Mobile Usability Testing Requirements and their Implementation in the Automation Engineering Industry

Werner Wetzlinger, Dietmar Nedbal, Andreas Auinger
University of Applied Sciences Upper Austria
Wehrgrabengasse 1-3, Steyr, Austria
firstname.lastname@fh-steyr.at

Christian Grossauer, Clemens Holzmann, Florian Lettner
University of Applied Sciences Upper Austria
Softwarepark 11, Hagenberg, Austria
firstname.lastname@fh-hagenberg.at

ABSTRACT
Due to the massive increase in the number of mobile devices, high usage rates and the rising number of available applications, usability is a crucial factor for the success of mobile interactive systems. Since qualitative usability tests are quite expensive and time consuming we are developing a toolkit for remote usability testing that helps to detect usability problems without having to conduct traditional usability tests. Furthermore, the software should be widely applicable and not interfere with existing code. Based on this idea in this paper we (1) introduce our methodology to conduct qualitative requirements analysis, (2) show the results of a cross industry requirements analysis including the fields “m-commerce”, “information system development” and “automation engineering” and (3) present our approach to prioritize and implement the requirements of the automation engineering industry in a prototype.

Categories and Subject Descriptors
H.5.1. [Information Interfaces and Presentation (e.g. HCD)]: User Interfaces - Evaluation/methodology.

General Terms
Measurement, Design, Human Factors

Keywords
Usability testing; requirements analysis; interaction logging; handheld terminal; touch heatmap; navigation sequence visualization

1. INTRODUCTION
The market penetration of mobile consumer devices has matured in developed countries [8, 11, 23]. With the growing number of people using these devices the number of applications on the leading platforms (especially iOS and Android) also grew strongly. Consequently, many applications (“apps”) for similar purposes exist and functional differences are getting smaller. Therefore other distinguishing factors like price, design and usability become more important.

Additionally mobile devices are also getting more important in traditional industries. Smartphones and especially tablets are still primarily used as consumer devices [5], but their usage to carry out business tasks, as well as the development of industry specific mobile devices is increasing [9].

Since mobile devices are smaller, their screens display less content and different interaction paradigms are used. In addition, mobile apps are often used while moving around and interacting with the surrounding environment (e.g. while navigating). Thus, users are influenced by external stimuli [14]. The application design has to consider these influences and limitations [6] and has to support the user in carrying out tasks effectively, efficiently and with satisfaction in its current context [12].

To ensure the usability of applications and devices several methods have been developed [4]. Traditional usability testing methods with users in laboratory settings are very time consuming and expensive [13]. Furthermore, especially for mobile applications it is hard to obtain practically relevant results from laboratory evaluations, since they are based on a fixed situation that may only occur rarely in real world due to quickly changing and possibly strongly varying contexts [22]. Thus, choosing a usability analysis method for mobile devices should consider mobile contexts and real-world conditions which cannot be monitored in isolated usability laboratories without any influences or distractions.

Additionally, traditional field studies require up to twice the time and effort compared to laboratory studies, due to time-consuming tasks like preparation, pre-testing, supervision and analysis. In the case of mobile devices especially pre-tests and evaluations are considered critical, since changing contexts have to be considered and compared accordingly [15].

Remote evaluation of mobile applications through the automatic logging, analysis and visualization of user interaction data can identify a number of usability problems and therefore lower these costs [1, 2]. There are already several analytics tools that track user interactions on mobile platforms (Flurry Analytics, Google Mobile App Analytics, Apsalar, Localytics etc.) or websites and...
calculate corresponding metrics. But these tools are only available for websites and the predominantly used platforms of mobile consumer devices (iOS, Android, Blackberry, Windows Phone etc.) and not for platforms used in the automation engineering industry. Furthermore they focus on logging interaction data, measuring metrics and visualizing this data [25], but hardly consider the current context of use and therefore might miss underlying usability problems.

In addition, in unsupervised field tests, users are not guided and influenced by supervisors or assigned tasks. Therefore data is more representative, since data is based on the intentions of users and typically more user data is available for quantitative analysis. Furthermore, unsupervised field studies are cheap and do not require much preparation [21].

Therefore, we believe that especially in the case of mobile devices conducting unsupervised remote field studies will provide relevant data at a low cost. Furthermore, data can be analyzed at any given time without preparation. But existing tools do not support this kind of automatic usability evaluation [27].

Since there seems to be a lack of tools to evaluate usability of mobile applications, we decided to develop a toolkit to contribute to automatic and unsupervised usability evaluations. The toolkit should be capable of monitoring and analyzing interaction data as well as context data of users, calculating usage metrics, identifying usability problems and providing recommendations to aid developers and designers to improve mobile applications. As a result the toolkit should help to discover as many usability problems as possible, lowering the effort for subsequent traditional usability analysis. To ensure the toolkit can be used in a wide range of industries and devices, it should be as independent as possible from the underlying platform and no modifications of the analyzed application should be necessary.

Based on these goals in this paper we present the first steps of the development by (1) introducing our methodology to conduct a qualitative cross industry requirements analysis, (2) showing the results of our analysis including the fields “m-commerce”, “information system development” and “automation engineering” and (3) and presenting our approach to prioritize and implement the requirements of the automation engineering industry in a prototype.

The remainder of this paper is structured as follows: In section 2 we describe the methodology of the conducted empirical survey. In section 3 we present the results. This includes a cross industry analysis of the requirements, a prioritization of the requirements of the automation engineering industry and a description of the current state of a prototype we implemented for a project partner in the automation engineering industry. We close the paper with a discussion of the main findings and future work in section 4.

2. METHODOLOGY

Identifying user requirements is an important part in the design of every information system. Covering these requirements with corresponding software functionality is critical to the success of information systems. As, for example, discussed by Maguire and Bevan, there are several different methods to do this [19].

Therefore, choosing a particular data collection method has to be based on the particular situation. Within our research project, collecting requirements for an automatic and unsupervised usability evaluation of mobile devices in different industries was characterized by the following circumstances:

- Due to the different industries and companies, very diverse requirements were expected. Comparing and integrating diverse requirements is hard using a standardized method. Therefore semi-standardized and non-standardized methods seemed more appropriate.
- To ensure that the whole data collection process (e.g. cross industry interviews) covered the same topics at least a semi-structured method seemed necessary.
- Data collection methods in groups (e.g. focus groups) seemed too complex at this stage of the project.
- Oral data collection methods were considered more appropriate than written methods because complex requirements can be clarified more easily. Therefore, oral personal, face-to-face data collection methods were considered appropriate.
- The explorative setting of identifying requirements favors using open questions.

Based on these considerations semi-standardized, qualitative, oral, face-to-face interviews with experts in the field using open questions were used. We developed an interview guide consisting of 23 questions that covered the main topics of the ISO 4291 usability framework [12] tailored to our individual requirements:

- Context of use
- Usability measures/metrics
- Evaluation methods
- Data presentation

2.1 Participants

We wanted the resulting toolkit to be applicable in different contexts and industries. Since mobile devices are widely used in the consumer market we included the industries software development and M-Commerce in the study. As traditional industry we additionally included the automation engineering industry, where mobile devices are used for very specific tasks (e.g. to control manufacturing lines) and have a different designs than consumer devices. Thus, we hoped that the requirements of this field can help us ensure the toolkit can be used for different devices and in diverse contexts.

Overall 7 interviews were conducted. As indicated in Table 1 the survey covered 3 interviews with the automation industry, 3 interviews with the software development industry, and 1 interview with an m-commerce development company. Four were individual interviews and in three cases two experts were interviewed at once. This happened on request of the companies.

We also wanted to include different perspectives on usability and asked the companies to provide participants that can cover the software developer perspective with engineering focus as well as participants with a product manager perspective having a business and user focus. The headquarters of all companies were located in Upper Austria. Their business operating areas varied from local to global. Since all companies were participants of our research project, the willingness to take part in the survey was high, because identified requirements were included in the development plan of the toolkit.
2.2 Interview Settings

All interviews took place in the participating companies and were conducted by two researchers to ensure a consistent quality. One person was leading the interview by asking questions, ensuring all questions of the interview guide were covered.

The other person (“co-interviewer”) was taking notes, comparing answers with previous interviews and asking additional questions to ensure all topics were covered consistently across the interviews. Furthermore, the interviews were digitally recorded for subsequent checks of consistency and completeness. All nine interviews took place between November and December 2013 and lasted from 45 to 120 minutes.

2.3 Data Analysis

Considering the above mentioned complex situation of incorporating statements of experts from different companies and industries in one integrated software requirements catalog, a combination of qualitative and quantitative methods was used. In a first step the interviews were analyzed using qualitative data analysis methods to identify requirements. In a second step the number of references of these requirements was evaluated to determine their importance.

The qualitative identification of requirements was based on a triangulation of methods from Lamnek [17], Mayring [20] and Bortz/Döring [3] (see Figure 1).

**Transcription:** The 1:1 transcription of interviews was omitted, since no additional benefit to the audio recordings could be identified. The identification of possible hidden or suppressed opinions or attitudes like in other social studies was not considered important in this context.

**Single Interview Analysis:** For every interview the following steps were executed.

- **Paraphrasing:** Important statements were identified, rephrased in short concise sentences and exact moment in the audio file was referenced to ensure all requirements can be traced back to interview statements.
- **Generalization of Statements:** Statements were generalized to a common level of abstraction to be able to compare them. This was done be using nouns and verbs consistently over all statements and identify certain topics the statements were directed at.
- **First Reduction:** Relevant content was identified, by eliminating all statements that could not be classified as requirements (e.g. general statements about experiences, recommendations etc.).

**Cross Interview Analysis:** This analysis integrated statements of all conducted interviews.

- **Second Reduction:** The second reduction compared and merged the statements of all interviews. This was done by grouping all statements into certain topics using an iterative process of combining, eliminating and structuring the topics and requirements. Requirements that were mentioned by multiple interviewees were reduced to one requirement, leading to unique requirements for the development process of the toolkit. Furthermore these topics were associated with one of the areas of the interview guide (context of use, evaluation methods, metrics, and data representation).

- **Encoding:** To guaranty the requirements can be compared easily a consistent phrasing was used that ensured all requirements expressed a desired functionality or quality of the system.

**Control and Review:** It was checked whether after the two reductions the resulting requirements still represented the initial statement. This was done by comparing them with the corresponding statements in the recorded audio files using the time references of the paraphrasing phase.

---

**Table 1: Conducted interviews**

<table>
<thead>
<tr>
<th>Industry</th>
<th>Type</th>
<th>Business Focus</th>
<th>Interviewees</th>
<th>Focus of Interviewee</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Software Development</td>
<td>SME</td>
<td>Local</td>
<td>2</td>
<td>1 Product Owner</td>
<td>75 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 Software Developer</td>
<td></td>
</tr>
<tr>
<td>2 Software Development</td>
<td>SME</td>
<td>Local</td>
<td>1</td>
<td>1 Product Owner</td>
<td>75 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 Software Developer</td>
<td></td>
</tr>
<tr>
<td>3 Software Development</td>
<td>SME</td>
<td>Global</td>
<td>2</td>
<td>1 Product Owner</td>
<td>45 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 Software Developer</td>
<td></td>
</tr>
<tr>
<td>4 M-Commerce</td>
<td>SME</td>
<td>Global</td>
<td>2</td>
<td>1 Product Owner</td>
<td>90 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 Software Developer</td>
<td></td>
</tr>
<tr>
<td>5 Automation Engineering</td>
<td>Large Company</td>
<td>Global</td>
<td>1</td>
<td>1 Product Owner</td>
<td>45 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 Software Developer</td>
<td></td>
</tr>
<tr>
<td>6 Automation Engineering</td>
<td>Large Company</td>
<td>Global</td>
<td>1</td>
<td>1 Product Owner</td>
<td>45 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 Software Developer</td>
<td></td>
</tr>
<tr>
<td>7 Automation Engineering</td>
<td>Large Company</td>
<td>Global</td>
<td>1</td>
<td>1 Product Owner</td>
<td>120 min</td>
</tr>
</tbody>
</table>

**Figure 1: Triangulation of qualitative data analysis methods**

The overall analysis was based on the 4-phases model of Lamnek [17]. Since this model only provides basic guidelines for the important phases 2 (single interview analysis) and 3 (cross interview analysis), we extended these phases using methods from Mayring [20] and Bortz/Döring [3] the following way:
These steps led to a hierarchy where one area includes a number of topics and a topic includes multiple requirements. This structure formed the basis of the quantitative frequency analysis.

This data analysis was carried out using the software ATLAS.ti (atlasti.com), which is specifically designed for qualitative data analysis and provides a number of functions to import multiple sources (text, audio and video), mark certain sections with codes, classify codes and perform quantitative analyses.

2.4 Frequency Analysis

Besides identifying all requirements, their relative importance was also of interest to ensure the development of software functionality was prioritized accordingly. To determine the importance a quantitative content analysis was used. Frequency analysis is considered an appropriate method to quantify qualitative results and create a ranking of relevant statements. The analysis considers a statement to be more important, if it is made more often. Therefore every statement gets classified and associated with a certain topic. For every topic the number of statements is calculated to show its importance [7].

In the case of our study the importance was not determined based on topics (see first reduction) and the number of requirements in these topics (second reduction). The more unique requirements per topic could be identified the more important it was considered. Consequently we ranked the topics by

- **Number of requirements in a topic**: The more requirements for a topic could be identified, the more important it was considered.
- **Number of companies with requirements for a topic**: The more companies had requirements concerning a certain topic, the more important it was considered, because this indicated a wide adoption of the corresponding functionality.
- **Number of industries with requirements for a topic**: The more industries had requirements concerning a certain topic, the more important it was considered, because this indicated a broader adoption of the corresponding functionality.

3. RESULTS

In this section we describe the results of our research. We start by comparing all participating industries to show their differing focus (3.1). We then present the automation engineering prototype (3.2) by describing the prioritization of requirements and showing the current state of the prototype implementation.

3.1 Cross Industry Analysis

The comparison of industries followed the above mentioned hierarchical assignment of requirements to topics and topics to the interview guide areas.

The number of identified statements and requirements differed between the industries (Figure 2). But these absolute quantities should be considered with care, since they are related to the length of the interviews, their intensity and the interviewed person.

Since some requirements were mentioned by more than one industry, they were consolidated into a single requirement (second reduction). This led to a reduction from 267 requirements (see Figure 2) to 222 unique requirements for the development of the toolkit.

![Figure 2: Number of statements and requirements](Image)

**Figure 2: Number of statements and requirements**

Since a subsequent quantitative analysis of requirements per topic is highly dependable on the granularity of the topics, we paid special attention to not introduce topics of equal granularity. Additionally every requirement was assigned to only one topic.

Figure 3 shows the most important topics based on the total number of requirements per industry. It incorporates the metrics **number of requirements in a topic and number of industries with requirements for a topic**. Table 2 further elaborates on this by showing the top 10 requirements for every industry. This diminishes problem that the absolute number of requirements is related to the length of the interviews, their intensity and the interviewed person and shows the relative importance of the requirements for the industries. From these cross industry analysis the following key points were derived:

- The **usage intensity** is the most important metric for all considered industries.
- Analyzing the behavior of users by tracking and evaluating **user navigation** through the toolkit is important for all participating industries.
- The **analysis of forms, the misinterpretation of content, conversion funnels and battery issues** seem to be important on consumer devices (m-commerce or software development), but not so much in the industrial context (automation engineering).
- Compared to the other industries, the software development industry has a strong focus on **conversion-funnels, A/B testing and misinterpretations of content**.
- Compared to the other industries, the m-commerce industry has a strong focus on **battery issues**. Taking into account the smaller number of interviews the relative importance of **forms/form handling, external hardware issues** and **environmental influences** also seem high.
- Compared to the other industries, the automation engineering industry has a strong focus on **user groups analysis**, the **position of the device**, the **efficiency of task execution** (speed and dwell time), **deviation analysis** of **user navigation paths**, **haptic controls** and **error rates** (the latter two are not part of the top topics).

The analysis of the **number of companies with requirements for a topic** showed the expected results. The top 5 topics include requirements from all companies. With the decreasing number of requirements the number of companies also goes down. One exception is the topic **heatmaps** which has a low number of unique requirements but incorporates all companies.
The analysis of the number of companies with requirements for a topic showed the expected results. The top 5 topics include requirements from all companies. With the decreasing number of requirements the number of companies also goes down. One exception is the topic heatmaps which has a low number of unique requirements but incorporates all companies.

### Table 2: Top 10 requirements per industry

<table>
<thead>
<tr>
<th>Software Development</th>
<th>M-Commerce</th>
<th>Automation Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Usage Intensity of Functions</td>
<td>Usage Intensity of Functions</td>
<td>Usage Intensity of Functions</td>
</tr>
<tr>
<td>2 Forms/Form Handling</td>
<td>User Navigation - Recording and Presentation</td>
<td>Data Segmentation Dimensions</td>
</tr>
<tr>
<td>3 Interaction Problems Detection</td>
<td>Forms/Form Handling</td>
<td>User Navigation - Tasks Identification</td>
</tr>
<tr>
<td>4 User Navigation - Tasks Identification</td>
<td>Battery Information/Issues</td>
<td>User Navigation - Recording and Presentation</td>
</tr>
<tr>
<td>5 User Navigation - Recording and Presentation</td>
<td>Efficiency</td>
<td>User Groups Analysis</td>
</tr>
<tr>
<td>6 Conversion-Funnel</td>
<td>Interaction Problems Detection</td>
<td>User Navigation - Deviation Analysis</td>
</tr>
<tr>
<td>7 Misinterpretation of Content</td>
<td>Data Segmentation Dimensions</td>
<td>Position of Device</td>
</tr>
<tr>
<td>8 Personal Data</td>
<td>User Navigation - Tasks Identification</td>
<td>Efficiency</td>
</tr>
<tr>
<td>9 Data Segmentation Dimensions</td>
<td>User Navigation - Deviation Analysis</td>
<td>Interaction Problems Detection</td>
</tr>
<tr>
<td>10 A/B-Testing</td>
<td>Identification of Interacting Object</td>
<td>Identification of Interacting Object</td>
</tr>
</tbody>
</table>

Altogether 49 topics were identified and also assigned to one of the question areas of the interview guide (context of use, evaluation methods, metrics and data representation). This classification showed that companies were more interested in the analysis of the context of use and evaluation methods:

- Context of use: 22 topics, 76 requirements
- Evaluation methods: 14 topics, 84 requirements
- Metrics: 8 topics, 37 requirements
- Data representation: 5 topics, 25 requirements

### 3.2 Automation Engineering Prototype

In this section we present a prototype we implemented together with our project partner KEBA AG. There is already work on detecting unexpected browsing behavior in user navigation paths on websites [24] as well as tracking and visualizing navigation paths to identify patterns and provide a user-guided data exploration and clustering [26], but tracking user interactions and analyzing usability is a little investigated field with respect to mobile device usage in industrial environments and the devices used in the automation engineering industry. Furthermore the developed logging framework uses aspect oriented programming which provides the possibility to embed it in application using various programming languages without having to change existing code.

The presented implementation builds on of a previously presented work [10] that provides logging of touch interactions on industrial teach pendants. Teach pendants are handheld terminals that are used for controlling and programming machines and have touch panels for interaction.

Touch panels are used because they allow for a higher flexibility in designing the user interface than physical elements, since UI controls can be adjusted to the current state of a process or context.

#### 3.2.1 Prioritization of Requirements and Features

Due to the diversity of requirements a prioritization was needed. This was done as described in the previous section. The most important requirements are described in Table 3 in further detail.
These top 10 requirements show some dependencies (see Figure 4). Usage intensity and three different topics concerning the user navigation are considered very important issues. The second most requested topic data segmentation dimensions applies to the analysis of recorded data. Therefore data recording has to be implemented before that. Tasks Identification can only be done if data is already recorded. Therefore it is based on a User Navigation Recording and Presentation. The Deviation Analysis is also based on Tasks Identification and consequently also User Navigation Recording and Presentation. The User Groups Analysis is also a kind of analysis of recorded data and thus based on data recording and a kind of data segmentation. Metrics like Usage Intensity of Functions and Efficiency can also only be calculated based on recorded data, but their implementation is not that complex or challenging.

The following sections discuss the implementation of these requirements with a project partner of the automation engineering industry. Consequently we first describe how the logging of user interactions and navigation paths is implemented and present afterwards how navigation paths, usage intensity and efficiency are visualized.

3.2.2 Logging Implementation

The prototype is implemented on a KeTop teach pendant, which is used to operate a KeMotion robot controller (see Figure 5). The KeMotion robot controller is a combination of a classic PLC (Programmable Logic Controller) and a fully-fledged robot controller. The KeTop teach pendant displays the interface and is used to operate the system by creating, executing and modifying sequential robot programs, confirming error messages or moving the robot manually. Since this is the device where all interactions are made on, the logging takes place on it.

Table 3: Top automation engineering requirements topics

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  Usage Intensity of Functions</td>
<td>Requirements cover how often certain UI controls, screens, software functions etc. were used.</td>
</tr>
<tr>
<td>2  Data Segmentation Dimensions</td>
<td>Requirements cover which characteristics are provided in the data analyzing tool.</td>
</tr>
<tr>
<td>3  User Navigation - Tasks Identification</td>
<td>Requirements cover the possibility to automatically identify which tasks users wanted to execute.</td>
</tr>
<tr>
<td>4  User Navigation - Recording and Presentation</td>
<td>Requirements cover which data should be tracked and which visualizations should be provided.</td>
</tr>
<tr>
<td>5  User Groups Analysis</td>
<td>Requirements cover user characteristics for segmentation.</td>
</tr>
<tr>
<td>6  User Navigation - Deviation Analysis</td>
<td>Requirements cover the possibilities to identify how differently users execute tasks.</td>
</tr>
<tr>
<td>7  Position of Device</td>
<td>Requirements cover where the device is located and how the device is held in relation to the user.</td>
</tr>
<tr>
<td>8  Efficiency</td>
<td>Requirements cover a number of metrics concerning speed, dwell time etc.</td>
</tr>
<tr>
<td>9  Interaction Problems Detection</td>
<td>Requirements cover the identification of problems due to sizes of elements, gestures, scrolling etc.</td>
</tr>
<tr>
<td>10 Identification of Interacting Object</td>
<td>Requirements cover the detection the interacting item (pen, finger, glove etc.)</td>
</tr>
</tbody>
</table>

Figure 4: Dependencies between requirements and features

Taking into account the prioritization (Figure 3), these dependencies (Figure 4), and their complexity, the following features were identified as most important:

- logging the interactions of users
- analyzing the usage intensity of functions by aggregating the interaction data and visualizing this data
- logging user navigation sequences and providing visualizations of navigation sequences
- analyzing the efficiency of users

Since logging forms the foundation of all other features, Table 4 lists the top requirements of prioritized topics of the automation engineering industry.

Table 4: Requirements of top topics

<table>
<thead>
<tr>
<th>User Navigation – Recording and Presentation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 A user navigation path should track and present the ways users navigate through the application.</td>
<td></td>
</tr>
<tr>
<td>2 User navigation paths should be visualized graphically.</td>
<td></td>
</tr>
<tr>
<td>3 Every screen and the actions on screens should be traceable.</td>
<td></td>
</tr>
<tr>
<td>4 The relative distribution of actions and paths should be visible.</td>
<td></td>
</tr>
<tr>
<td>5 The order of actions on screens should be traceable</td>
<td></td>
</tr>
<tr>
<td>6 For every screen a heatmap should show the interaction density.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Usage Intensity of Functions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 The number of interactions per screen should be visible.</td>
<td></td>
</tr>
<tr>
<td>2 The number of function invocations should be visible.</td>
<td></td>
</tr>
<tr>
<td>3 The number of interactions on hardware/software controls should be visible.</td>
<td></td>
</tr>
<tr>
<td>4 The usage frequency of controls should be visible.</td>
<td></td>
</tr>
<tr>
<td>5 The number of function invocations should be visible separately for every screen of the app.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 The speed of executing tasks should be measured.</td>
<td></td>
</tr>
<tr>
<td>2 The dwell time on screens should be measured.</td>
<td></td>
</tr>
<tr>
<td>3 The time between interactions on a screen should be measured.</td>
<td></td>
</tr>
<tr>
<td>4 The speed should be measured by the time spent between interactions (independently from software or hardware controls).</td>
<td></td>
</tr>
</tbody>
</table>
Taking into account these considerations the implementation of the selected requirements had to extend the framework with the following services:

- A service to record touch interactions (MouseEventService)
- A service to record the user navigation within the TeachView application (LifecycleService)
- A service to take screenshots for visualizations (ScreenshotService)

These services are part of the InteractionLoggerLib and receive the corresponding events by registering as listeners at the LoggingKernel. In the following subsections we will describe these services in further detail.

### 3.2.2.1 Mouse Event Service

Since the device does not support multi-touch the Mouse Event Service only records touch clicks. The included coordinates are used for mapping them on the corresponding screen to identify touch points. The implementation uses a queue of click events that is serialized and written to the file in certain time intervals to minimize I/O operations and to reduce the computational burden.

To get the X/Y-coordinates of touch points events had to be intercepted at the lowest possible level. Therefore, a listener that gets notified of all AWT events was used. This is necessary, because the TeachView application might run as Java process on a desktop PC (“emulating mode”) and in this case the application window will most likely not be in the top-left corner of the screen. Consequently, touch points must be calculated relative to the application frame and its position on the screen.

Additionally the X/Y-coordinates, every touch click is also associated with the underlying UI control (e.g. button). Since controls in the TeachView application do not provide unique IDs, a custom ID mechanism, using AOP based on load-time weaving and inter-type declarations (altering classes at runtime), was used. Currently none of the visualizations (see 3.2.3) uses this data, but we plan to implement a Hotspot Analysis visualization [1] that aggregates clicks for the underlying UI elements and therefore shows click intensity based on controls and not X/Y-coordinates.

### 3.2.2.2 Screenshot Service

The screenshot service makes images of the current screen content by writing raw image data to a file.

Since multiple touch events can occur on a certain screen without having effects on its content, it is not necessary to take a screenshot for every interaction event. Consequently screenshots are only taken when the screen content changes. This is not just the case when a completely new view is opened. Also smaller changes on a certain view (e.g. selection of a combo box entry), popups etc. are considered important changes. Therefore, upon receiving corresponding events the screenshot service creates a screenshot of the current view and saves it as BMP file to the internal memory using the ID of the view in the TeachView application as unique identifier.

To generate a heatmap for a certain view, the click events have to be associated with the saved screenshot. Therefore the service also listens to click events, and assigns them to the last captured screen in an event log. The event log is a json-file whose lines...
represent touch clicks that are associated with the screenshot and contain further information about the absolute and relative position to the application frame and its position on the screen.

3.2.2.3 Lifecycle Service
The lifecycle service is responsible for tracking the displayed views. Typically multiple views are needed to teach a robot a certain behavior. But since multitasking is not supported at a certain time, only one view can be visible. Hence the navigation path can be reproduced for a single user. By using a timestamp, the time spent on each view can be calculated as well.

The service is registered at the kernel as listener for events that occur when a view is opened (MaskOpenEvent) or closed (MaskCloseEvent).

The number of transitions between views is also aggregated to provide a quantitative analysis. This is done by saving all realized transitions to a list (edge list) and counting their occurrence (edge visits).

3.2.3 Visualizations
Besides the logging of user interactions and navigation paths, the visualization of this data was the second topic we focused in the development of the prototype. For this visualization a data viewer application was implemented using Swing. To illustrate usage intensity, heatmaps were used. To show the user navigation paths directed graphs, timeline views and Sankey diagrams were used.

3.2.3.1 Heatmap
Based on the logging of the screenshot service, each touch click is associated with a certain screen. Therefore all touch clicks from a single user on a single screen can be aggregated in heatmaps. This aggregation leads to three possible data representations:

- A heatmap of all clicks of one user visiting a screen one time
- A heatmap of all clicks of one user visiting a screen multiple times
- A heatmap of all users visiting a screen multiple times

Heatmaps aggregate clicks and visualize intensity. Intensity is illustrated by different colors from blue to red (red symbolizes high intensity) (see Figure 7). Furthermore, the underlying single events can be seen in a context menu.

Some issues may emerge when aggregating the data, because of dynamic content on these views. As seen in line 2 of Figure 7 elements can vary. Consequently the aggregation shows a blending of these elements (in this case text).

3.2.3.2 Timeline View
In the timeline visualization (see Figure 8) touch clicks are displayed as red crosshairs on the screenshot. Indexes beside the crosshairs denote their numerical order of appearance. Thus the sequence of clicks and their position on the screen are visible.

![Timeline visualization](image)

Figure 8: Timeline visualization

The lower section of the visualization features a time bar spanning the entire dwelling time from the first to the last click on the view. It visualizes the click density of interactions over time, visualizing efficiency metrics speed, dwell time on screen and time between interactions.

3.2.3.3 Directed Graph
Navigation paths are illustrated using directed graphs. They show which views a user has visited in a certain session. Each view is represented by a node in the graph. Transitions of one view to another are illustrated using a connection between the nodes. The arrows on this connecting lines show the direction of the transition. Furthermore the quantity of transitions is indicated by the thickness of the line (see Figure 9).

![Directed graph](image)

Figure 9: Directed graph

Furthermore, detailed information about transitions and views is visible in context menus of connecting lines and nodes. The context menus of lines show the names of the connected nodes and the numbers of transitions. Right-clicking on a node will
display further information about the number of screenshots that were made of the view, their file names and a list of touch clicks that occurred on that view. However, the graph does not provide information about the chronological order of navigation steps.

3.2.3.4 Sankey Diagram

The information in a Sankey Diagram is similar to the directed graph, but shows the amount of transitions in a more visual way. Each view is represented by a vertical bar. Transitions leading to a view are visualized as inbound streams, the transitions leading away from a view are visualized as outbound streams. The quantity of transitions is proportional to the width of the arrows (see Figure 10).

![Sankey Diagram](image)

**Figure 10: Sankey diagram**

From a technical point of view the graph is generated differently than the other graphs. It is rendered on an HTML page using JavaScript, which is then rendered within a JavaFX WebView that is displayed on the Swing UI (see Figure 11).

![Technical implementation of Sankey Diagram](image)

**Figure 11: Technical implementation of Sankey Diagram**

The JavaScript module accepts a comma-separated list of nodes as parameter and generates the Sankey diagram from it. Since that list needs to be generated dynamically, it is executed from the JavaFX-WebEngine class as soon as the page loading is completed.

4. DISCUSSION AND FUTURE WORK

In this paper we introduced our methodology to conduct qualitative requirements analyses, showed the results of a cross industry requirements analysis including the fields m-commerce, software development and automation engineering and presented our approach to prioritize and implement the requirements of the automation engineering industry.

The study showed that the requirements of the analyzed industries and their prioritization are quite different. This is especially true for the automation engineering industry, since the industries software development and m-commerce focus on consumer devices (smartphones, tablets) whilst the automation industry uses different types of mobile devices for very specific tasks.

The analysis itself has clear limitations as there was only one interview taken with the m-commerce sector, compared to three interviews with both the automation industry and the software development branch. This leaded to the fact that more requirements were gathered from these sectors with more interviews taking place. The same applies to the length of the interviews. Nevertheless, the method used in gathering the requirements from a cross industry perspective helped us to identify common and reusable features for implementation. As usual in agile software development it is necessary to involve users from the beginning and implement their requirements. The interviewees, as well as further users will be asked to comment on the features of the prototypes and provide feedback.

Based on the requirements analysis we implemented a prototype for the automation engineering industry. The system is based on the hardware of our project partner KEBA AG that uses KeTop teach pendants to operate KeMotion robot controllers. We prioritized the requirements based on dependencies of the features and started by tracking the usage intensity of functions, recording and presenting user navigation paths and laying out efficiency metrics. Using aspect-oriented programming, we recorded the necessary interaction data (touch interactions and navigation data) without having to change existing code. The tracked data is aggregated based on the views of the application and visualized using heatmaps, a timeline view, a directed graph and a Sankey Diagram.

In future work, we want to provide a toolkit that is platform and device independent and provides immediate usability analyses. Therefore we already implemented the analysis methods as web based services. Hence we keep the code to capture events on the devices small and generic and send interaction data immediately to the server with no need to export them. Subsequent analyses are processed on a server. This allows us to rapidly adapt the tracking code to new platforms while providing the same analysis tools through a web interface.

Furthermore we consider recording interaction data and providing visualizations as a valuable first step, but in future work we want to extend the toolkit by implementing the missing requirements and providing automated unsupervised usability checks. These checks should analyze the tracked interaction and context data and identify potential usability problems in an autonomous way.

5. ACKNOWLEDGEMENTS

The research presented is conducted within the Austrian project “AUToMAte – Automated Usability Testing of Mobile Applications” funded by the Austrian Research Promotion Agency (FFG) under contract number 839094.

6. REFERENCES


