al. [34] including the problem identification, the deduction of objectives, the design process, and finally the demonstration and evaluation in order to design a usability framework for smartwatch applications. Overall we address the following research questions:

RQ1: Which requirements arise during the analysis of usability for smartwatch applications?

RQ2: How can existing methods be implemented in a framework to analyze the usability of smartwatch applications automatically?

To answer these research questions, the remainder of this article is structured as follows: First we present definitions of basic terms introducing the domain of smartwatch applications and usability and outline related research in section 2. Second, we describe our research method based on the design science research framework of Peffers et al. [34] in section 3. By applying the research framework to our problem, we illustrate the results of our design science approach in section 4. Finally, we discuss our findings and outline our research contributions for theory and practice in section 5.

2. Theoretical Foundation and Related Research

Since literature has not focused on usability analysis for smartwatches so far, we survey recent approaches and techniques targeted at mobile systems to gain a holistic view, build a foundation for further considerations and transfer the results to smartwatches. First, we provide definitions for the basic terms and then present the related research.

For a first containment and delimitation of our examination, we sharpen the range of the considered devices. Mobile devices are designed for mobile use and are characterized by high independence of physical locations, accessibility and localizability [13]. The devices natively provide connectivity over wireless

technologies and are driven by operating systems, which can be extended as required with additional installable and executable applications [22]. The span of devices ranges from smartphones and tablets to wearable computers like smartwatches. Mobile applications are special application programs that are designed to run on a mobile device, covering the special characteristics of mobile devices [29]. A smartwatch is a digital wristwatch extended by a touch screen and other common computer hardware components, such as a processor, working memory and battery. In addition, smartwatches provide a wide range of sensors and wireless technologies such as Near Field Communication (NFC), Global Positioning System (GPS) or Bluetooth as well as a microphone. The interaction with a smartwatch can be done with hardware components, such as the touch screen, buttons, voice control, or a coupled smartphone. Furthermore, smartwatches are equipped with a hardware-independent operating system, which can be executed on different devices, and delimit from other similar devices through the ability to install and execute additional software applications. Not all digital wristwatches, e.g., fitness tracker, meet these criteria and can rather be considered as featurewatches (c.f. featurephones [22]) that provide simple interaction through the coupling with a smartphone [30] and wireless interfaces. The implementation of applications for smartwatches depends on the platform and the operating system and is primarily done natively and fully independent of a smartphone in the platform-specific programming languages (e.g., Java) and the operating system's own Software Development Kit (SDK) accessing the platform-specific hardware and software components over the application programming interface (API).

The user-friendliness or usability of an application can be considered as a quality feature of a product and is defined as intuitive access to the operation of a product in order to accomplish a specific task. Usability is thus understood as a pragmatic quality of software in terms of achievement of objectives. Usability is defined according to ISO 9241-11 (2018) as the product of (1) effectiveness in the sense of usability for the fulfillment of tasks, (2) efficiency as a measure of the time and effort required to fulfill tasks, and (3) satisfaction as a measure for the positive attitude towards the use of the product in a particular context. It has to be distinguished from user experience, which is the users' perception of a system in consideration of the expected utility. In addition, Nielsen [32] considers the following criteria to play an important role in usability: (1) learnability - how easy can a user learn the operation of an application, (2) memorability - how good can a user operate an application after a certain amount of time without use,

and (3) error frequency - how many errors does a user provoke, how serious are these errors and how easily the user can find a solution to resolve the problem.

The mentioned usability attributes can be assigned to the People at the Centre of Mobile Application Development (PACMAD) model [17]. The PACMAD model focuses on the usability of a mobile application and identifies the user, the task, and the context as the primary influencing factors for usability. The context got a special role, as the applications are used in different contexts under various influencing conditions. With reference to smartwatches, this factor gets even more important, since the devices, concerning their form factor, are used in highly dynamic contexts. Due to this high mobility including simultaneous or interfering activities and environmental influences, not the full cognitive attention of a user can be presumed as in traditional usability investigations of desktop applications. For this reason, PACMAD uses the cognitive load which is necessitated by an application as a core usability attribute [17].

The term evaluation is generally used to describe a structured and objective evaluation of an object of investigation. A usability problem can be defined as a problem that a user encounters when using the system to complete a task within an application scenario [3]. A usage problem is attributed to a usability defect arising due to a violation of a usability principle and can have negative consequences for the user [28]. For the early detection of problems and thus avoidance and limitation of the negative consequences, usability evaluation methods are used. The methods can be classified into qualitative methods producing data, which has to be interpreted (testing, observing and questioning), and quantitative methods, which are based on defined metrics having numerical and objective data as a result (simulation and analytical modeling) [19]. For qualitative methods, moderated method types with little automation are common, such as the observation and recording, interviews, think-aloud protocols or heuristic methods. For quantitative methods in practice, unmoderated method types are frequently used, such as online questionnaires based on the usability scale system [39], the automated metric recording of an object of investigation or a task model [32].

The methods are used in various test environments, which is one influencing factor in the four-factor framework of contextual fidelity that describes the quality of the results of a usability evaluation [37]. Accordingly, the test environment has to resemble the actual operational environment, in order to avoid a negative impact on the quality. The laboratory test is one of the most frequently used test environments [22] since it takes place in a controlled and open definable context almost free of accidental environmental influences. This

allows to collect data through a variety of instruments during a moderated evaluation, which is highly specified and consequently exactly reproducible. Due to the versatile use cases of a smartwatch, the simulation of the particular environment in a laboratory test is a considerable challenge [43]. The research on automated usability measurement of smartwatches is still in its infancy. Recent methods split into static analysis, evaluating the source code and especially the design files during the development, and dynamic analysis considering user interactions. With reference to the previous remarks, the focus of this work are quantitative and automated usability evaluation methods.

Besides the theory about usability, there is related research especially in the domain of mobile and web applications. Gossen et al. [15] have expanded qualitative usability analysis by including the results of search engines or social media. Harrison et al. [17] did an extensible literature review on the usability of mobile applications and demand a new usability model. Balagtas-Fernandez and Hussmann [7] propose a methodology and a framework to aid developers during the preparation of mobile systems for usability analysis. Ahmad et al. [2] evaluated the usability of smartphones with a usability testing approach considering *Android* and *iOS*. Lettner and Holzmann [24] developed an automated and unsupervised system for usability evaluation by user interaction logging. Furthermore, there are the *HUI Analyzer* of Baker et al. [6], the *EvaHelper* framework [7] and the *toolkit for usability testing* of Ma et al. [25]. A number of studies cover logging on websites like Grigera et al. [16] who used usability smells to automatically generate a usability report. Beyond the scientific work, there are several commercial products, such as *Google Analytics*, *Flurry Analytics*, *Localytics* or *User Metrix*, which allow the user logging on native and web-based applications.

In the domain of smartwatches, initial efforts arose in the last couple of years. Chun et al. [11] conducted a qualitative study to access the usage and usability of smartwatches and elaborated guidelines for future smartwatches. Park et al. [33] examined different types of menu interfaces for smartwatch applications in a qualitative study. Finally, Wong et al. [41] considered the usability of smartwatches used for cheating in academic examinations.

3. Research Design

To target the research gap regarding the dynamic usability analysis of smartwatch applications, we applied a mixed-methods approach based on the problem-centered design science research process model by Peffers et al. [34] as shown in Figure 1. According to the process model, the development of the usability framework should be grounded in the problem

identification phase (step 1). To this aim, we rely on a structured literature review following vom Brocke et al. [9]. The main goal of this literature review is to gain a holistic view of recent approaches to usability analysis on mobile devices. This builds the foundation for an investigation of eligibility and possible adaption in order to apply these methods on smartwatch applications considering the device-specific characteristics. With these characteristics, we can infer objectives and requirements for the framework design and development (step 2). Following the design science research process model, we implemented a prototypical framework called *usabilityWatch* based on the requirements (step 3). Subsequently, we did a demonstration and evaluation according to Peffers et al. in step 4. For this, we integrated the *usabilityWatch* framework into an exemplary smartwatch application and conducted a laboratory study. We asked the participants to perform a task within a given scenario using a smartwatch application that supports employees in workflows. During the task, multiple paths and UI-elements have to be used and usability-events are logged by the framework. Finally, the gathered data can be analyzed to access usability-insights.

4. *usabilityWatch* Framework

In this section, we present the design of the usability framework *usabilityWatch*, which addresses the identified research gap. It simplifies the typical set of tasks for usability evaluation conducted by a developer including the preparation of a targeted application and the test environment, the data collection, the extraction of information and the data analysis [7].

4.1. Problem Identification

Based on the structured literature review, we identified a lot of research regarding usability for mobile information systems (see section 2). But so far there is little effort to analyze usability on smartwatches. Certainly, most qualitative methods, e.g. laboratory tests, can be applied to smartwatches as well. Since, 60 % of software problems are associated with the graphical user interface, which though in 5 % lead to a system crash, but have a negative effect on usage in 65 % [36], the users' behavior can reveal most of the

usability defects. However, there are no approaches to automatically and dynamically assess usability by analyzing the users' interaction with the application considering the special characteristics of smartwatches.

4.2. Objectives of a solution

In order to address the first research question (RQ 1), the existing literature is analyzed for requirements for the automated measurement of usability on mobile devices. From more than 40 occurring requirements we elaborated seven requirements for our usability framework by selection and adoption in regard to smartwatches. We structured these into the domains data collection and data analysis (see Table 1).

Our aim is to implement a framework, that provides a dynamic usability analysis. Although in *Wear OS* development structured layout files (XML) exist, which can be analyzed statically beforehand, we focus on the direct user interaction due to the highly restricted range of input elements on smartwatches. The static analysis does not offer a substitute for insights from the actual use of an application by the user captured by defined metrics [6] and depends strongly on the target device size and form factor. In order to determine the actual use of an application in the context of dynamic analysis, the recording of user interactions is a core functionality (R₁) [40]. The degree of automation should, as far as possible and reasonable, be considered [6] and the evaluation should be transparent for the user and has not to interfere or disturb normal use [31]. For the data collection, the framework has to provide appropriate metrics (R₂) that can provide measurements, e.g., a swipe-to-touch ratio or dwell times, based on the recorded data. They have to be selected for the special characteristics of smartwatches as small display sizes and a broad range of hardware. The metrics should be tailored for the interest groups of the evaluation results in order to provide them with easy access to the necessary information. Overall, the framework should be designed for simple integration in existing smartwatch applications without a high programming effort (R₃). Since laboratory environments compromise the detection of usability defects due to an unrealistic situation, the framework should be robust, inconspicuous and therefore usable in real application

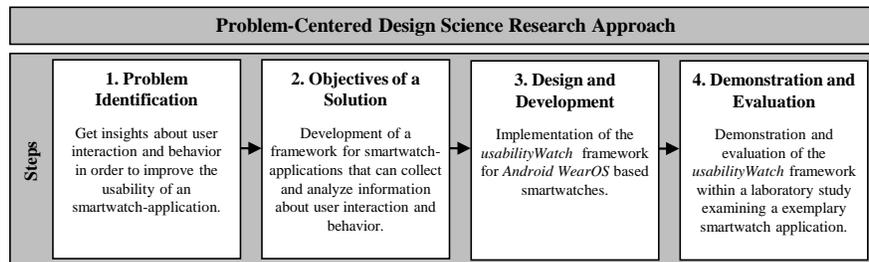


Figure 1. Research design adapted from Peffers et al. [34]

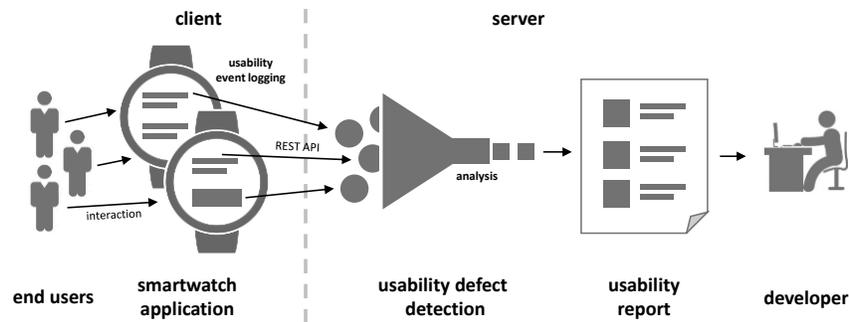


Figure 2. *usabilityWatch* architecture

environments [31]. Due to the high level of miniaturization, the limited computing and battery capacity get into the focus [23]. In addition, the connectivity of a smartwatch to wide area networks cannot be assured for any point in time. Furthermore, the available transfer volume of data is only seldom unlimited and should, therefore, be taken into account. Thus, the framework has to provide solid data transfer (R_4). The purpose of data analysis is to draw conclusions. For that in a first step, data segmentation is required (R_5) facilitating to view and compare the data in different dimensions [18]. In order to meet the changing demands of evaluation, a flexible and modular architecture is necessary [31]. Furthermore, it should be possible to process and analyze the collected data using appropriate methods (R_6) [40]. Since data collection can get extensive over time and scales with the number of users, computationally involving operations have to be handled in a way that does not exhaust hardware capacities of smartwatches. Finally, usability defects should be derived from the prepared data (R_7), which makes it possible to improve a smartwatch application due to these insights [18].

4.3. Design and Development

To meet the elaborated objectives, we designed and developed the usability framework for smartwatch applications *usabilityWatch*. The overall architecture (illustrated in Figure 2) is split into a smartwatch component, that is integrated into a targeted *Wear OS* (previously *Android Wear*) smartwatch application and a server component that gathers the arising data and provides usability reports to the developer. This architecture enables us to utilize the smartwatch for

direct data collection observing the behavior of the user and overcome device limitations for a decent data analysis due to higher computing capacities provided by a server. In reference to R_5 , data should be visualized on an appropriate screen size, which is not the case with a smartwatch. Furthermore, regarding R_6 , it exceeds the computing power of a smartwatch to process large amounts of data. Anyway, a server is required to gather the data from multiple devices and users.

In the domain of the smartwatch application of interest, we provide a lightweight framework component, which in reference to R_3 can be easily integrated by including and compiling the framework's Java package into the application's main activity. It seamlessly hooks into the required event handlers, overloads non-invasively application methods and implements the usability event logging as well as the communication to the server component. To access the full potential of the framework, the integration can benefit from aspect-oriented programming, e.g., *AspectJ*, which increases modularity, full separation of the frameworks and the application code and weaves the framework functions into the desired event listeners during the build process [14]. Besides the wireless connection to the server, the framework does not require more effort to implement and it is completely invisible to the user and does not interfere with the normal usage since it runs in the background within a separate thread. As nowadays wireless network access is ubiquitous and already constitutes a prerequisite for many smartwatch applications, the framework can be applied in a broad range of environments.

In order to capture significant usability events from the interaction of a user with the smartwatch application

Table 1. Requirements for data collection and analysis

data collection		data analysis	
R₁	automated recording of user inputs and interactions	R₅	flexible data segmentation and visualization on a decent screen size
R₂	tester-oriented usability metrics handling the broad range of hardware and display resolutions of smartwatches	R₆	evaluation methods for a large amount of data with decent processing capabilities
R₃	simple integration in existing smartwatch applications to collect data within real application environments	R₇	usability-defect analysis
R₄	solid data transfer in spite of limited connectivity and power		

to be examined and to meet R_1 , *usabilityWatch* automatically logs the mayor issues occurring on a smartwatch. This includes (1) touch events (cf. clicks), (2) swipes (cf. scrolling) and (3) navigation events (changing the context of the screen). Since other components are mostly used to call operating system functions or other applications, e.g., a voice assistant, which interrupts the use of the targeted application, we neither consider interactions using hardware buttons due to the large heterogeneity of hardware devices providing a broad range of different numbers of buttons equipped with different functions nor touch gestures which are differently assigned for every underlying operating system. We consider R_2 by capturing metadata for all usability events outlined above. These are timestamps for all events, the coordinates for touch events, the start and end coordinates for swipe events and a screenshot after navigation events. This also contains information about the UI elements that were interacted with and information about the device as the screen size as well as the form factor. In the analysis phase, the data can be combined in different ways to obtain usability insights.

For smartwatches, persistent network access cannot be assumed due to possible poor wireless coverage or overload, and transmissions reduce the limited power of smartwatch devices. To address R_4 the framework first stores occurring usability events internally. Occasional, this buffer is automatically sent to the server. If an error occurs this is repeated until a connection is available and the server consequently returns successfully. For the communication, we implemented a REST interface [26] which is easy to use, fast, reliable and incorporates security aspects by using HTTPS.

For the server component, we use the combination of PHP and a relational MySQL database to benefit from their abilities related to web applications. In this way, we provide a desktop backend that is empowered with modern web technologies like HTML5 and makes it easy for developers to configure and access the usability analysis. As presented in Figure 3 *usabilityWatch* provides five main sections that can be accessed over the menu. First, there is a *Dashboard* that gives an overview including important key figures. Furthermore, it surveys how many users and sessions for each tracked application have already been recorded. Second, in the *Application* section smartwatch applications can be added, configured and removed. Only data of registered applications are recorded, other requests are being rejected. In addition, the overall behavior of the REST interface can be configured in the API section.

In order to address R_5 , we implemented the *Session* section (depicted in Figure 3) which provides data segmentation over sessions and different dimensions as well as various visualizations of the recorded data. On the left side panel, *usabilityWatch* provides a comprehensive timeline that visualizes all events of a selected session. User interactions like touch and swipe events are illustrated in blue, a particular icon and show their coordinates of occurrence. Navigation events, which can be the result of a touch or are triggered by the smartwatch application, are illustrated in orange and respectively show the name of the reached screen. In addition, the navigation paths can be investigated with a Sankey diagram. The upper right side panel shows heatmaps that aggregate all touch (left) and swipe (right) events which can be segmented by the corresponding

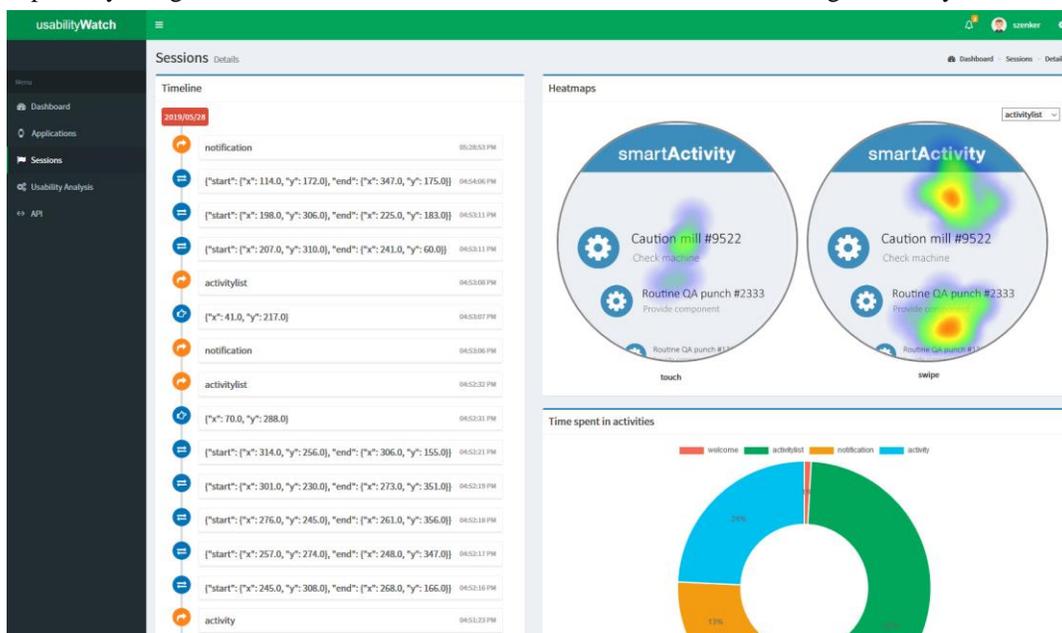


Figure 3. *usabilityWatch* session analysis

screen name. Areas of the screen, which show a high number of interactions, are dyed red, areas with low interaction are dyed blue. Since *usabilityWatch* captures screenshots, these heat maps can overlay the visible contents to facilitate the interpretation of this visualization. On the lower right panel, the relative distribution of dwell times is shown in a doughnut chart. It illustrates how much time a user stayed on a certain screen which is the time difference between two subsequent navigation events.

Finally, R_7 is implemented in the Usability Analysis section. Here the data is analyzed with a holistic view in order to generate insights into usability-defects. We elaborated and implemented several usability smells made for the specific needs of smartwatch applications. These can identify evidence for usability-defects, which are attributed to a violation of a usability-guideline leading to a problem for the user, by a specific pattern of usability events in the collected data. Similar to [16], we list the usability smells, anomalies of events and suggested refactoring that we derived from our previous studies in Table 2. To some extent similar smells in different contexts were also identified in the literature (like *unresponsive element* and *distant content* in web-applications [16]). The *unresponsive element* smell occurs whenever a user attempts to touch on an element, that does not respond to touch events. This happens when elements look like buttons but they are not. The smell can be detected by scanning for touch attempts that do not have a subsequent action. Similar to this smell *inappropriate swipe area* appears when user attempt to scroll on elements with a swipe gesture but the target is not able to scroll. This can happen if an element either does not support scrolling or the user started the swipe outside of the swipe area and can be identified by looking for swipe attempts without further action. Next, the framework provides the swipe-to-touch ratio metric. Looking at this value for each screen individually, the *incomprehensible list* smell can be detected if the value is unusually high. Ordinarily, a user

scrolls through a list and touches the element of interest. In the optimal case, the mentioned ratio is 1, because it needs one single swipe to locate the desired item and one touch to activate it. A high ratio indicates, that the user has to swipe a lot until the element is found. This happens for lists with many elements in an unfavorable order or a confusing list structure. The *missing confirmation* smell occurs when a touch to an element instantly leads to an influential action, e.g. change of data or the application state. If this is unintended by the user, the restoring action can be found in the logs. Slightly different is the *missing feedback* smell. Here the user tends to check a change of data or an application state due to missing feedback subsequent to an action. Loops in the navigation path can reveal this in the data. Next, the *missing processing indicator* smell identifies computationally involving actions which block the UI for a time. For users, it is confusing if the application is not responding and they start to touch somewhere. To avoid that, a processing indicator can clarify that actually an action is performed and the user has to wait. Finally, there is the *distant content* smell that occurs for unnecessarily complicated navigation. A user has to navigate through several screens until the targeted content is arrived. If repeating navigation patterns without any other interaction on the screens in between are detected in the data, a direct navigation element can facilitate the user to use the application more effectively. Ultimately, since the analysis of the huge amount of data is done on the server-side R_6 is met as well.

4.4. Demonstration and Evaluation

For demonstration and evaluation, we conducted a laboratory study with 12 participants. We implemented the *usabilityWatch* framework in the exemplary smartwatch application *smartActivity* which provides collaborative support for employees in industrial workflows [42]. For that, an employee can receive, process and return activities according to a defined workflow. The application is composed of four screens:

Table 2. Usability smells with the associated usability events and recommended refactoring

usability smell	usability events	refactoring
Unresponsive element	touch attempt on an element without any subsequent action	change UI appearance or add functionality to the element
Inappropriate swipe area	swipe attempt on an element without any subsequent reaction	change UI appearance, add UI interaction to the element or increase and highlight swipe area
Incomprehensible list	high swipe-to-touch ratio on a list	increase size of list widget, revise sorting or reduce number of elements
Missing confirmation	repeating action while restoring the previous state	add confirmation prompt before action execution
Missing feedback	repeating loops in navigation path pattern	add visual feedback when the action was performed
Missing processing indicator	long request delays navigation after button touch	add processing indicator
Distant content	repeating navigation patterns without non-navigation touch and swipe interaction in between	add direct navigation element

(1) a welcome page at the start of the application (*welcome*), (2) a list of assigned activities as illustrated in Figure 3 (*activitylist*), (3) a notification screen that informs a user about incoming activities that can be accepted or postponed (*notification*) and (4) a detail screen for a selected activity with a list of possible next steps according the workflow as illustrated in Figure 5 (*activity*). During the study, the participants took the role of a technician who is responsible for several computer-operated milling and punching machines and traverse a scenario including various machine alerts and requests of a quality assurance department. After the scenario was completed, they were asked to fill in a predominantly qualitative questionnaire in order to evaluate the overall usability and usability problems occurred during the operation of the smartwatch application. This enables us (1) to collect and analyze realistic data with *usabilityWatch* and (2) have insights about the usability problems of real users. Matching both assessments allows us to evaluate the utility of the developed framework.

After conducting the laboratory study, we asked the participants to provide us feedback about usability. On the one side, the participants highlighted several positive aspects regarding usability like the clear arrangement of the application, intuitive use, a low number of touches to process activities and fast loading times, on the other side several problems were stated. Concerning lists, the participants mentioned, “*the overview of activities automatically jumps up again very quickly, which makes the selection difficult*” (participants 5 and 6, 8, 9, 10) and “*the selection of the possible next work steps on the detail screen is very small*” (participants 5 and 2). Both comments reveal serious problems since the list at the *activitylist* screen jumps to the top every five seconds whenever the list is updated due to a messy implementation, which disturbs the selection of the desired element and requires another swipe. *usabilityWatch* detects both problems utilizing the swipe-to-touch metric shown in Figure 4.

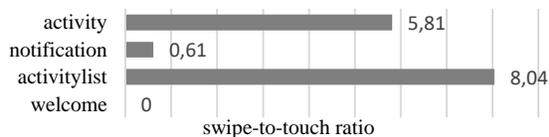


Figure 4. Swipe-to-touch ratio for the different screens

The swipe-to-touch ratio outlines a very high value for *activitylist* indicating that for each selection many swipes are required. The list at the *activity* screen also triggers a high value that is more related to the small size which can be proved by the high number of unsuccessful swipes in the vicinity of the list. The *incomprehensible list* and *inappropriate swipe area* smells are reported accordingly since the optimal sequence is to swipe to the element and touch it resulting in a value of 1. Another

issue is described as “*the back button was only half displayed and therefore hard to reach*” (participants 4 and 5, 6, 7, 8). This can easily be seen in Figure 5 and is caused by an unintended shift of the whole layout of *smartActivity* to the bottom (small white area at the top). *usabilityWatch* reports the *unresponsive element* smell for touches close to the button. In combination with the heat map given in Figure 5, this issue can be detected.



Figure 5. Touch heat map of activity screen

As last commonly listed usability problem we got “*faulty touches quickly lead to unwanted entries*” (participants 4 and 2) and “*I like to have more feedback that an action was executed after I touched a button*” (participants 3 and 8). So far there is neither clear feedback that an action succeeded nor a confirmation prompt if an action should be performed. This leads to user behavior in which the action is checked or restored subsequently. The framework reports the *missing feedback* and *missing confirmation* smell due to a looping index of 3.2 and 2.7 respectively.

Summarizing, *usabilityWatch* can identify the reported usability problems within the recorded data. Some of the defects can be found completely automatically, for others the usability smells are just an indication and have to be combined with other (visual) metrics to conclude the defect.

5. Discussion and Conclusion

In this paper, we presented a usability framework for smartwatches. Inspired by the design science research method [34], we illustrated a problem-orientated research design. We first identified and described usability methods which are recently used for mobile devices (RQ1), since the usability analysis of smartwatches is a research gap. We formulated objectives and inferred requirements based on the conducted structured literature review and considered the unique characteristics of smartwatches. We presented the *usabilityWatch* framework composed of a smartwatch component, and web backend (RQ2). It provides easy integration into a smartwatch *Wear OS* application, automated logging of user interactions, visualization of the collected data with, e.g., heat maps and the analysis of usability defects. For that, we elaborated a list of usability smells suited for smartwatches. Finally, we proved in a demonstration

and evaluation that the framework can find similar usability defects as the participants of a laboratory study for an exemplary smartwatch application.

There are some limitations to our research study. Since usability is a well-researched topic the related literature is extensible and we cannot claim our review to be complete. Second, we tested the framework with just one exemplary smartwatch application within an exemplary scenario. We are planning to do tests with more applications in order to improve the modularity and simplicity of integration of the framework. Furthermore, we want to extend the list of usability smells and like to optimize the thresholds for the existing smell metrics towards realistic values by expanding the practice. Though, the application of the framework requires a proper interpretation of the results in order to benefit of the generated insights and to identify false positives that may occur in the automated analysis. In addition, the user of the framework has to be aware of metrics like the swipe-to-touch ratio which can be misleading whenever multiple scrollable elements appear at the same screen (unlikely due to small screen size) or the screen itself can be scrolled.

6. References

- [1] Acton, T., S. Gudea, M. Scott, and W. Golden, Usability and acceptance in small-screen information systems, COLLECTeR (Europe), University of Surrey, Guildford, UK, 2004.
- [2] Ahmad, N., M.W. Boota, and A.H. Masoom, "Smart Phone Application Evaluation with Usability Testing Approach", *Journal of Software Engineering and Applications*, 07(12), 2014, pp. 1045–1054.
- [3] Alshamari, M. and P. Mayhew, "Technical Review: Current Issues of Usability Testing", *IETE Technical Review*, 26(6), 2009, p. 402.
- [4] Apitz, G., F. Guimbretière, and S. Zhai, "Foundations for designing and evaluating user interfaces based on the crossing paradigm", *ACM Transactions on Computer-Human Interaction*, 17(2), 2010, pp. 1–42.
- [5] Arning, K. and M. Ziefle, "Ask and You Will Receive", *International Journal of Mobile Human Computer Interaction*, 2(1), 2010, pp. 21–47.
- [6] Baker, S., F. Au, G. Dobbie, and I. Warren, "Automated Usability Testing Using HUI Analyzer", in 19th Australian Conference on Software Engineering: ASWEC. 2008. IEEE Computer Soc: Los Alamitos, Calif. USA.
- [7] Balagtas-Fernandez, F. and H. Hussmann, "A Methodology and Framework to Simplify Usability Analysis of Mobile Applications", in 24th IEEE/ACM International Conference on Automated Software Engineering: ASE '09, Auckland, New Zealand. 2009. IEEE: Piscataway, NJ.
- [8] Billinghurst, M. and T. Starner, "Wearable devices: new ways to manage information", *Computer*, 32(1), 1999, pp. 57–64.
- Since hardware buttons or digital crowns are noted as very pleasant, these should also be included in the corresponding scrolling metrics, which remains a complicated problem due to heterogeneous hardware and software widgets.
- Nevertheless, we verified the utility of *usabilityWatch* in a realistic scenario and contribute to practice and research. The developer of smartwatch applications can benefit from usability insights in order to reduce a user's cognitive load and to improve their applications. This can easily be done by analyzing the user's interactions and no time consuming and expensive qualitative studies like laboratory tests are required. For practice, we created an applicable software solution for targeting automatically usability analysis on smartwatch devices in order to support developers. Within the research domain, we reviewed recent approaches and methods, modified and complemented them according to the unique characteristics of smartwatches covering main aspects of the PACMAD model. This transfer of methods forms the foundation for future studies for usability analysis on smartwatches.
- [9] Brocke, J., A. Simons, B. Niehaves, B. Niehaves, K. Reimer, R. Plattfaut, and A. Clevén, "Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process", in *ECIS: Proceedings*. 2009.
- [10] Chaparro, B.S., J. He, C. Turner, and K. Turner, "Is Touch-Based Text Input Practical for a Smartwatch?", in *HCI International 2015 - posters' extended abstracts*, C. Stephanidis, Editor. 2015. Springer: Cham.
- [11] Chun, J., A. Dey, K. Lee, and S. Kim, "A qualitative study of smartwatch usage and its usability", *Human Factors and Ergonomics in Manufacturing & Service Industries*, 28(4), 2018, pp. 186–199.
- [12] Davis, F. d., "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology", *MIS Quarterly*, 13(3), 1989, p. 319.
- [13] Durlacher Research Ltd, "Mobile Commerce Report", in *Mobile Networking with WAP: The Ultimate Guide to the Efficient Use of Wireless Application Protocol*. 2000. Vieweg+Teubner Verlag: Wiesbaden.
- [14] Enriquez, J. and S. Casas, "Variability in usability tests for Android applications", in *Proceedings of the 2nd International Workshop on Variability and Complexity in Software Design*. 2017. IEEE Press: Buenos Aires, Argentina.
- [15] Fournay, A., R. Mann, and M. Terry, "What can internet search engines "suggest" about the usage and usability of popular desktop applications?", in *Perlin (Hg.) 2010 – Adjunct Proceedings of the 23rd*.
- [16] Grigera, J., A. Garrido, J.M. Rivero, and G. Rossi, "Automatic detection of usability smells in web applications", *International Journal of Human-Computer Studies*, 97, 2017, pp. 129–148.
- [17] Harrison, R., D. Flood, and D. Duce, "Usability of mobile applications: literature review and rationale for a

- new usability model", *Journal of Interaction Science*, 1(1), 2013, p. 1.
- [18] Heo, J., D.-H. Ham, S. Park, C. Song, and W.C. Yoon, "A framework for evaluating the usability of mobile phones based on multi-level, hierarchical model of usability factors", *Interacting with Computers*, 21(4), 2009, pp. 263–275.
- [19] Ivory, M.Y. and M.A. Hearst, "The state of the art in automating usability evaluation of user interfaces", *ACM Computing Surveys*, 33(4), 2001, pp. 470–516.
- [20] Jitesh Ubrani, Ramon Llamas, Michael Shirer, IDC Forecasts Slower Growth for Wearables in 2018 Before Ramping Up Again Through 2022, IDC Corporate USA, 2018.
- [21] Kim, J., J. He, K. Lyons, and T. Starner, "The Gesture Watch: A Wireless Contact-free Gesture based Wrist Interface", in 11th IEEE International Symposium on Wearable Computers. 2007. IEEE: Piscataway, N.J.
- [22] Kolbe, L.M. and T.J. Ruch, "Mobile Security: Herausforderungen neuer Geräte und neuer Nutzeransprüche", *HMD Praxis der Wirtschaftsinformatik*, 51(1), 2014, pp. 9–23.
- [23] Lee, J., J. Ko, and Y.-J. Choi, "Dhrystone million instructions per second–based task offloading from smartwatch to smartphone", *International Journal of Distributed Sensor Networks*, 13(11), 2017, 155014771774007.
- [24] Lettner, F. and C. Holzmann, "Automated and unsupervised user interaction logging as basis for usability evaluation of mobile applications", in *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia*, E. Pardede, Editor. 2012. ACM: New York, NY.
- [25] Ma, X., B. Yan, G. Chen, C. Zhang, K. Huang, J. Drury, and L. Wang, "Design and Implementation of a Toolkit for Usability Testing of Mobile Apps", *Mobile Networks and Applications*, 18(1), 2013, pp. 81–97.
- [26] Malik, S. and D.-H. Kim, "A comparison of RESTful vs. SOAP web services in actuator networks", in *Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, Milan. 2017. IEEE.
- [27] March and Storey, "Design Science in the Information Systems Discipline: An Introduction to the Special Issue on Design Science Research", *MIS Quarterly*, 32(4), 2008, p. 725.
- [28] Marcilly, R., E. Ammenwerth, F. Vasseur, E. Roehrer, and M.-C. Beuscart-Zéphir, "Usability flaws of medication-related alerting functions: A systematic qualitative review", *Journal of biomedical informatics*, 55, 2015, pp. 260–271.
- [29] Masi, E., G. Cantone, M. Mastrofini, G. Calavaro, and P. Subiaco, "Mobile Apps Development: A Framework for Technology Decision Making", in *Mobile Computing, Applications, and Services: MobiCASE 2012. Revised Selected Papers*. 2013. Springer Berlin Heidelberg.
- [30] McGrath, M.J. and C.N. Scanai, *Sensor Technologies: Healthcare, Wellness, and Environmental Applications*, Apress, Berkeley, CA, 2013.
- [31] Muhi, K., G. Szöke, L.J. Fülöp, R. Ferenc, and Á. Berger, "A Semi-automatic Usability Evaluation Framework", in *Computational Science and Its Applications - ICCSA 2013: 13th International Conference, Part II*. 2013. Springer: Berlin/Heidelberg.
- [32] Nielsen, J., *Usability engineering*, 4th edn., AP Professional, Boston, 1997.
- [33] Park, K., M. Jeong, and K. Kim, "Usability evaluation of menu interfaces for smartwatches", *Journal of Computer Information Systems*, 219(2934), 2018, pp. 1–10.
- [34] Peffers, K., T. Tuunanen, M.A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research", *Journal of Management Information Systems*, 24(3), 2007, pp. 45–77.
- [35] Rawassizadeh, R., B.A. Price, and M. Petre, "Wearables", *Communications of the ACM*, 58(1), 2014, pp. 45–47.
- [36] Robinson, B. and P. Brooks, "An Initial Study of Customer-Reported GUI Defects", in *International Conference on Software Testing, Verification, and Validation Workshops*, Denver, CO, USA. 2009. IEEE.
- [37] Sauer, J., A. Sonderegger, K. Heyden, J. Biller, J. Klotz, and A. Uebelbacher, "Extra-laboratorial usability tests: An empirical comparison of remote and classical field testing with lab testing", *Applied Ergonomics*, 74, 2019, pp. 85–96.
- [38] Seneviratne, S., Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, and A. Seneviratne, "A Survey of Wearable Devices and Challenges", *IEEE Communications Surveys & Tutorials*, 19(4), 2017, pp. 2573–2620.
- [39] Tullis, T. and B. Albert, *Measuring the user experience: Collecting, analyzing, and presenting usability metrics*, Elsevier/Morgan Kaufmann, Amsterdam, 2013.
- [40] Wetzlinger, W., D. Nedbal, A. Auinger, C. Grossauer, C. Holzmann, and F. Lettner, "Mobile Usability Testing Requirements and their Implementation in the Automation Engineering Industry", in *MoMM: The 12th International Conference on Advances in Mobile Computing and Multimedia*, Y.-H. Tao, Editor. 2014. ACM Press: New York, USA.
- [41] Wong, S., L. Yang, B. Riecke, E. Cramer, and C. Neustaedter, "Assessing the usability of smartwatches for academic cheating during exams", in *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '17*. 2017. ACM Press: New York, USA.
- [42] Zenker, S. and S. Hobert, "Design and implementation of a collaborative smartwatch application supporting employees in industrial workflows", in *Proceedings of the 27th European Conference on Information Systems (ECIS): Research Papers*. 2019.
- [43] Zhang, D. and B. Adipat, "Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications", *International Journal of Human-Computer Interaction*, 18(3), 2005, pp. 293–308.
- [44] Ziegler, J., S. Heinze, and L. Urbas, "The potential of smartwatches to support mobile industrial maintenance tasks", in *Proc. IEEE Conf. Emerging Technologies & Factory Automation (ETFA)*, Luxembourg. 2015. IEEE: Piscataway, NJ.