Usability Mining

Automated Analysis of Information System Usability Based on Process Mining

Sharam Dadashnia, Constantin Houy, and Peter Loos

Abstract The usability of information systems (IS) is a key characteristic in the context of software selection and IS design. IS are supposed to support various functions in business organizations in an effective and efficient way, to be *easy to use and easy to learn*, and to produce *satisfactory outcomes* for users. Research has sought to engineer IS based on automated usability checks, but while the concept of process mining offers considerable potential in this context, so far only little research has been done on the potential of process mining approaches for automating analyses of IS usability. We describe the journeys through and the results of several design research projects that have investigated the potential of combining process mining approaches and usability engineering (i.e., *usability mining*). The design artifacts presented in this study elucidate the potential of usability mining in the context of usability studies, focusing on mobile policing applications developed and used in several projects in Germany. We present a dedicated reference framework for the design of usability mining solutions and a software implementation that we use to illustrate the artifacts' applications in the mobile policing scenario. We present the results of several design projects in which we gathered experience concerning usability mining and its application in real-world scenarios. While the development of a usability mining solution can be managed according to certain design recommendations, data preparation and data cleansing present particular challenges in usability mining endeavors.

S. Dadashnia · C. Houy · P. Loos (✉)
Institute for Information Systems (IWi), German Research Center for Artificial Intelligence (DFKI), Saarland University, Saarbrücken, Germany
e-mail: peter.loos@iwi.dfki.de

S. Dadashnia
e-mail: sharam.dadashnia@iwi.dfki.de

C. Houy
e-mail: constantin.houy@iwi.dfki.de

J. vom Brocke et al. (eds.), *Design Science Research. Cases*, Progress in IS,
https://doi.org/10.1007/978-3-030-46781-4_7

1 Introduction

Information systems (IS) are supposed to support many functions in business organizations effectively and efficiently, so they must be *easy to use*, *easy to learn*, and lead to a *satisfactory outcome* for users. In addition to these central aspects of *usability*, according to the common definition of *usability* provided by the International Organization for Standardization (ISO), other important usability characteristics include the *efficiency* of use ("quick task performance") and its *memorability* ("quick re-establishment of proficiency after a certain period of not using a product") (Nielsen, 1993). Usability is a key consideration in the selection of business application software and IS design (Thaler, 2014).

Against this background, the field of usability engineering has gained significance in the IS discipline, as it provides results that support the design and development of highly usable IS (Adams, 2015). Considerable research has gone into the usability of IS in general and in such environments as IS management (Batra & Srinivasan, 1992), visual programming (Green & Petre, 1996), websites and web-based business IS (Geng &Tian, 2015; Harms & Schweibenz, 2000a, b), and applications' user interfaces (Hilbert & Redmiles, 2000; Ivory & Hearst, 2001). In addition, several attempts have been made to develop automated approaches to usability analysis (Montero, González, & Lozano, 2005; Schuller, Althoff, McGlaun, Lang, & Rigoll, 2002), especially approaches that are based on established usability metrics (Hornbaeck, 2006; Seffah, Donyaee, Kline, & Padda, 2006), to support the design of IS. However, only a little research has been done on the potential of using process mining approaches to analyze the usability of IS with a dedicated reference to the business processes that an information system or application is supposed to support (Thaler, Maurer, De Angelis, Fettke, & Loos, 2015).

Process mining is a sub-area of data mining and a sub-area of business process management (BPM). The basis for process mining is log data that is produced by business applications or business IS, such as enterprise resource planning (ERP) systems or workflow management systems (WfMS), especially in the form of *event logs* that document the occurrence of particular events in a business process. Information can be extracted from this log data that can support the identification and description of business processes that have actually been executed (van der Aalst & Weijters, 2004). Process mining can serve several purposes, but three process mining approaches are distinguished with regard to their objectives: *discovery*, *conformance checking*, and *enhancement* (van der Aalst, 2012a, b):

Discovery refers to the procedure through which a process model is derived from an event log. Process discovery, which is used frequently, is an effective procedure with which companies can identify and document their actual business processes and working procedures.

Conformance checking supports the comparison of an existing process model (a to-be process model) to a model that is based on event logs (an as-is process model) or to the log data itself. Thus, conformance checking enables the identification of deviations between defined and actually executed business processes.

Enhancement supports the improvement of existing process models and process definitions by using new findings from the analysis of the actually executed processes that is documented in the event logs.

The authors took part in several research endeavors that investigated the potential of combining process mining approaches and usability engineering, focusing on a strong relationship between the use of application software or an information system and the underlying business process that is supported. The authors have also developed several artifacts and software components that demonstrate the potential of combining these approaches in several design science research (DSR) projects. Process mining can automate usability studies and usability engineering, which has been treated in scientific contributions under the umbrella term *usability mining* (Thaler, 2014; Thaler et al., 2015). Even real-time usability improvement is possible based on log data, which are detailed recordings of actual user behavior in a software application (Dadashnia, Niesen, Hake, Fettke, & Mehdiyev, 2016b). The artifacts presented in this study demonstrate the potential of usability mining in the context of mobile applications that support the German police in the acquisition of accident data and data concerning criminal complaints on the street.

The remainder of this study proceeds as follows: The next section describes the context of the design project, which brings together the worlds of process mining and usability engineering. Then we provide an overview of the research journey we undertook to develop and elaborate our idea of using process mining in usability studies. Several development phases or maturity levels of the resulting artifacts are described as we moved step-by-step toward a more automated analysis procedure while using process mining in our usability studies. We present the dedicated reference framework and software prototype that we developed to support automated process mining-based usability analysis in IS based on event logs. The next section presents the resulting artifact, its application, and its evaluation in a real-world context, as well as some considerations concerning the progression toward a dedicated design theory. Finally, we discuss the key lessons learned in this design journey.

2 The Context

This study explains and demonstrates the potential of process mining approaches in the context of usability engineering. Usability mining offers considerable potential for automated analyses of the usability of business IS. The design journey described in the following is related to several projects in the context of analyzing the use and usability of mobile applications that support the German police in the acquisition of accident data and data concerning criminal complaints. This topic is typically addressed in the literature using the term *mobile (digital) policing*.

The term *mobile policing* describes the use of mobile application systems, such as specialized software applications on mobile devices like smartphones or tablets, to support policework-related processes on the street or in the field with the goal of better information management independent of stationary information and communication systems, access networks, or specific locations (Houy, Gutermuth, Dadashnia, & Loos, 2019). Application software on mobile devices can support both the activities of the police in the field and their follow-up activities in the office, thus facilitating integrated information management and reducing the number of interfaces. Using a paper notebook and a pen to document relevant information related to a traffic accident is still common practice for police throughout the world. Then, after returning to the office, police officers have to enter the content of handwritten notes into the information system, which is not efficient and can result in faulty entries. Using mobile applications in an integrated information infrastructure can help them avoid errors and make the whole process more efficient. These positive effects are more likely if the underlying mobile applications have good usability.

Against this background, the following design journey describes several design phases and project iterations in the context of two usability engineering projects in which the authors participated:

(a) a *proof-of-concept* project that investigated the potential of mobile application software in the context of accident-data acquisition in Germany, using only a few mobile devices in a well-defined small application scenario, and
(b) a *pilot* project that investigated the economic aspects of using mobile application software in the context of accident-data acquisition and data concerning criminal complaints in Germany, using a larger number of mobile devices in a well-defined but broader application scenario.

The basis for the development of design ideas in these endeavors was the usability mining lifecycle proposed in Thaler (2014), which comprises six phases:

(1) *User monitoring*: In this phase the user behavior and interaction with an information system is monitored and documented by means of system log files.
(2) *Trace clustering*: In this phase, the log data is clustered according to criteria that are relevant to the following analyses (e.g., clustering the data concerning specific user groups).
(3) *Usage model derivation*: Based on the clustered log data, a usage model of the information system is automatically developed by means of process discovery approaches, resulting in an as-is process model of the information system usage. The log data allows the metrics that support the following analysis of the usage model to be computed.
(4) *Usage model analysis*: The usage model can be analyzed considering various potential metrics, including *model metrics* (e.g., concerning model size, model complexity, sequentiality), *process metrics* like execution time and error rate, and common *usability metrics* like irrelevant actions, undo actions, and use of the software's help function.
(5) *Recommendation derivation*: In this phase, the results of the analysis are interpreted to develop concrete design recommendations that will improve the usability of the system based on the users' needs.
(6) *Implementation of improvements*: Finally, the design recommendations derived from the analysis are implemented in the software system.

Although we developed our artifacts for usability mining in the context of police-work processes supported by mobile devices, the developed reference framework and the software prototype can be used in many other contexts because the basis for the resulting usability analyses are log files produced by application software, viz., business IS. Hence, the results of our design endeavor could easily be transferred to other application scenarios and application contexts.

3 The Journey

Like almost every design science endeavor, our design journey did not follow a linear process. Our understanding of the particular problems in the context of the *proof-of-concept* and the *pilot* project in the field of mobile policing, as well as possible solutions, improved throughout several design iterations. In the following, we describe our journey toward our prototype for automated usability mining in the case of analyzing mobile application software for the acquisition of accident data and data concerning criminal complaints used by the police in Germany.

3.1 Preliminary Studies

Several preliminary studies of usability mining were conducted at the authors' institution in various application contexts, most of which were relevant to business organizations (Thaler, 2014; Thaler et al., 2015; Dadashnia et al., 2016a, b; 2017). However, in the design science endeavor described here, the particular problems and possible solutions were heavily influenced by the specific conditions of the police context.

As mentioned, it is still common practice for police to use notebook and pen to document information related to a traffic accident and then to enter the content of the handwritten notes manually into the information system in the office. Therefore, at the beginning of the proof-of-concept project, we conducted interviews with police officers to model an "ideal" or at least a commonly accepted structure of this data-acquisition process on the street. We also observed the process of "manual" data acquisition in order to document several cases of actual data-acquisition processes, as well as the execution time of the process instances. Documenting the common process structure and some real-world process instances supported the design and customization of the data acquisition form provided by the mobile policing app on the mobile devices.

To support usability mining concerning the actual use of the mobile policing application, three design iterations were executed.

Table 1 Extract of an exemplary usage event log

Activity	Person	Time stamp
Start app usage	Officer 3	00:00:00
Create new accident case	Officer 3	00:00:20
Document time of accident	Officer 3	00:00:34
Document category of accident	Officer 3	00:00:48
…	…	…

3.2 Lap 1—Initial Usability Mining Solution

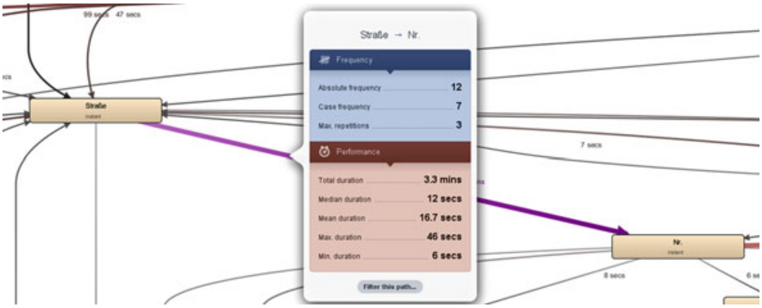
Understanding the Problem: When the proof-of-concept project began, the mobile policing app was not configured to produce event logs, as this feature could not be provided in the test setting. To be able to use process mining approaches in our usability mining solution, we had to capture event logs in another way. To avoid disturbing police officers in their work, the research team could not be present in most data-acquisition cases to observe and document the usage, so we had to find a workaround.

Understanding the Solution: In configuring our usability mining solution, we had to deal with the missing event log, so we used an additional screen-capturing software on the mobile devices and asked all proof-of-concept participants to record their interactions with the mobile policing app. Then we manually transcribed the resulting videos, thereby manually producing event logs that could be clustered and used for the derivation and analysis of the usage model by means of process mining techniques. Table 1 is an extract of an exemplary event log.

It was also possible to extract more coarse-grained usage data from the central SharePoint server, which received data from all mobile devices that participated in the mobile policing infrastructure. This usage data contained information about the use of various sub-areas of the app (form pages for persons involved (e.g., witnesses) or the cause of an accident) and could also support usability mining activities. Although no exact click-stream information could be obtained this way, this workaround helped us to gather further information, especially in cases that were not recorded by means of the screen-capturing software. This information served as an input for usability mining in the next phase.

The usage model partly shown in Fig. 1 was created using the filtering mechanisms of the process mining tool *Disco,*[1] which we used to analyze the usage and system interaction data. Based on this usage model, we could, for example, provide a recommendation concerning the identification of the nearest house number ("Nr.") on the street ("Straße") in which an accident had happened. Identifying a house number may quite time consuming if it is not directly recognizable, such as in the area of a large intersection or in streets with commercial buildings. While this problem has nothing to do with the application design but is due to the underlying technical process, such

[1] https://fluxicon.com/disco/.



Straße → Nr.

Frequency

Absolute frequency	12
Case frequency	7
Max. repetitions	3

Performance

Total duration	3.3 mins
Median duration	12 secs
Mean duration	16.7 secs
Max. duration	46 secs
Min. duration	6 secs

Filter this path...

Fig. 1 Exemplary part of the usage model using the process mining tool Disco

cases often occur in real-life policing processes and are well suited to improvements in the work process and the supporting IT infrastructure. GPS-supported localization services were recommended based in this usage model to automate this step in the data acquisition.

Thus, the usability mining solution used in the proof-of-concept project consisted of a data-acquisition environment using a screen-capturing software on a mobile device and an existing process mining tool to enable usability mining. Figure 2 presents an overview of the usability mining solution that resulted from the first design iteration. An improvement on the approach that was realized in the following pilot project is demonstrated in the next section.

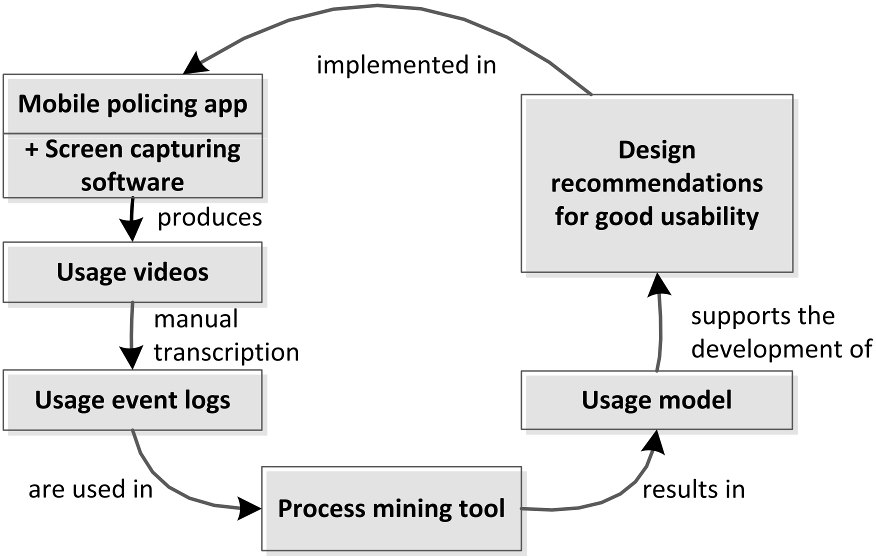


Fig. 2 Usability mining solution resulting from the first design iteration

3.3 Lap 2—Automation of Capturing the Usage Event Log

The second design iteration took place during the pilot project, which followed the proof-of-concept project. The pilot project used a larger number of mobile devices to investigate the economic aspects of using the mobile application software in the context of accident data acquisition and data concerning criminal complaints in Germany.

Understanding the Problem: The usability mining solution that was developed in the first design iteration relied on manual transcription procedures and would have caused too much effort in a scenario with a large number of mobile devices (about 100) over a period of several months. Hence, further automation steps concerning capturing the usage event log in the user-monitoring phase were needed.

Understanding the Solution: At the beginning of the pilot project, we developed a concept and a software implementation of a usage-logging script that allows usage event logs to be captured automatically in the mobile policing app. The software vendor then integrated this logging script into the mobile policing app used in the pilot project. Thus, the user monitoring phase could be automated, and the usability mining procedure was much more efficient and could also deal with the higher amount of data acquired during the pilot project. We used two tools to analyze the usage event logs: Disco for process mining and Microsoft Power BI for further data analysis procedures that were useful in the usability mining context. Figure 3 illustrates the usability mining solution that resulted from the second design iteration.

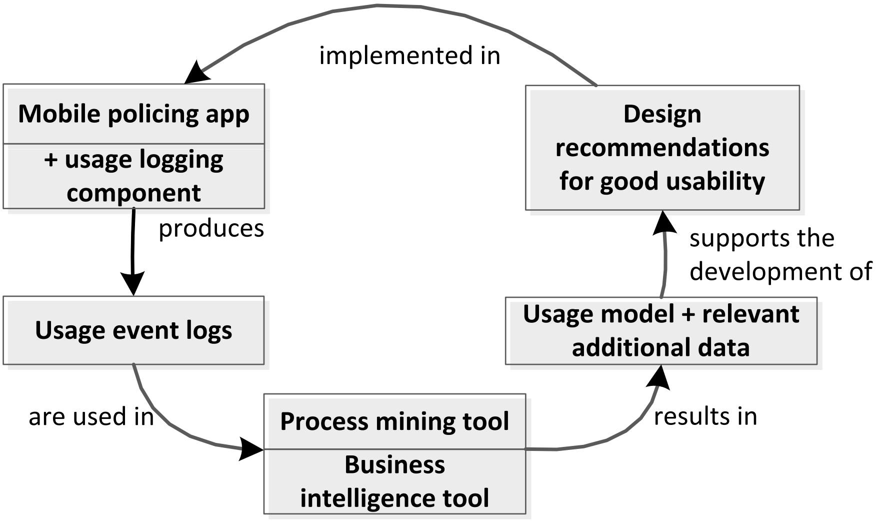

implemented in
Mobile policing app
+ usage logging component
produces
Usage event logs
are used in
Process mining tool
Business intelligence tool
results in
Usage model + relevant additional data
supports the development of
Design recommendations for good usability

Fig. 3 Usability mining solution resulting from the second design iteration

The next section demonstrates an improvement of our second approach that was also realized in the pilot project.

3.4 Lap 3—Automatic Calculation of Usability Metrics

Understanding the Problem: While the usability mining solution developed in the second design iteration provided the usage models and helpful data analyses like common process key performance indicators (KPI), we still lacked precise information related to peculiarities of the system usage that could be measured with metrics commonly used in the field of usability engineering. Usability metrics and related information generated based on process mining methods can be helpful in detecting usability problems concerning IS quickly.

Understanding the Solution: To improve the functionality of our solution for analyses, we developed a concept and a software implementation for the automated calculation of usability metrics from the automatically captured usage event logs. This development step resulted in a concrete usability mining solution and a more general reference framework for automated usability mining, which is presented in the following section in more detail. Figure 4 illustrates the usability mining solution that resulted from the third design iteration.

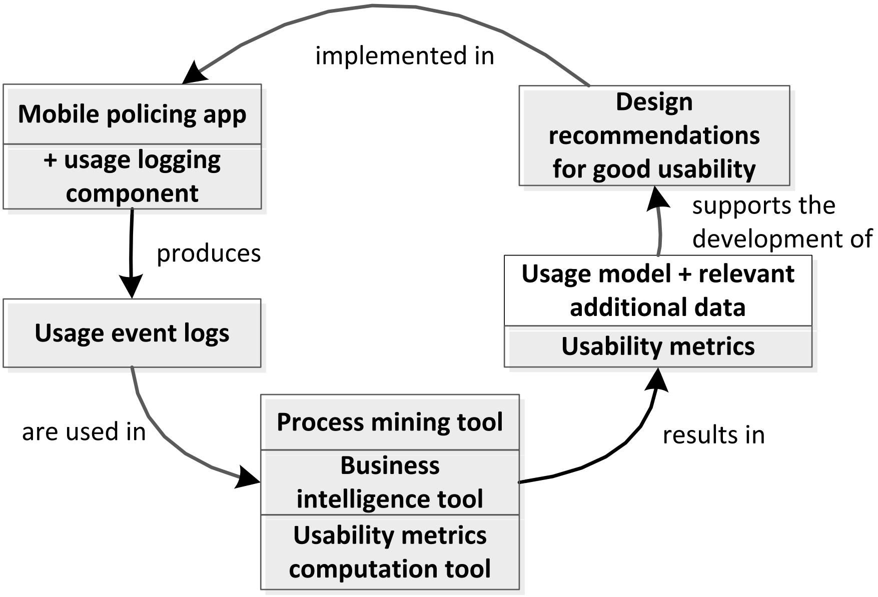

implemented in
Mobile policing app
+ usage logging component
Design recommendations for good usability
supports the development of
produces
Usage model + relevant additional data
Usability metrics
Usage event logs
Process mining tool
Business intelligence tool
Usability metrics computation tool
are used in
results in

Fig. 4 Usability mining solution that resulted from the third design iteration

4 The Results

4.1 Presentation of Artifact(s)

4.1.1 Reference Framework for Automated Usability Mining

This section introduces a new reference framework for automated usability mining. In contrast to the lifecycle concept presented in Thaler (2014), the automation of all possible steps of the usability mining process is a central aspect of our reference framework.

We focus on a detailed presentation of the automated calculation of usability metrics and the related components in the reference framework. The reference framework can serve as a design recommendation for individual automated usability mining solutions. An instantiation of this reference framework was used in the pilot project. The term *artifact* in the following discussion refers to either the reference framework or its instantiation in the context of the pilot project. The major purpose of the artifact is to support both software developers and usability experts with its usability engineering knowledge. The reference framework consists of three major components, visualized in the Fig. 5 and explained in more detail in the following.

The advantage of the framework is its focus on business IS and the underlying business processes, which allows the conformance of the actual user behavior with the defined business processes to be measured.

1. **Data collection**: The first component offers experts and system developers the possibility to generate usage event logs while using business IS, to store them appropriately, and to ensure they are in a suitable form for later analysis. This component should offer the use of multiple data sources and should also be easily extendable.
2. **Automated metric calculation**: The second component supports the automated calculation of usability metrics based on the data collected by the first component. Usability metrics and methods from the field of process mining are used to exploit the potential for automation. The artifact provides calculation rules for the metrics in the form of, for example, pseudocode and the use of process mining algorithms

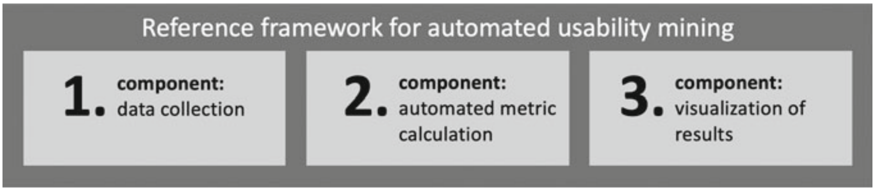

Reference framework for automated usability mining
1. component: data collection
2. component: automated metric calculation
3. component: visualization of results

Fig. 5 Three components of the reference framework for automated usability mining

or their extensions or a combination of approaches. Conceptual interfaces for extending metrics by adding additional data sources are also part of the artifact.

3. **Visualization of results**: The third component of the artifact manages the visualization of the generated results in a process-aware way. Here, the calculated usability metrics are added in order of the process based on the usage model created by means of process discovery approaches. Hence, the usage model is a fine-grained process model in which individual click activities can be assigned to a function in the business process model, ensuring that technically incorrect sequences or paths in the system can be detected visually. The goal is to highlight individual click activities in the same color if they belong to the same business process function, which is essential in applications that depict large business processes. (For example, efficiency improvements can easily be visualized through color gradients.) Here, exploration in the usage model can directly support the detection of rebounds or poorly arranged functionalities, as such patterns that occur frequently can indicate inefficiently arranged elements of the user interface. Frequent rebounds can also indicate an outdated process. The color highlighting, in combination with calculated and annotated metrics, is an innovation in the context of existing applications for analyzing application systems' operational usability.

4.1.2 Instantiation of the Reference Framework for Automated Usability Mining in the Context of the Pilot Project

Here we provide a detailed review of the artifact using a running example of an automated calculation of usability metrics. We focus on the second component, *automated metric calculation.* While we explain the basic concepts for all components, we also describe the technical concept and the implementation of the second component and illustrate the component's application by means of an example. Figure 6 illustrates the focus in the following explanations of our instantiation of the reference framework.

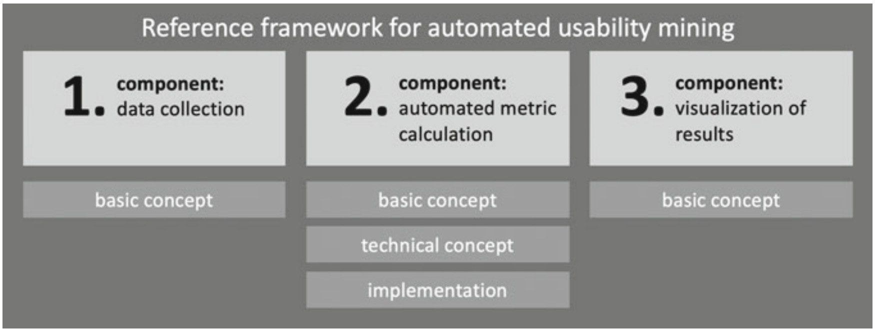

Reference framework for automated usability mining
1. component: data collection
2. component: automated metric calculation
3. component: visualization of results
basic concept
basic concept
basic concept
technical concept
implementation

Fig. 6 Focus of the explanations of the reference framework

Literature has provided about fifty usability metrics with automation potential, which we classified in terms of their automation potential in literature review (the publication of the literatur review including with the respective classification is forthcoming). Certain metrics were already automatable, but some metrics can be raised to a new level of granularity in the information provided using process mining methods. In the following, we present the automated calculation of one metric, *usage effectiveness*, to demonstrate the artifact's development process. The automated calculation of this metric is based on the approach presented in Saleh, Ismail, and Fabil (2017), which supports analyses of software systems' effectiveness. To provide meaningful results, certain manual steps, such as task definition and measurement of duration times, must be executed at the beginning. The ISO definition of usage effectiveness is important for the software systems that support a firm's essential core business processes if the processes are to be effective and have low duration times (ISO:9241, 1998). Effectiveness refers to how well the system supports the user in achieving high-quality results. Saleh et al. (2017) refer to the *number of touches* with the software as indicating effectiveness by showing how many interactions are required to achieve a goal. The metric provides insights into *successfully executed tasks*, so it indicates the software's effectiveness in a usability-aware way. For example, in the context of our pilot project, the *usage effectiveness* metric indicates how many interactions are necessary for a police officer to acquire all of the accident data when using the mobile policing application.

Next, we present the basic concepts concerning the artifact in relation to our running example.

1. **Data collection**
   Usage event logs play an important role as an input variable to the metric calculation. In the context of our pilot project, the usage event log records the actual use of the accident-data acquisition forms and the police officers' interactions with the mobile device. Here we provide requirements specifications to ensure that all necessary data is collected by a corresponding software component. We describe the necessary data attributes for log entries to calculate the *usage effectiveness* metric in the context of the pilot project:

   1. **caseID**: The individual caseID is automatically generated by the system when an officer records an incident. In the pilot project scenario, the caseID is generated by a control system from a previous process step. The caseID remains unchanged during the entire recording and the subsequent post-processing.
   2. **timeStamp**: The time stamp, which saves the exact time of every interaction with the system, consists of a customer-specific time specification for when the action is executed so the sequence of activities and the time between them are recorded.
   3. **divElementID**: The div-element ID is a unique ID for each separate part of the form document (div-element) used during the action, such as the Textbox, the Dropdown Menu, or the Checkbox. This ID enables a clear mapping

of a log entry and a corresponding screen element. For two identical div-elements on different views, different labels are used, so the div-elements can be distinguished in later analyses.

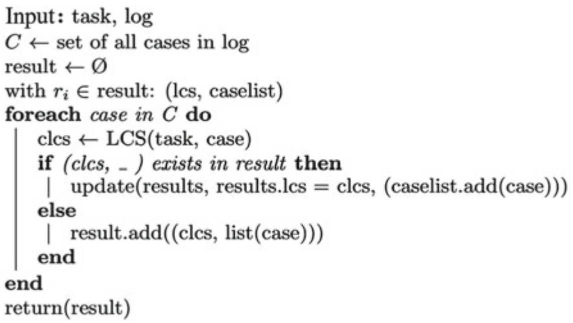
4. **versionNumber**: The version number, which refers to the application's version, highlights the differences in the versions and documents the process of the application development.

Besides the user's interaction data, a process model designed for the common workflow must be used in the analysis. The common workflow can be imported to the system via a sequence of click events, which must be enriched with the corresponding activity from a business process perspective to ensure conformance with the process. In our pilot project, we defined a best-practice process model for the accident-data acquisition in mobile policing, which was the basis for the conformance checks in the project.

2. **Automated metric calculation**
   The described component measures how many of the activities specified in a task are actually executed by the user. For this purpose, the *longest common subsequence* (LCS) of process steps executed and corresponding activities in the model is gauged. The length of the LCS is compared to the number of required activities, relaying the cases in which only certain activities were completed and which activities are the most frequent. Hence, weak points can be uncovered and badly functioning task sections can be improved. The metric used is based on Tullis and Albert (2008), who introduced the binary or ordinal evaluation of tasks under the term *task success*. A clear start and end state must be defined at the beginning of any study, and success must be defined. In our case, we already knew the start and end events as well as the process's goal (e.g., successfully saved accident data). Saleh et al. (2017) propose an automatic measurement of the number of successfully completed tasks in relation to the number of tasks begun, but the extant research describes no evaluation algorithm. Therefore, we describe the technical concept considering the available process knowledge.

Technical Concept: The usage behavior we consider here normally deals with the completion of a task like acquisition of accident data. Otherwise, the log must be examined for the task's activities and its start and end states, and relevant data must be determined. Therefore, the LCS is used, and the information that can be deduced from the defined process (in our case, the best-practice process for acquisition of accident data). We use the task to be analyzed and the given log as input variables. In this context, we describe the task as a sequence of single activities. For each case, we calculate the LCS to get an overview of the executed process instances' conformance. These subsequences are stored in a result set, which is the input parameter for the metric calculated later regarding the effectiveness ("correctly executed instances of a corresponding task") (Fig. 7).

The score is calculated as follows: The sum of the frequency of the single LCS and the length of the single sequences is divided by the related cases and the length of the given task. (We ensure that we calculate with only the subsequences of a given



```
Input: task, log
$C \leftarrow$ set of all cases in log
result $\leftarrow \emptyset$
with $r_i \in$ result: (lcs, caselist)
foreach case in C do
    clcs $\leftarrow$ LCS(task, case)
    if (clcs, _ ) exists in result then
        | update(results, results.lcs = clcs, (caselist.add(case)))
    else
        | result.add((clcs, list(case)))
    end
end
return(result)
```

Fig. 7 Pseudocode for the calculation of the LCS result set

task; otherwise, the result set would be empty.) The result is a metric that gives the relative frequency of the task to be executed and the actual executed sequences. If the result is 1, the software system and the corresponding business process (e.g. the actual mobile policing data acquisition process) are effective, as they conform to the best-practice process in terms of time and order of activities. If the result is <1, there are problems with the process, and critical tasks should be investigated. The metric is calculated according to Eq. (1):

$$E_p = \frac{\sum_{i=0}^{n} length(lcs_i) * |caselist_i|}{|cases| * |task|} \quad using \quad i = |result| \tag{1}$$

The Eq. (1) Automated metric calculation for *usage effectiveness.*

3. **Visualization of results**
   The visualization component presents the calculated metrics and other information in an appropriate dashboard. Other information displayed in this dashboard includes information about the conformance of as-is processes presented in the form of (sub-)sequences, and all variants of the process documented in the usage event logs.

4.1.3 Technical Aspects of the Artifact's Instantiation

To ensure proper use of the artifact, a software prototype was developed based the findings of our design journey. In the first step, we designed an appropriate system architecture. The prototype is a software artifact that primarily illustrates the concept and is the basis for the further development of the artifact. The software prototype was developed as a web application.

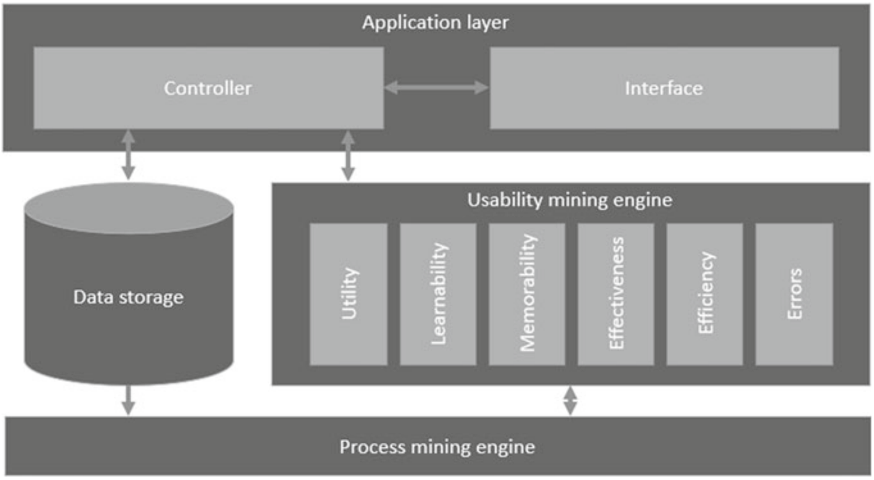


Fig. 8 Usability mining solution architecture from the third design iteration

For the implementation, we needed a suitable process mining engine. We used an R-based solution called *bupaR*[2] in the third design iteration. The developed application is a "classic" web application with a client server architecture. During the implementation, we set a high value on the possibility of integrating additional components into the artifact in the future. The individual components were developed using *ShinyR* and *Shiny Dashboard*,[3] which are extensive packages for setting up a web app and supporting quick creation of interactive interfaces. We used a file-based data model to guarantee a high level of autonomy and quick operational use of the prototype. In addition to the *bupaR* package, the *QualV* package was used for the sequence analysis. To create interactive diagrams, we use the JavaScript-based libraries *plotly* and *ggplot2* in combination. We also modified the *bupaR*-generated process models with the library *svg-pan-zoom* to ensure ease of navigation through the graphs.

The technical overview of the software prototype is shown in Fig. 8. The application layer describes the interface for developers and usability experts.

Besides the user interface of the dashboard, we developed the usability mining engine that retrieves the relevant data from a controller component, which itself retrieves the data from the local storage. We also use a process mining algorithm for the automated calculation of every usability metric.

[2] https://www.bupar.net/.

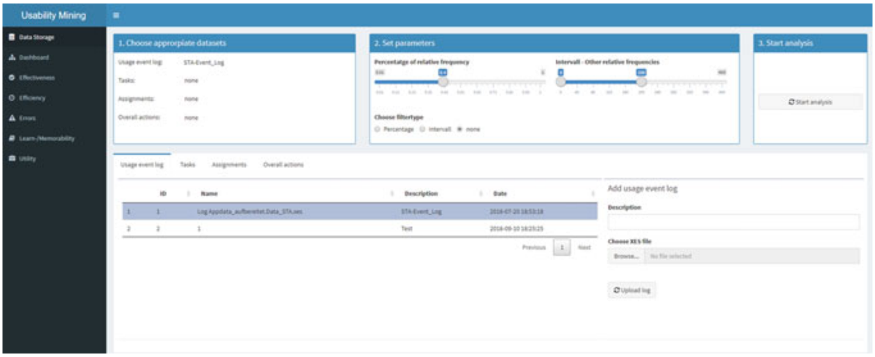
[3] https://shiny.rstudio.com/.

4.2 Application of Artifact(s)

In addition to the technical implementation, we present the concrete application of the implemented concept for the use case “data acquisition concerning criminal complaints.” The usage event logs of the mobile policing application in the pilot project were also captured using the logging script that we introduced in the journey section. We used the import screen shown in Fig. 9 to import the appropriate user interaction data, which was collected over a six-month period for this use case (24 data sets), into the usability mining solution.

The import view consists of three sections. Section 1 provides an overview of currently uploaded datasets. We need four datasets for the calculation: the log (“Usage event log”), a sequence of the defined usage process (“Tasks”), the assignment of the tasks and the click events (“Assignments”), and a list with all possible actions in the system (“Overall actions”). Section 2 provides selection fields based on the various parameters for the process mining algorithm. The user can choose which activities should be shown in the process model based on their occurrence. Section 3 allows the analysis of the implemented usability metrics to be started. For data management, the user should be able to import documents in XES or CSV format, save them, delete them if necessary, and obtain an overview of all available documents. The XES file contains the usage log to be examined, and the CSV files contain the activity grouping, the task, and the number of elements in the system. The input data for analyses should be stored separately from each other so they are reusable for further analyses.

The system generates a process model based on the event log provided and enriches it with classic process mining information like duration times, frequencies, and an appropriate visualization. For each of the developed usability dimensions, a separate site provides the results and the visualization of the metrics. In addition, a key measure can be determined and displayed for each dimension. In the *usage effectiveness case* presented later, we find an *effectiveness* value of 0.33 (Fig. 10). Under the assumption that the analysis of real data can lead to unforeseen deviations



1. Choose approprpiate datasets

Usage event log: STA-Event_Log
Tasks: none
Assignments: none
Overall actions: none

Percentatge of relative frequency

Intervall - Other relative frequencies

Choose filtertype: ○ Percentage ○ Intervall ● none

Usage event log | Tasks | Assignments | Overall actions

	ID	Name	Description	Date
1	1	Log Appdata_aufbereitet.Data_STA.xes	STA-Event_Log	2018-07-20 18:53:18
2	2	1	Test	2018-09-10 18:25:25

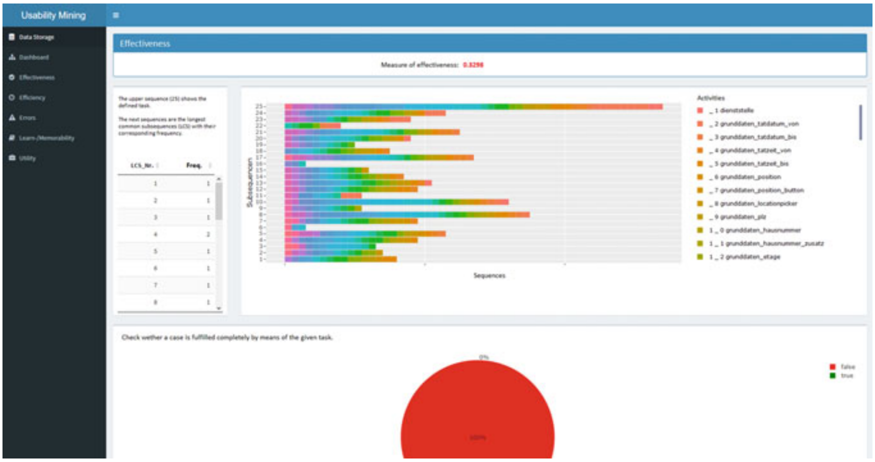
Add usage event log

Description

Choose XES file

No file selected

Fig. 9 Usability mining tool: data import screen


Usability Mining
Data Storage
Dashboard
Effectiveness
Efficiency
Errors
Learn-/Memorability
Utility
Effectiveness
Subsequencen
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
Sequences
Activities
🟥 _ 1 dienststelle
🟥 _ 2 grunddaten_tatdatum_von
🟥 _ 3 grunddaten_tatdatum_bis
🟧 _ 4 grunddaten_tatzeit_von
🟧 _ 5 grunddaten_tatzeit_bis
🟧 _ 6 grunddaten_position
🟧 _ 7 grunddaten_position_button
🟧 _ 8 grunddaten_locationpicker
🟨 _ 9 grunddaten_plz
🟨 1 _ 0 grunddaten_hausnummer
🟨 1 _ 1 grunddaten_hausnummer_zusatz
🟩 1 _ 2 grunddaten_etage
0%
100%
🟥 false
🟩 true

Measure of effectiveness: 0.3298

The upper sequence (25) shows the defined task.

The next sequences are the longest common subsequences (LCS) with their corresponding frequency.

LCS_Nr.	Freq.
1	1
2	1
3	1
4	2
5	1
6	1
7	1
8	1

Check wether a case is fulfilled completely by means of the given task.

Fig. 10 Usability mining tool: screen for the measurement of effectiveness

from the theoretical concept and that there is no one correct result but a multitude of correct results, the presentation of the analysis' results should be interactive and explorable (Günther & van der Aalst 2007).

Figure 10 presents the measured *effectiveness* and the task-execution sequences. One sequence effectiveness screen shows all activities of the complete process ("Task"), while others present the LCS of all twenty-four recorded usage process instances. The screen also shows the frequencies of each calculated LCS, and users can access an overview of detailed information by hovering the pointer over the screen's elements, such as the corresponding cases. The screen also provides a pie chart that shows how often a task was successfully executed. In this particular use case, no cases were completely performed in the intended way; every sequence and subsequence was either not completed or not performed in the intended way.

4.3 Evaluation of Artifact(s)

We applied the developed artifacts to the real-world scenario of the pilot project to validate and evaluate them. This section presents the results of using the prototype and demonstrates the artifact's feasibility and the value added (Gregor & Hevner 2013). In our case, the functional feasibility of an innovative solution for a previously unsolved problem is shown, along with additional insights into existing design problems. For this purpose, the usage data on the mobile policing application we captured was analyzed with the help of the prototype. The automatically generated usability metrics should also be compared with the findings of manual in-depth analyses of the developed usage models to determine the explanatory power and the informative value of the automatically computed metrics.

The mobile policing application is intended to accelerate the recording of relevant data in the field and, thus, to improve the administrative process and reduce its costs. We investigated the case of capturing and analyzing data concerning criminal complaints using a data set with twenty-four user-interaction logs.

First, the data available in CSV format was prepared. Then incorrectly formatted entries were corrected, and the labels, especially the activity labels, were normalized to ensure the resulting process models are easy to understand. The entries were then converted to the XES format using Disco and saved in application-specific event logs. Each event has the following attributes: caseID, timestamp, activity, location, form type, and data origin. We analyzed the data under the assumption that the group of users is stable and that the individual users have approximately the same level of experience after their introduction to the mobile policing app at the beginning of the project. To create the data entries, we assigned an interface element to each activity in the log. The target model, which describes the recording of the criminal complaint using the app, was defined as a task made up of individual interactions and the sequence of using the interface elements.

The automatically calculated value of *effectiveness* and the visualization of the sequence diagrams in Fig. 10 already indicate that the execution quality of the task in our example case can be significantly improved. The score indicates that an average of 33% of the defined activities were executed in the intended order in the use case example and that none of the twenty-four cases fulfills the task completely; even the "best" LCS reaches less than two-thirds of the target. None of the process instances contains all of the defined steps. This result can be traced back to activities that never occur in the target model, as they are seldom used in real-life cases. Thus, there is a considerable discrepancy between the target model and the actual executions, indicating potential for improvement in the application's usability. Next, we made several improvement recommendations and suggestions for further customizing of the mobile policing app based on the results provided by our prototype. One example was already illustrated in the journey section of this chapter. (see Fig. 1 and the related explanations.)

We concluded that the artifact, especially its software implementation, provides a feasible and valuable solution to the problem of automating the usability metric calculation based on process mining techniques.

4.4 Growth of Design Theory

The design of our usability mining solution was developed in the context of real-world projects with the German police. This project context had considerable influence on design decisions and the resulting artifact, especially the earlier design iterations. However, we believe that the current state of our usability mining artifact has considerable potential for many classes of business and governmental IS. The results of our design journey can also contribute to the growth and development of design theory of usability mining.

While there is no widely accepted definition of the term *design theory* and no consensus on what the constituent parts or components of a design theory should be—the discussion can be traced in, for example, Baskerville and Pries-Heje (2010), Fischer, Winter, and Wortmann (2010), Gregor (2006), Gregor and Jones (2007), Mandviwalla (2015), Suh (1998), and Walls, Widmeyer, and El Sawy (1992, 2004), and —there is consensus that design theory "says how a design can be carried out in a way which is both effective and feasible" (Walls et al. 1992, p. 37). We believe that the reference framework we developed is useful in many contexts of IS usage. Against this background, we are currently developing a more detailed presentation of the reference framework that takes the information presented here to a more generalized level. However, we can present some essential *design prescriptions* (DP) concerning how to develop a usability mining tool that is both effective and feasible and how to use it.

DP$_1$: The development of a scalable usability mining tool requires that the developer use adequate interfaces to acquire event log data automatically from the IS that is to be analyzed.

DP$_2$: The development of a versatile usability mining tool with all the functionality needed to go through the common usability mining lifecycle requires that the developer integrate automated analysis functionalities concerning the discovery of the usage model, the analysis of process execution metrics (etc.), and the computation of usability metrics.

DP$_3$: To provide useful application design recommendations based on the results of using a developed usability mining tool, the developer and system administrator should ensure that the users' system usage interactions are always documented in relation to the underlying tasks in a business process.

DP$_4$: To provide proper conformance checking results with a developed usability mining tool, those responsible for business process modeling should annotate additional business process-related information, such as the duration of the process or task, to the business process model.

DP$_5$: To provide useful design recommendations with a usability mining tool, the user should use all available information from the usage model, the process metrics, and the usability metrics.

The next section presents key findings and lessons learned from our design journey in the process of developing the usability mining reference framework and its instantiation in the proof-of-concept and the pilot project.

5 Key Lessons

As in many data science projects, data cleansing and preparation were a major issue in our design endeavor. Unusable data sets, target-oriented clustering, and obvious outliers must be addressed before meaningful usage models can be created and can serve as a basis for additional improvements of the business IS being analyzed.

Therefore, we had to ensure we included enough time in our schedule to deal with these issues.

Information about business processes and the information from the user interaction logs must be included if new usability information is to be generated, along with consideration of the underlying business process, which probably provides the most important benefits from using usability mining in organizations.

A particularly important success factor in using our usability mining solution in the various project phases was the granularity of the data. In our case, we first manually collected data in an "old fashioned" but also business-process-function-aware way (*Lap 1*). This data was useful in such analyses as overall duration time and other metrics in the context of business process analysis. (However, this data is not appropriate for detailed analyses regarding interaction problems.) Later, using the logging script, we had more fine-grained data, which could support our usability analyses.

Besides the granularity of the user-interaction data, the addition of relevant metadata can be useful, although such data were not part of the projects described here for reasons of privacy. User data like age and department could serve in additional investigations. The collection of other information can also be helpful in classifying process instances and the surrounding circumstances of the process instance execution. For example, when a user collects data in a noisy environment like a highway, the noise could cause certain parts of the process to take a longer time to complete, as noise can affect concentration and communication.

Another important success factor was the use of a tailored process mining solution in later iterations. In the first lap, we used an out-of-the-box process discovery solution. The detection of potential usability problems that resulted from using this tool was promising. However, the necessary detailed analysis could be done only using significant manual effort, so there was a demand for the automated detection of usability problems based on established usability engineering methods, especially the metrics. Therefore, we provided a framework and used adaptable process mining functionalities in our instantiation, which can be extended further. The goal is to enable users of the framework to build on its existing components to develop new approaches regarding new data sources or new usability metrics. There is also a demand to use standards like XES for the log generation and to provide concepts and methods to extend such standards.

Clearly, every solution can be developed further, which is also the case with the artifact presented here and certain details concerning the automated calculation of usability metrics. We are working on a more detailed presentation, definition, and implementation of the most common usability metrics, which can serve for automated usability analyses.

Acknowledgments The research described in this chapter was supported in part by a grant from the German Federal Ministry of Education and Research (*Bundesministerium für Bildung und Forschung*, *BMBF*), project name: "RUMTIMe: Real-Time Usability Improvement based on Process Mining," supported by the Software Campus Initiative. The research was also supported in part by a grant from the German Federal Ministry of Education and Research, project name: "ProPlanE—Echtzeit-Process-Mining in der Produktionsplanung mit Kundenbezug," support code FKZ

S01IS16040A. The authors of this chapter thank Matthias Adams for supporting the development and furthermore all project partners for their cooperation. Our special thanks go to all involved ministries and police authorities in Saarland and Rhineland-Palatinate, Accenture Deutschland GmbH, Avanade Deutschland GmbH, icomedias GmbH and Microsoft Deutschland GmbH.

References

Adams, K. M. (2015). *Understandability, usability, robustness and survivability. Nonfunctional requirements in systems analysis and design* (pp. 201–220), Heidelberg.

Baskerville, R., & Pries-Heje, J. (2010). Explanatory design theory. *Business and Information Systems Engineering, 2*(5), 271–282.

Batra, D., & Srinivasan, A. (1992). A review and analysis of the usability of data management environments. *International Journal of Man-Machine Studies, 36*(3), 395–417.

Dadashnia, S., Niesen, T., Fettke, P., & Loos, P. (2016a). Towards a real-time usability improvement framework based on process mining and big data for business information systems. *Tagungsband Multikonferenz Wirtschaftsinformatik (MKWI-16),* Ilmenau, Germany.

Dadashnia, S., Niesen, T., Hake, P., Fettke, P., Mehdiyev, N., & Evermann, J. (2016b). Identification of distinct usage patterns and prediction of customer behavior. In *Sixth International Business Process Intelligence Challenge (BPIC'16) located at BPI Workshop/BPM* 2016, Rio de Janeiro, Brazil.

Dadashnia, S., Houy, C., & Loos, P. (2017). Mobile Verkehrsunfallerfassung bei der Polizei im Saarland—Zur wissenschaftlichen Begleitforschung des Projektes VU-App. *Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI)—IWi-Heft 204.* Saarbrücken, Germany.

Fischer, C., Winter, R., & Wortmann, F. (2010). Design theory. *Business and Information Systems Engineering, 2*(6), 387–390.

Geng, R., & Tian, J. (2015). Improving web navigation usability by comparing actual and anticipated usage. *IEEE Transactions on Human-Machine Systems, 45*(1), 84–94.

Green, T. R. G., & Petre, M. (1996). Usability analysis of visual programming environments: A cognitive dimensions framework. *Journal of Visual Languages and Computing, 7*(2), 131–174.

Gregor, S. (2006). The nature of theory in information systems. *MIS Quarterly, 30*(3), 611–642.

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly, 37*(2), 337–355.

Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the AIS, 8*(5), 312–335.

Günther, C. W., & van der Aalst, W. M. P. (2007). Fuzzy mining—Adaptive process simplification based on multi-perspective metrics. In G. Alonso, P. Dadam, & M. Rosemann (Eds.), *Business Process Management (BPM)* (pp. 328–343), Berlin.

Harms, I., & Schweibenz, W. (2000a). Testing web usability. *Information Management & Consulting, 15*(3), 61–66.

Harms, I., & Schweibenz, W. (2000b). Usability engineering methods for the web. Results from a usability study. In G. Knorz, & R. Kuhlen (Eds.), *Informationskompetenz—Basiskompetenz in der Informationsgesellschaft. Proceedings des 7. Internationalen Symposiums für Informationswissenschaft* (ISI 2000) (pp. 17–30), Dieburg, Germany.

Hilbert, D. M., & Redmiles, D. F. (2000). Extracting usability information from user interface events. *ACM Computing Surveys, 32*(4), 384–421.

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

Houy, C., Gutermuth, O., Dadashnia, S., & Loos, P. (2019). Digitale Polizeiarbeit. In T. Klenk, F. Nullmeier, & G. Wewer (Eds.), *Digitalisierung in Staat und Verwaltung,* Wiesbaden.

ISO:9241. (1998). Ergonomic requirements for office work with visual display terminals. Part 11: Guidance on usability.

Ivory, M. Y., & Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys, 33*(4), 470–516.

Mandviwalla, M. (2015). Generating and justifying design theory. *Journal of the AIS, 16*(5), 314–344.

Montero, F., González, P., Lozano, M., & Vanderdonckt, J. (2005). Quality models for automated evaluation of web sites usability and accessibility. In J. Vanderdonckt (Ed.), *International COST294 workshop on user interface quality models* (pp. 37–43). Rome: Italy.

Nielsen, J. (1993). *Usability engineering,* Boston.

Saleh, A., Ismail, R., & Fabil, N. (2017). Evaluating usability for mobile application: A MAUEM approach. In *Proceedings of the 2017 International Conference on Software and e-Business* (pp. 71–77), New York, NY, USA.

Schuller, B., Althoff, F., McGlaun, G., Lang, M., & Rigoll, G. (2002). Towards automation of usability studies. In A. El Kamel, K. Melloui, & P. Borne (Eds.), *2002 IEEE International Conference on Systems, Man and Cybernetics.*

Seffah, A., Donyaee, M., Kline, R. B., & Padda, H. K. (2006). Usability measurement and metrics: A consolidated model. *Software Quality Journal, 14*(2), 159–178.

Suh, N. P. (1998). Axiomatic design theory for systems. *Research in Engineering Design, 10*(4), 189–209.

Thaler, T. (2014). Towards usability mining. In E. Plödereder, L. Grunske, E. Schneider, & D. Ull (Eds.), *Informatik 2014* (pp. 2269–2280), Innsbruck, Austria.

Thaler, T., Maurer, D., De Angelis, V., Fettke, P., & Loos, P. (2015). Mining the usability of business process modeling tools: Concept and case study. In J. Mendling, & J. vom Brocke (Eds.), In *Proceedings of the Industry Track at the 13th International Conference on Business Process Management 2015 (BPM 2015)* (pp. 152–166), Innsbruck, Austria.

Tullis, T., & Albert, B. (2008). *Measuring the user experience: Collecting, analyzing, and presenting usability metrics*, Amsterdam.

van der Aalst, W. (2012a). Process mining. *Communication of the ACM, 55*(8), 76–83.

van der Aalst, W. M. P. (2012b). Process mining: Overview and opportunities. *ACM Transactions on Management Information Systems, 3*(2), 7:1–7:17.

van der Aalst, W. M. P., & Weijters, A. J. M. M. (2004). Process mining: A research agenda. *Computers in Industry, 53*(3), 231–244.

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an information system design theory for vigilant EIS. *Information Systems Research, 3*(1), 36–59.

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (2004). Assessing information system design theory in perspective: How useful was our 1992 initial rendition? *Journal of Information Technology Theory and Application, 6*(2), 43–58.