



HAL
open science

Data-Driven Usability Test Scenario Creation

Maikel L. van Eck, Else Markslag, Natalia Sidorova, Angelique Brosens-Kessels, Wil M.P. van Der Aalst

► **To cite this version:**

Maikel L. van Eck, Else Markslag, Natalia Sidorova, Angelique Brosens-Kessels, Wil M.P. van Der Aalst. Data-Driven Usability Test Scenario Creation. 7th International Conference on Human-Centred Software Engineering (HCSE), Sep 2018, Sophia Antipolis, France. pp.88-108, 10.1007/978-3-030-05909-5_6 . hal-02270717

HAL Id: hal-02270717

<https://inria.hal.science/hal-02270717v1>

Submitted on 26 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Data-driven Usability Test Scenario Creation

Maikel L. van Eck¹, Else Markslag², Natalia Sidorova¹, Angelique Brosens-Kessels², and Wil M.P. van der Aalst^{3,1}

¹ Eindhoven University of Technology, The Netherlands

{m.l.v.eck,n.sidorova}@tue.nl

² Philips, The Netherlands

{else.markslag,angelique.kessels}@philips.com

³ RWTH Aachen University, Germany

wvdaalst@pads.rwth-aachen.de

Abstract. In this paper, we present a data-driven approach to enable the creation of evidence-based usability test scenarios. By utilising product usage data to create usability test scenarios, we aim to improve the reliability of the test results and to provide better insights into product usability. The approach consists of four elements: the collection of product usage data, the transformation of these data into logs of user activities, the creation of models of user behaviour, and the guided creation of usability test scenarios based on the models. We discuss the challenges that can be encountered when applying this approach based on our experiences with two case studies in product development. We have created a prototype scenario planning tool and performed a preliminary evaluation of the tool with usability engineers working at Philips Healthcare. The evaluation shows that tool-supported evidence-based usability test creation would be valuable in their daily work.

Keywords: User-Centered Design Usability testing Data-driven design Process mining

1 Introduction

Usability is an important quality that a product needs to possess in order to be successful [19]. If a product is usable, then its users can do what they want to do in the way they expect without hindrance or hesitation. Unfortunately, reports from the industrial practice indicate that it is not straightforward to ensure the usability of a newly developed product [2].

User-Centered Design (UCD) [15] has been devised to cope with this challenge when developing interactive systems [2, 19, 20]. The central principles of UCD are an early focus on users and their activities, the evaluation and measurement of product usage, and an iterative design process. Through this approach the findings from user testing related to usability and user experience can be used to inform the designer of a product in a relevant manner [2]. By focussing on the user of a product, UCD aims to consciously incorporate usability at every step of the design process of a product.

The purpose of performing a usability test as part of UCD is the collection of empirical data to measure usability aspects in a reliable and objective manner and to identify design problems [2, 3, 19]. An important element of a usability test is the *test scenario*: the sequence of tasks and activities executed by the test participants. The closer the test scenarios represent reality, the more reliable the test results and the resulting insights into product usability [19].

The creation of usability test scenarios is challenging for several reasons. Test scenarios are usually created manually by usability engineers, covering a limited number of tasks and activities compared to the real behaviour of product users [6], so the creation is affected by the biases of the engineers. These biases can arise because the usability engineers generally know a lot about the product being tested and what the full set of possible product features is, but they may have limited knowledge regarding the way product features are actually used in practice, in which combinations and in what order [17]. This issue can lead to insufficient test coverage, with a focus of testing whether product improvements are effective in very few scenarios, without considering other product functions that may be negatively affected by the change. The challenge of limited test coverage is especially important in safety critical systems, where testing is done to ensure that there are no usability issues that can lead to hazardous situations [10]. Furthermore, finding the right participants to represent the target user group is essential [3], but not trivial if there exist heterogeneous subgroups. For example, test participants should be selected with different levels of training if training levels are found to strongly influence the set of product functions used by a user.

To address these challenges and improve the reliability of product usability testing, we propose a *data-driven approach* to create evidence-based usability test scenarios based on product usage data. An increasing number of products contain functionalities that collect product usage data that is sent back to the manufacturer [9, 17]. This type of data can be used to create models that represent the behaviour of the product users, using data science techniques such as process mining [1]. Such models provide insights into existing user behaviour [2, 16], which we can use to create usability test scenarios that accurately reflect product usage [4, 6]. Additionally, the models can help identify differences in behaviour between user groups to assist test participant selection. The usability engineers will still need their domain knowledge to create good test scenarios, but they can be assisted with observational data to make the right choices when deciding on test tasks and their ordering.

The main contributions of this paper are the following. We present a data-driven approach for the creation of evidence-based usability test scenarios, as shown in Fig. 1. Its four main phases are: (1) the collection of data regarding product use, (2) the transformation of the data into logs of user activities, (3) the creation of models of user behaviour using process mining techniques, and (4) the creation of usability test scenarios based on the models. We build upon earlier work related to specific phases of this approach in the context of the development of two different products within Philips [9, 11], in order to discuss the challenges that can be expected in practice for each phase of the approach. We have

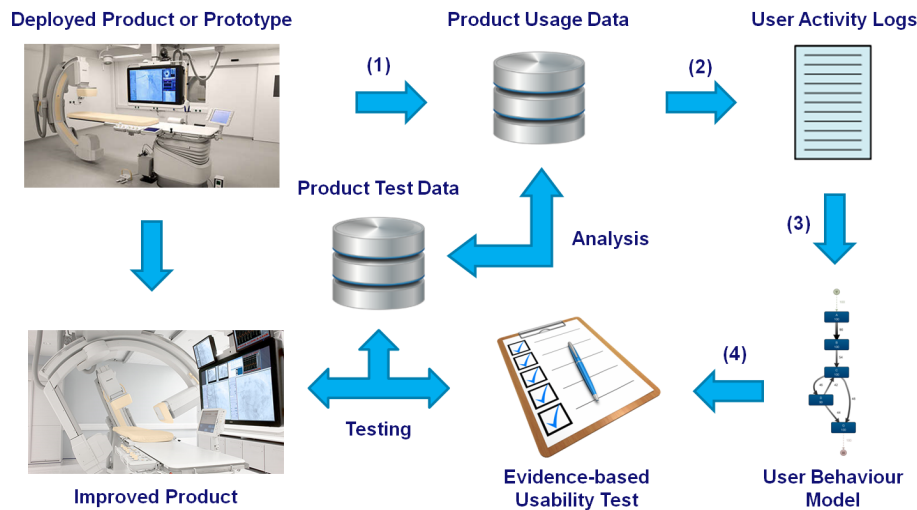


Fig. 1: A data-driven approach for the creation of evidence-based usability test scenarios. Its main phases: (1) product usage data collection, (2) user activity log creation, (3) user behaviour discovery, and (4) usability test scenario creation.

implemented a prototype scenario planning tool to support the approach, guiding usability engineers in the creation of evidence-based usability test scenarios. Finally, we discuss a preliminary evaluation of the developed scenario planner.

The structure of this paper is as follows. First, in Section 2 we review related work on usability testing, data and model-driven product development, and test scenario creation. In Section 3 we introduce the two cases studies and in Section 4 we briefly describe usability testing and the challenges of conducting reliable usability tests. In Section 5 we present the approach for the creation of evidence-based usability test scenarios. Then in Section 6 we describe the design and implementation of the scenario planning tool. In Section 7 we discuss the evaluation of the developed tool. Finally, in Section 8 we provide conclusions and discuss future work.

2 Related Work

In [17] a framework is proposed for post-deployment product data usage, describing the necessary development practices and organisational mechanisms to take advantage of the collection of product data in the development of software-intensive embedded systems. This framework can be used for a coarse classification of the level of product data usage within an organisation, but it does not provide details on how to achieve higher levels of product data usage maturity or the challenges encountered when implementing such levels of data usage.

In the areas of Model Driven Development (MDD) and Behaviour Driven Development (BDD) there exist various product development approaches that

argue for the need to model user behaviour [2, 14, 16, 20]. Task models for describing interactive systems are used during the early phases of the user-centered development cycle to gather information about user activities. Such models bring additional advantages to task analysis: they structure the gathered information about user activities and enabling the use of software tools to analyse and simulate user behaviour [4]. However, these approaches feature manual modelling of product user tasks and activities by the product developers, which is very time consuming [10, 14]. Instead, we propose to leverage product usage data and advances in analytics techniques [1] to mine models of user behaviour.

The main goal of our approach is to improve the reliability of usability testing, which is especially important in safety-critical systems [4, 10]. Demonstrating that the design of a medical device is compliant to relevant safety and usability requirements is a serious problem. The use of models of user behaviour can help to show that hazardous situations can be avoided or mitigated. However, as argued by [10], it is not feasible to expect regulators to construct models of products after their development, so ideally manufacturers produce models as part of their design process.

Several product development approaches argue for the creation of usability test scenarios based on models of user behaviour [6, 10, 14]. However, these models are based on the understanding of the developers of possible user behaviour, without explicit support of actual data regarding e.g. the frequency and ordering of use of specific product features. With these approaches, the test scenario generation is based on state machine models and their simulation with either exhaustive enumeration of all possibilities or scenarios that are randomly generated on the basis of the enabled user tasks in each state. Unfortunately, exhaustive usability testing is not possible in most organisations due to the balance of benefits and costs of testing [19]. Therefore, we propose a combination of domain expertise from usability engineers supported by statistical information based on product usage data to create evidence-based usability test scenarios.

Usability testing can also be seen in the context of requirements engineering and its related testing practices. Requirements engineering is concerned with the identification, modelling and documentation of requirements for a product or system and the context in which the system is used [18]. The aim of requirements engineering is to learn what to develop before the system design is finished. Usability characteristics can be part of the non-functional requirements of a product, however the focus of requirements engineering is usually on the functional requirements of a product. Validation of the implementation of requirements generally involves testing, but handling of non-functional requirements in such tests is often ill defined [18].

3 Case Studies

In the following, we discuss the challenges encountered in practice for each phase of the data-driven approach we propose. This discussion is supported by examples



(a) A medical imaging system used in minimally invasive treatments. (b) A smart baby bottle sleeve with an interactive app.

Fig. 2: Two products developed in Philips for which usage data is collected.

from two different product design case studies within Philips, which we were involved in during earlier work on specific phases of the approach [8, 11].

The first product is a *medical imaging system* shown in Fig. 2a. This X-ray system is used to perform minimally invasive treatments during medical procedures. During such procedures, it is sufficient to make a small incision through which an introduction element such as a catheter is used e.g. to place a stent in a blocked artery. The collection of product usage data and modelling of the clinical workflow during these procedures is described in more detail in [11].

The second product, shown in Fig. 2b, is a *smart baby bottle* sleeve equipped with various sensors and connected to an interactive app, described in [5, 8]. The sleeve sensors included a temperature sensor, a 3D accelerometer, a light intensity sensor, and a sound level sensor. The purpose of this product is to enable parents to collect personal and meaningful insights into the feeding of their baby. The app provides reports, data visualisations and recommendations to the parents based on the collected sensor data.

Although both products are developed by Philips, one at Philips Healthcare and the other at Philips Design, there are some very clear differences between the two case studies. The medical imaging system is an example of a very complex device with many functionalities, different ways in which it can be used and its usage requires extensive training. By contrast, the smart bottle is a simple device, with one main usage scenario that can be executed by any caregiver. The development cycle and the product maturity is also different, with the medical imaging system already deployed in many different hospitals and new improved versions of the system being developed iteratively, while the smart bottle is a new product for which the usage data comes from prototypes tested with a select group of parents. In addition, the type of data collected is different with the smart bottle recording sensor data measurements and the imaging system logging various messages for service and maintenance purposes.

There are also several aspects that the two products from the case studies have in common. Both are internet connected products that have functionalities to send usage data back to the manufacturer, which is essential for the first phase

Steps	Task	Instruction
	- Scenario	As a physician, today you will perform a Cerebral Coiling for an aneurysm on a biplane X-ray system. The next patient, Mark Smith, is now on the table in the exam room.
1	Correct identification of patient	Confirm if the correct patient is selected so the procedure can start.
	- Scenario	As we will be working in the exam room, put on the Lead Apron.
2	Correct identification of live image	Where do you expect to see Live Frontal and Lateral X-ray images here in the Exam room?
	- Scenario	In this case, access will be gained via the groin.
3	C-arm motion	Turn the Frontal C-arm to the groin and move the table if necessary so that you can visualize the groin on the screen.
4	Change Source Image Distance	Move the Frontal detector closer to the patient's body.
5	Activate fluoro with footswitch	Now use Frontal fluoro to place the guiding sheath.

Fig. 3: Example of a part of a test scenario for a a medical imaging system.

of our approach. Both products also have strong safety requirements. For the imaging system used in a medical environment this is evident, but for the smart bottle it is also important e.g. that the reported food temperature is accurate to prevent burns.

4 Usability Testing

The purpose of performing a usability test is to identify design problems through the collection of empirical data measuring usability aspects in a reliable and objective manner [2,3,19]. The basic elements of a usability test are the following: a set of *goals* the product users aim to achieve, the corresponding *tasks* or *activities* involving the product through which they aim to achieve those goals, and an accurate *representation of the actual working environment* or the context in which the product is used. These elements can be combined into test scenarios that are executed by a representative sample of end users during a usability test. An example of a sequence of tasks in a test scenario is shown in Fig. 3.

The test scenarios executed in a usability test are meant to imitate actual work that the participant would perform using the product and therefore they should be realistic. For example, when testing a medical imaging system, a test scenario is only mirroring realistic operation if the user first moves the scanner to the correct position and then captures an image. During validation of the product functionality, these tasks could also be performed in reverse order, but

this would not reflect normal operational behaviour. A test scenario adds context to the tasks that the testers perform and provides them with a motivation to carry out each task. With a realistic scenario, participants will find it easier to stay in their role and overcome hesitation during their use of the product and the closer that the test scenarios represent reality, the more reliable the test results and the resulting insights into product usability [19].

In addition to providing insights on product usability, usability tests can also form an essential part of the testing for product safety [4, 10]. In fact, user error has been a significant factor in over 50.000 adverse event reports, including at least 500 deaths, between 2005 and 2009 related to medical infusion pumps in the United States [10]. However, to demonstrate the safety of a product it is not sufficient to only show that each task can be safely executed in isolation because some safety problems may only manifest through a succession of actions. For example, making several x-ray images with a medical imaging system may result in a safety issue if the setting of the radiation dose used per image has been increased beforehand to a level not meant for making many successive images. Therefore, it is important to test the entire context in which the tasks occur with realistic test scenarios that consist of successions of actions that can also be expected to occur in practice [6].

Unfortunately, creating realistic test scenarios can be challenging [2, 6, 14]. To create a realistic test scenario, the usability engineer needs to have in depth knowledge of the product features, the tasks performed while using the product and the context in which the product is used. The list of tasks has to match those that the product users would perform to achieve the intended goals of the product, both in content and the order in which they have to be performed during the test. However, usability engineers on the product development team may not be familiar with all possible variations of use that occur in practice [17]. For example, the way in which medical imaging systems are used differs per country and sometimes even per hospital. This means that, especially for complex products that require significant training or education to use, it can be challenging to create test scenarios that accurately represents how the intended target audience will use the final product in a real setting.

5 Approach

To address the issues presented above, we propose a data-driven approach that enables evidence-based usability test scenario creation. A graphical overview of the approach is shown in Fig. 1. This approach consists of the collection of data and knowledge regarding product use, the transformation of such data into logs of user activities, the creation of models of user behaviour, and the creation of usability test scenarios based on those models and statistical data.

5.1 Product Usage Data Collection

The first step in the User-Centered Design cycle is to understand the context in which the product is used [15]. This step is essential because you can only improve

EventTimestamp	FrStand_Blo	FrStand_Ca	FrStand_Dt	FrStand_Pv	FrStand_RD	FrStand_Sa	FrStand_ZR	Table_Ht	Table_Lt	Table_Lo	Table_Tl	Table_cr	Table_pv	Table_tia	Table_pva	BrakeEngaged	AbnormalBehaviorLAT	AbnormalBehaviorLNG
01-01-15 10:48:25	2820	-1.7	1190	0	0	PARK	-86.9	821	70	1074	0	0	-1.4	HORIZONTAL WORK	TRUE	FALSE	FALSE	
01-01-15 10:48:25	2820	-1.7	1190	0	0	PARK	-86.9	821	70	1074	0	0	-1.4	HORIZONTAL WORK	TRUE	FALSE	FALSE	
01-01-15 10:48:28	2820	-1.7	1190	0	0	PARK	-86.9	874	70	1074	0	0	-1.4	HORIZONTAL WORK	TRUE	FALSE	FALSE	
01-01-15 10:48:28	2820	-1.7	1190	0	0	PARK	-86.9	886	70	1074	0	0	-1.4	HORIZONTAL WORK	TRUE	FALSE	FALSE	
01-01-15 10:48:28	2820	-1.7	1190	0	0	PARK	-86.9	883	70	1074	0	0	-1.4	HORIZONTAL WORK	TRUE	FALSE	FALSE	
01-01-15 10:48:29	2820	0	1195	0	0	PARK	-86.9	908	70	1074	0	0	-1.4	HORIZONTAL WORK	TRUE	FALSE	FALSE	
01-01-15 10:48:29	2820	0	1195	0	0	PARK	-86.9	908	70	1074	0	0	-1.4	HORIZONTAL WORK	TRUE	FALSE	FALSE	
01-01-15 10:48:29	2820	-0.1	1195	0	0	PARK	-86.9	906	70	1074	0	0	-1.4	HORIZONTAL WORK	TRUE	FALSE	FALSE	

Fig. 4: Example of a machine log describing how the table position of a medical imaging system was changed by a surgeon during a procedure.

a product if you know exactly what the problems are that users encounter and what the underlying root cause of each issue is. Without understanding of the context of use, it is difficult to set up an effective usability test [19]. To understand how users behave when using the product in practice it is important to collect sufficient data and combine this with knowledge of the product itself [17].

In the context of the product data usage framework presented in [17], our approach assumes that the data usage maturity of the product is at least at the level of *Diagnostics*. This means that we assume a real-time, or close to it, collection of usage data that is effectively stored and accessible. The usage data is ideally linked to specific product functionalities so that it becomes possible to relate functions to user tasks. Evidence-based usability testing can then help to reach the maturity levels of *Feature Usage* analysis and *Feature Improvement*.

It is easier to collect data for a product that is already deployed and is now being improved or for which a successor is being created than for a completely new product. For truly new products, for which perhaps only prototypes exist, there may not be any usage data available to work with. However, if product prototypes are given to real users for testing purposes, as happened in the development of the smart bottle, then some usage data can already be collected early in the development cycle. Alternatively, if access to end users is really not possible, the developers can create artificial usage data by using product prototypes in a manner similar to end users during a test role-playing session.

There are different types of data that can be collected to help understand the context of use. For example, observational data of activities from field studies, subjective descriptions of users from reviews, sensor data measured by the product itself, or usage logs from an accompanying application or service. For the purpose of this approach, sensor data and product usage logs, like the one shown in Fig. 4, that are linked to product functions are the most useful types of data, as these machine logs provide objective information to quantify product feature usage. However, observational data can be used to relate the machine logs to actual user tasks, if this knowledge is not known to the developers. User reviews can be helpful in determining which tasks are problematic and important to be tested.

5.2 User Activity Log Creation

The different types of data collected above do not guarantee a clear view on the activities of the users. Depending on the type of data, it can be necessary to transform the raw product usage data into actual logs of activities [4].

Usability test scenarios can be seen as sequences of activities that a test participant has to execute. This means that we are also interested in obtaining such sequences of user activities from the field in order to get a better understanding of the user behaviour. Essential information to obtain is therefore which activities the users performed, in what order and how long they took.

Often, product usage data is collected for maintenance and service purposes or as a side-product of debugging functionalities [17]. In these cases, the usage data may not immediately reveal what people are doing exactly and a relation needs to be established between the product functions instrumented with logging and the actual user task during which the function is used. For example, in the medical imaging system different sensors can detect movement in specific directions of the table on which the patient is lying. However, the recorded data contains detailed technical information, as shown in Fig. 4. From this we need to deduce what task is executed by the user, e.g. specific movements positioning the patient under the scanner or adjusting the table to a convenient working position for the surgeon.

There are different techniques to transform low-level sensor data or logged events into higher-level user tasks and activities. There is a large body of work on activity recognition and complex event processing [7, 12]. For example, it is possible to group low-level events based on behavioral activity patterns in order to identify high-level activities that make sense to domain experts [13]. In the case of the smart bottle usage data, it was necessary to use techniques from the signal processing domain combined with clustering techniques to detect shifts in the sensor data that corresponded to actual user activities [9]. For the imaging system, the logged machine-generated events and diagnostic messages were related to user tasks through a combination of domain knowledge and user activity logs obtained from observational studies and self-reporting [11].

In some cases, it can also be necessary to modify the logging developed by the product designers in order to get a better view of what the users are doing. For example, one of the medical imaging systems developed by Philips has a sensor that detects whether a patient is currently lying on the operating table. However, the signal detected by the sensor was not logged in the data sent back to Philips. After changing the data logging, it is possible to recognise the moments where the patient is present on the table, which in turn allows for the recognition of the activities in the clinical workflow of a surgical procedure of putting the patient onto the table and removing the patient.

5.3 User Behaviour Discovery

Once there are activity logs then we can create models of user behaviour. This is often done manually based on the understanding of the developers regarding user goals and product functionalities, but this is very time consuming [10, 14]. Therefore, we propose to apply techniques from the field of process mining to automatically discover models of behaviour from activity data [1].

Process mining can be defined as the analysis of processes using the data recorded during their execution. A process in this context is a set of logically

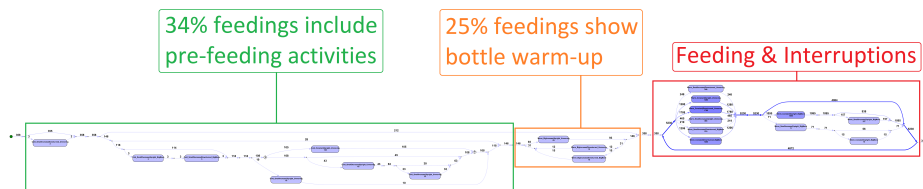


Fig. 5: A discovered end-to-end process model for the use of the smart bottle, showing which activities are performed and in what order. The model is annotated with statistics regarding choices and activity durations.

related tasks to achieve a certain goal. The data corresponding to the execution of a process can be captured in a log of events, where each event corresponds to the execution of a task at a specific point in time, possibly associated with other data. Hence, user activity logs can be seen as a particular type of event log.

Process discovery works by taking an event log and applying a discovery algorithm to produce a process model that represents the behaviour captured in the log. There are many different algorithms [1] such as the Alpha Miner, the ILP Miner and the Inductive Miner that have been implemented in the open-source process mining framework ProM [21]. Various commercial tools are also available that can discover process models from event logs [1].

The models discovered using a process discovery algorithm are often annotated with statistical data [1, 8]. The process model annotations often provide information on the number of occurrences of activities and the likelihood of an activity being followed by another activity. Many process mining tools also show statistics regarding the duration of activities. These annotated models can be used to gain insights into the likely flow of activities of people using a product and how much time is being spent doing what. For example, the model in Fig. 5 was discovered from smart bottle usage data [9] and it showed e.g. that many people filled the bottle before attaching the sleeve, resulting in limited visibility on pre-feeding activities.

There also exist process discovery techniques that aim to provide insights into the relation between different process artifacts [8]. These techniques are suitable for data from environments with multiple products or complex products that consist of multiple objects, sensors or modules that each generate data. The use of such techniques can make the resulting models of each artifact easier to understand than the complex behaviour of the entire system. We used these techniques in the creation of models for the medical imaging system [11] because of the complexity of the system and the difficulty in creating user activity logs. It was not possible to recognise all user activities in the machine data, so we used artifact-centric techniques to discover interacting models for different parts of the system behaviour and to obtain correlations between those artifact models and the activities that were detectable.

5.4 Usability Test Scenario Creation

After obtaining user behaviour models based on activity logs, we can use them to generate evidence-based usability test scenarios. The main challenge we address is in creating the list of tasks that is executed during the scenario. This is achieved by using the models annotated with flow statistics, activity frequencies and durations to provide a usability engineer with the information needed.

Different approaches and strategies exist for the generation of a test scenario from a task or process model [6, 10, 14], such as the creation of random test scenarios or exhaustive generation. Exhaustively generating all possible sequences of user behaviour from the models is possible in automated test settings, but not for usability testing with real end users. Test participants have limited availability and there is a cost associated to their employment [19]. There are techniques to simplify the behaviour in a model [6], but simplified models can still generate thousands of test cases. When creating random test scenarios, there is no guarantee that they will contain activities that are closely related to the area for which the developers have made product improvements. The risk of this is ineffective usability testing. Therefore, we propose an approach combining domain knowledge from usability engineers with evidence-based user behaviour models.

One combined strategy is to order a set of mandatory tasks in the most likely configuration according to the model. The usability engineer determines the tasks that are essential for the test and those that are most likely to be affected, based on the nature of the product improvement being tested, the features it affects and in what tasks those features are used. For example, if a product improvement has been developed for the medical imaging system that should make it easier to position the patient then usability testing should also focus on activities related to the patient positioning itself and those that are affected by either proper or incorrect patient positioning. The usability test scenario would then be the most likely path through the state-space of the model, given that it contains at least the critical tasks selected by the usability engineer.

Another combined strategy is through tool-supported interactive guidance of the usability engineer while they are creating the test scenario. While selecting tasks to be included in the test scenario, the engineer is presented with information from the model regarding the expected preceding or succeeding tasks and their frequency. This is the approach that we implemented in a scenario planner after discussions with usability engineers, as covered in the next section.

Given a usability test scenario, it is also important to select the appropriate test participants [3]. The discovered models of user behaviour can assist in this if task frequency and ordering statistics are split by user subgroup. If for example certain features in the medical system are used more frequently by specific types or surgeons then product changes affecting those features should be tested with these subgroups. Alternatively, if the models show that a specific group of users perform undesirable or incorrect behaviour more frequently than other groups of users for a certain task then product improvements affecting this task should also be tested with these users that potentially benefit most.

Table 1: An overview of the main requirements of the usability test scenario planning tool.

Requirement	Priority
1. Connection to a static clinical workflow model of user activities with statistical annotations	Must have
2. The current structure and essential elements of the test scenarios are reflected	Must have
3. Possibility to choose the type of imaging system for the usability test	Must have
4. Possibility to choose the type of clinical professional participating in the usability test	Must have
5. Possibility to choose the type of department from the hospital for the usability test	Must have
6. Possibility to choose the type of clinical procedure executed during the usability test	Must have
7. Possibility to create a test scenario for a part of the clinical workflow, not only the whole workflow	Should have
8. Possibility to choose the region where the hospital is located for the usability test	Should have
9. The expected duration of the test scenario is shown	Could have
10. Integration with Philips database systems whose data the static workflow model is based on	Won't have

6 Scenario Planner

Based on the approach presented in Section 5 and discussions we had with usability engineers at Philips Healthcare, we developed a usability test scenario planning tool. The purpose of the scenario planner is to assist usability engineers when creating a usability test scenario by providing them with evidence-based information on the behaviour of product users.

6.1 Requirements Analysis

We performed exploratory research to determine the requirements of the scenario planner. Based on literature regarding usability testing, interviews with usability engineers, and observations in the field (observing both end-users of the medical imaging systems and usability engineers) we established a set of requirements for the scenario planner shown in Table 1.

A central concept of the scenario planner is its use of an internal workflow model to provide information to the usability engineer, based on the product usage data collected by Philips from medical imaging systems in the field. As Philips Healthcare collects data from its imaging systems located all across the world, used in various different hospital departments and for different types of procedures, there is also a need for the selection of the correct model of user behaviour depending on the specific context for which a usability test is being created. However, as the developed tool is only a prototype we did not aim for

live integration with the database systems and instead used a static snapshot of the data for the internal model of user activities, i.e. a model of the clinical workflow of using the imaging system.

6.2 Tool Concepts

Based on the requirements analysis several different concepts of the scenario planner were developed. The design process was iterative with feedback from three different usability engineers during the development cycles.

Paper prototypes were developed to discuss the concepts with usability engineers and to get an understanding of their way of working during usability test scenario creation. The first concept, shown in Fig. 6, involved an interactive selection of individual tasks based on observed sequences of activities and possible choices. The user interface was envisioned to show the different phases of a clinical workflow and for each phase a model annotated with transition probabilities would be presented. By clicking on user tasks they would be included in the test scenario. The feedback from the usability engineers for this concept was that showing a complete overview of all possible activities and their relations with corresponding data could result in a very complex model and make it difficult to put together a test scenario.

The second concept in Fig. 7 is a presentation of a number of pre-defined usability test scenarios based on the most likely sequences of user behaviour. The feedback from the usability engineers on the second concept was that, although easy to use, it would not provide much support for the creation of evidence-based usability test scenarios in cases where additional editing is necessary to include essential activities that were not part of the suggested activity flows.

After these design iterations, another paper prototype was built that combines the idea of presenting the user with a likely sequence of user behaviour and the option to interactively modify the scenario based on data. When modifying the scenario, the tool provides the user with suggestions for adding specific activities based on the part of the scenario being changed and data on e.g. the most likely next activity. The prototype was tested with two usability engineers and based on their feedback the implementation of the scenario planner was started.

6.3 Implementation

The scenario planner with limited functionality was implemented as a web application. The application was created using HTML, CSS, JavaScript and PHP. The goal of the prototype development was to allow usability engineers not involved in the development process to experience the idea of creating evidence-based usability test scenarios in order to provide feedback on the usefulness of the scenario planner.

A screenshot of the final implementation is shown in Fig. 8. The user can set the filter to restrict the usability test scenario to a specific context and is then presented with a frequent sequence of activities for each phase of the clinical workflow. Activities can be removed or added after existing activities, in which

Workflow

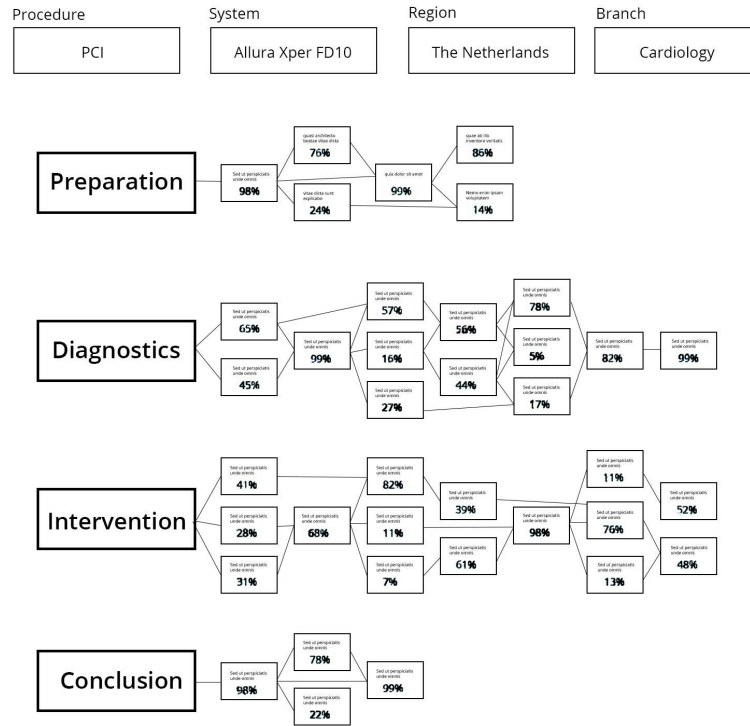


Fig. 6: A design concept of the selection of test activities. The activities are divided over the different clinical workflow phases and their frequency of occurrence is shown. The user selects which activities to include in the test scenario.

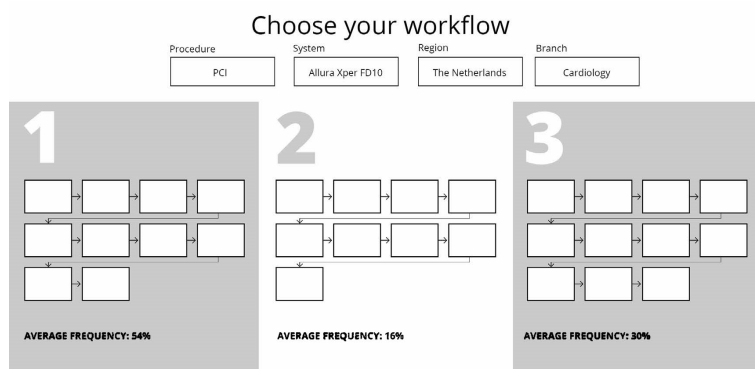


Fig. 7: A design concept based on the selection of a sequence of user activities based on its frequency of occurring in the field.

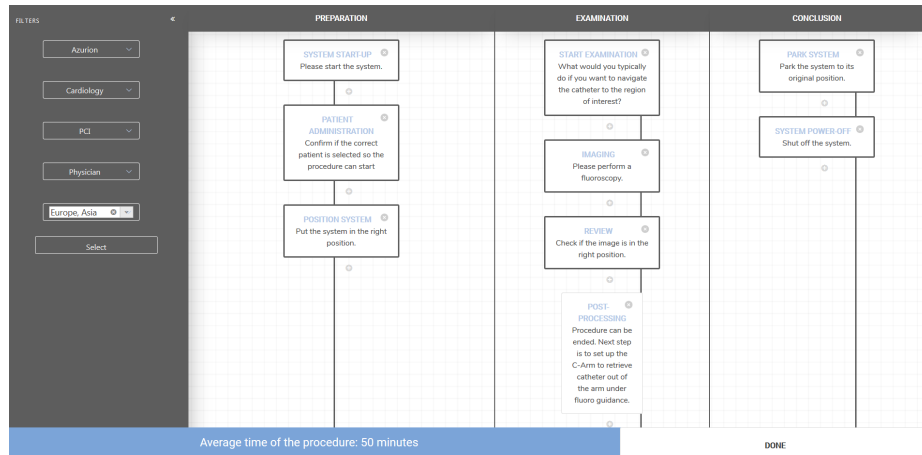


Fig. 8: The user interface of the interactive prototype implementation of the scenario planner. Filtering functionality is shown on the left side and the different phases of the clinical workflow are shown with their specific activities.

case the tool will suggest appropriate activities to insert at that specific point depending on the most likely behaviour for users corresponding to the chosen filter settings. Clicking on an activity provides a pop-up with additional information, displaying e.g. more detailed instructions for the test participants, task frequency information and expected duration. The total estimated time to execute the test scenario is shown at the bottom of the user interface.

7 Evaluation

The functionality of the scenario planner was evaluated during a usability test of the tool itself. The goal was to get feedback on the usefulness of the scenario planner and to determine design inconsistencies and usability problem areas within the user interface and content areas.

7.1 Test Setup

The usability test was conducted on-site at a Philips location and through Skype. The session captured the participants' navigational choices, task completion scores, comments, questions and feedback. At the end of the test every participant was asked what they like and dislike about the scenario planner, if they miss certain product functions and if they think that the scenario planner fits in their daily work when preparing usability tests.

All participants work as a usability engineer at Philips Healthcare and have experience with usability tests. Nineteen participants were invited and nine participants took part in the test, of which three were already involved in the

Table 2: The usability engineers participating in the evaluation were asked to complete these tasks using the scenario planner.

Step	Task	Instruction	Expected results
1	Start creating scenario	You have a usability test coming up. You are going to test an Azurion system with a physician from the cardiology department.	User uses the filters to give specifications to the scenario.
2	Add a task	You want to add a task between after review because you want to test a feature that could be used in that phase of the test. How would you do that?	User clicks on a task and gets to see two separate tasks. They will choose one of these.
3	Delete a task	You don't want to test reporting. Try to remove this.	User clicks on tasks and deletes it while getting the alert that they're deleting a frequent task.
4	Access task details	You want to know more about the imaging task. Try to find more info.	User clicks twice on the task and sees the overlay with the information.
5	Check test duration	What is the duration of the test you want to do?	User notices the time at the bottom of the page and knows how to change it.
6	Output results	You are done! Try to get the output of the scenarioplanner.	User clicks at done and notices the alert that they're done.

design process of the scenario planner. The test participants who were not involved in the design process were given a short description how the scenario planner came about and what it can be used for.

The participants were asked to perform the tasks described in Table 2 using the scenario planner. These tasks form a simple scenario where the test participants are asked to use the scenario planner to create a usability test scenario for the use of a medical imaging system developed at Philips. Participants were scored per task on completion success rate and their time spent.

7.2 Results

The ability of participants to complete a task without critical errors was rated according to the following four point scale:

1. User cannot complete the task and needs help.
2. User completes the task after a hint from the moderator.
3. User completes the task after some tries.
4. User completes the task immediately.

Participant	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
1	4	4	4	2	2	4
2	4	4	4	2	4	4
3	4	4	4	4	2	4
4	4	4	4	2	2	4
5	4	4	4	2	2	4
6	3	4	4	2	3	4
7	4	4	4	2	4	4
8	4	3	4	2	2	4
9	4	4	4	3	2	4
Success	9	9	9	2	3	9
Completion Rates	100%	100%	100%	20%	30%	100%

Fig. 9: The task completion scores for each participant on each task. A task scored with a 3 or 4 is considered successfully completed.

Table 3: An overview of the main recommendations given by the usability engineers after testing the usability test scenario planning tool.

Recommendation	Priority
1. More intuitive method to access the additional information for activities	High
2. Clear specification in the filtering UI regarding optional and required information	High
3. Better integration of the expected duration of the test scenario in the UI	High
4. Undo functionality to revert accidental deletions of activities	Medium
5. Options to add custom activities to the scenario that are not present in the data and to edit the description or instructions belonging to activities, to reduce the need for editing in Excel afterwards	Medium
6. Save and load functionality for test scenarios	Medium
7. Drag and drop functionality to move tasks around in the scenario planner	Medium
8. Provide more information regarding possible safety related issues or risks	Low

An overview of the completion scores of each participant for each task is shown in Fig. 9. Tasks scored with a 3 or 4 are considered as successfully completed. All participants managed to successfully complete tasks 1, 2, 3 and 6. However, very few participants completed tasks 4 and 5 without assistance.

Different parts of the user interface were used in completing the different tasks. The participants managed to successfully use the filters in the side menu to specify the type of scenario they wanted to create. The subsequent modification of the scenario generated by the planner based on an internal workflow model was achieved through buttons placed near the tasks where the user wanted to make the change. These parts of the user interface were intuitive to use. However, the option to access additional task details, which required the user to double click on a task, was less intuitive. The reason why participants required assistance with task 5 was that they assumed that the user interface element used in this task was part of the screen recording device used in the evaluation.

The participants provided feedback on the user interface of the scenario planner. Based on this feedback, the a list of recommendations for changes in the scenario planner was established and prioritised, shown in Table 3. The priority was based on a combination of ease of implementation and impact. In general, they liked the option to export the final scenario to Microsoft Excel and the user interface design for the activity flow. However, accessing the additional task information was not intuitive and it was not entirely clear to all participants what elements on the data filtering were optional and which were required. The participants also indicated that they would like to be able to revert editing mistakes and to save and subsequently edit old usability test scenarios.

The participants also commented on the perceived usefulness of the tool. They were happy with the overall functionality and the concept of evidence-based usability test creation. Eight out of nine participants expressed that they would want to use the tool in their daily work. They indicated the value of knowing how often specific activities are performed and what other activities they are related to. However, in this usability test it was not possible to evaluate how effective the activity and flow statistics are in guiding the scenario creation, due to the fixed nature of the test scenario that the participants created. Some participants suggested that it would be even more helpful if detailed reports were available on activity statistics, e.g. the number of times an activity is executed per procedure type, per day and per hospital.

8 Conclusion & Future Work

In this paper, we have presented an approach to enable the creation of evidence-based usability test scenarios. The approach consists of several different parts: the collection of data on product use, the transformation of the data into logs of product user activities, the creation of models of user behaviour, and the guided creation of usability test scenarios. Based on this approach, we have created a prototype usability test scenario planning tool in co-creation with usability engineers working at Philips Healthcare.

The prototype has been evaluated through a usability test of the tool itself. Overall, the participating usability engineers were enthusiastic regarding the evidence-based creation of usability test scenarios and eight out of nine participants expressed that they would want to use the tool during their normal work. They were able to successfully create a usability test scenario using the tool and provided feedback and recommendations for the future development of the scenario planner, its user interface and the information provided by the tool.

One of the main limitations of our approach is that it requires an existing product for which usage data is available, or a prototype that has been deployed for testing. The limitations of the conducted evaluation of our approach are that the usability test scenarios created during the evaluation were based on a simplified model of the clinical workflow activities and that the created usability scenarios were not subsequently used during a real usability test of a medical imaging system. As a result, the quality of the evidence-based usability test

scenarios was not directly evaluated. To address these limitations a more mature implementation of the scenario planner and additional user evaluation are needed.

As future work, we propose an integration of evidence-based usability testing into a full data-driven product development approach. By collecting and analysing data throughout the design cycle it can become possible to identify patterns that correspond to incorrect or undesirable user behaviour. This could be the input for suggestions for additional product improvements, which can then be tested in a subsequent product design iteration. Based on the results of usability testing, the impact of product improvements on different usability aspects can then be measured and compared to the data available on the user behaviour with the previous product version.

Acknowledgements

This work has been conducted in the context of the IMPULS collaboration project of Eindhoven University of Technology and Philips: “Mine your own body”, and within the ENABLE-S3 project that has received funding from the ECSEL Joint Undertaking under Grant Agreement No 692455. This Joint Undertaking receives support from the European Union’s Horizon 2020 research and innovation programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway.

References

1. van der Aalst, W.M.P.: *Process Mining - Data Science in Action*, Second Edition. Springer (2016)
2. Bernhaupt, R., Palanque, P.A., Manciet, F., Martinie, C.: User-test results injection into task-based design process for the assessment and improvement of both usability and user experience. In: *Human-Centered and Error-Resilient Systems Development, HCSE and HESSD 2016, Proceedings*. pp. 56–72 (2016)
3. Billestrup, J., Bruun, A., Stage, J.: Usability problems experienced by different groups of skilled internet users: Gender, age, and background. In: *Human-Centered and Error-Resilient Systems Development, HCSE and HESSD 2016, Proceedings*. pp. 45–55 (2016)
4. Billman, D., Fayollas, C., Feary, M., Martinie, C., Palanque, P.A.: Complementary tools and techniques for supporting fitness-for-purpose of interactive critical systems. In: *Human-Centered and Error-Resilient Systems Development, HCSE and HESSD 2016, Proceedings*. pp. 181–202 (2016)
5. Bogers, S., Frens, J.W., van Kollenburg, J., Deckers, E., Hummels, C.: Connected baby bottle: A design case study towards a framework for data-enabled design. In: *Proceedings of the 2016 ACM Conference on Designing Interactive Systems, DIS '16, Brisbane, QLD, Australia, June 04 - 08, 2016*. pp. 301–311 (2016)
6. Campos, J.C., Fayollas, C., Gonçalves, M., Martinie, C., Navarre, D., Palanque, P.A., Pinto, M.: A more intelligent test case generation approach through task models manipulation. *PACMHCI 1(EICS)*, 9:1–9:20 (2017)

7. Cugola, G., Margara, A.: Processing flows of information: From data stream to complex event processing. *ACM Comput. Surv.* 44(3), 15:1–15:62 (2012)
8. van Eck, M.L., Sidorova, N., van der Aalst, W.M.P.: Discovering and exploring state-based models for multi-perspective processes. In: *Business Process Management - 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings.* pp. 142–157 (2016)
9. van Eck, M.L., Sidorova, N., van der Aalst, W.M.P.: Enabling process mining on sensor data from smart products. In: *Tenth IEEE International Conference on Research Challenges in Information Science, RCIS 2016, Grenoble, France, June 1-3, 2016.* pp. 1–12 (2016)
10. Harrison, M.D., Masci, P., Campos, J.C., Curzon, P.: Verification of user interface software: The example of use-related safety requirements and programmable medical devices. *IEEE Trans. Human-Machine Systems* 47(6), 834–846 (2017)
11. Hoornaar, T.J.: *Extracting Real-life Workflow Models from Relational Data and Using These to Generate Field-based Usability Testing Scenarios at Philips Healthcare.* Master's thesis, Technische Universiteit Eindhoven (2017)
12. Liu, Y., Nie, L., Liu, L., Rosenblum, D.S.: From action to activity: Sensor-based activity recognition. *Neurocomputing* 181, 108–115 (2016)
13. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P., Toussaint, P.J.: *Guided process discovery-a pattern-based approach.* Information Systems (submitted) (2018)
14. Mori, G., Paternò, F., Santoro, C.: CTTE: support for developing and analyzing task models for interactive system design. *IEEE Trans. Software Eng.* 28(8), 797–813 (2002)
15. Norman, D.A., Draper, S.W.: *User Centered System Design; New Perspectives on Human-Computer Interaction.* L. Erlbaum Associates Inc. (1986)
16. Ogata, S., Goto, Y., Okano, K.: Framework for relative web usability evaluation on usability features in MDD. In: *Human-Centered and Error-Resilient Systems Development, HCSE and HESSD 2016, Proceedings.* pp. 73–85 (2016)
17. Olsson, H.H., Bosch, J.: Towards data-driven product development: A multiple case study on post-deployment data usage in software-intensive embedded systems. In: *Lean Enterprise Software and Systems - 4th International Conference, LESS 2013, Galway, Ireland, December 1-4, 2013, Proceedings.* pp. 152–164 (2013)
18. Paetsch, F., Eberlein, A., Maurer, F.: Requirements engineering and agile software development. In: *12th IEEE International Workshops on Enabling Technologies (WETICE 2003), Infrastructure for Collaborative Enterprises, 9-11 June 2003, Linz, Austria.* pp. 308–313 (2003)
19. Rubin, J., Chisnell, D.: *Handbook of Usability Testing, Second Edition: How to Plan, Design, and Conduct Effective Tests.* John Wiley & Sons (2008)
20. Silva, T.R., Hak, J., Winckler, M.: Testing prototypes and final user interfaces through an ontological perspective for behavior-driven development. In: *Human-Centered and Error-Resilient Systems Development, HCSE and HESSD 2016, Proceedings.* pp. 86–107 (2016)
21. Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: XES, XESame, and ProM 6. In: *Information Systems Evolution,* pp. 60–75. Springer (2011)