Improving the user experience of an Intelligent Tutoring System for proving mathematical statements

Bór, Dorina

Award date:
2023

Eindhoven, 06-02-2023

Improving the user experience of an Intelligent Tutoring System for proving mathematical statements

by Dorina Bór

identity number 1579347

in partial fulfilment of the requirements for the degree of

Master of Science
in Human-Technology Interaction

Supervisors:
dr.ir.Rianne Conijn
Department of Industrial Engineering and Innovation Sciences, TU/e
dr.Jim Portegies
Department of Mathematics and Computer Science, TU/e

Table of Contents

Acknowledgements

First, I would like to thank both my supervisors for the incredible support I have received throughout the period of writing my master thesis. I would like to thank Rianne Conijn for ensuring the flexibility in the project that was necessary to provide the Waterproof team with helpful results and for guiding me in the whole process of writing such a large scale work. I would also like to thank Jim Portegies for prioritizing this project and being always available for questions and meetings. Further, I would like to thank Jelle Wemmenhove, who volunteered a lot of his time and contributed a lot with his comments to this paper as the unofficial third supervisor of the project.

Additionally, I would like to thank my parents, sister and therapist for being by my side in the difficult periods during the last months; my roommates for the occasional warm meals after long study days and my friends. I would like to specifically thank Jordi, Henk, Ariel and Marci for the pep talks and encouragement. You guys rock.

Abstract

Intelligent Tutoring Systems (ITSs) have recently gained more attention as they address the increasing demand for personalized education and computer-based tutoring. ITS evaluation is crucial in ensuring the success of ITSs. The literature of ITS evaluation has recently been mostly focused on system performance and learner performance. Less attention has been given to learner experience, the users of ITSs and the implementation context of ITSs in evaluation studies. Additionally, evaluation designs are mostly using pretest-posttest experimental methods and fail to utilize user-centered methodologies such as log data analysis or the examination of user onboarding. The aim of this study is to propose and test an ITS evaluation framework which is built on service design principles. An evaluation case study is conducted on Waterproof, an ITS designed to help students learn to prove mathematical statements. The case study is built around three areas of inquiry: (1) examining the user onboarding of Waterproof, (2) building an understanding of user engagement with Waterproof and (3) examining the circumstances of users abandoning Waterproof. The four studies conducted to answer these inquiries utilized methods such as a qualitative co-creation session, survey studies, in-person user testing and log data analysis. Outcomes of the studies successfully addressed inquiries 1 and 2 by identifying user pain points in the onboarding process and creating a process map of user actions in Waterproof supplemented by qualitative data about means of interaction. Refinements in defining activities for log data analysis could help improve exploring inquiries similar to inquiry 3 in future studies. The outcomes of the study provide a proof of concept for the new service design-based ITS evaluation framework and contextualizing ITS evaluation; however, further case studies should be conducted to improve the generalizability of the framework and improve its applicability in the context of education.

Keywords : Intelligent Tutoring System, evaluation, methodology, service design, case study

Introduction

The recent developments in the capabilities of computer-based tutoring (Chughtai et al., 2016), and the increase in demand for personalized education (Mousavinasab et al., 2021) drew more and more attention to Intelligent Tutoring Sytems (ITSs). They have become widely applied across several educational domains (Mousavinasab et al., 2021), putting extra emphasis on the quality of their design and evaluation. In particular, evaluation studies carry great importance in determining the success of ITSs (Guo et al., 2021); therefore, ITS evaluation stands in the focus of this study.

Systems that use adaptive artificial intelligence to provide more personalized education to learners are called Intelligent Tutoring Systems (Mousavinasab et al., 2021). The goals of ITSs are mostly learner-centered, such as providing help in solving exercises when an instructor is not present (Chughtai et al., 2016) or supporting learners in achieving their educational goals (Chrysafiadi et al., 2022; Erümit et al., 2019). Even though learners are the most prominent user group of ITSs, the implementation of ITSs in educational contexts (Mousavinasab et al., 2021) introduces another user group, teachers (tutors, instructors) (Granić et al., 2002; Miller, 1988; Virvou & Tsiriga, 2000). Additionally, ITSs are usually supported by their development team or support staff, who are also significant stakeholders in providing the tutoring capabilities of ITSs. ITSs thus have learner-centered goals but the involvement of multiple other groups, i.e. teachers and the support team of the ITS also contribute to their successful ITS implementation. Consideration of all user groups is a principle of user-centered design and evaluation, just as placing emphasis on the implementation context (Still & Crane, 2017b).

Evaluation carries great importance in both the design phase of ITS development and the assessment of a complete system. Virvou and Tsiriga (2000) highlighted introducing multiple (formative) evaluation phases in the design stage focusing on system usability and

learner performance and a final (summative) evaluation in the end of the developmental cycle as a best practice in ITS evaluation. A similar focus on iteration also appears in user-centered design and evaluation (Norman, 2013; Still & Crane, 2017a). The aim of this study is to provide a framework for formative ITS evaluation.

The importance of evaluation in the ITS literature is underlined by the several evaluation frameworks developed throughout the years to provide holistic guidance to ITS evaluation (Andone & Sireteanu, 2008; Chughtai et al., 2016; Siemer & Angelides, 1998; Woolf, 2010). Three main approaches can be distinguished, namely evaluation along the lines of *system performance*, *learner performance* and *learner experience* (Mousavinasab et al., 2021).

There are three larger areas of shortcomings in the ITS evaluation literature regarding *learner experience*-based evaluation. (1) The neglect of the role of teachers and the ITS support team in *learner experience* (2) the lack of considering the implementation context in evaluation (data collection environment and timeline of evaluation) and (3) the lack of utilization of user-centered, iterative methodologies for data collection and analysis (examining user onboarding and log data analysis).

First, in the early years of ITS development, researchers advocated for user-centered evaluation involving both learners and teachers (Granić, 2008; Granić et al., 2002; Virvou & Tsiriga, 2000). However, recent developments in ITS evaluation left the evaluation aspect of *learner experience* neglected (Chughtai et al., 2016). Although usability studies are usually conducted with learners, none of the applied methods include the involvement of teachers or consider the support team (Erümit et al., 2019). Additionally, in some studies usability assessment is more *system performance*-focused (Mousavinasab et al., 2021).

Second, ITS usability studies are not considering the implementation context which is reflected in the circumstances and timeline of data collection. Usability evaluations are

usually performed in lab environments, at an arbitrary point in time (Erümit et al., 2019). User engagement data collected unmoderated and for longer periods of time is not yet utilized in ITS usability evaluation, or in ITS evaluation in general.

Third, empirical ITS usability studies are mostly using questionnaires or task completion-based usability assessment for summative evaluations with no iterations (Erümit et al., 2019). ITS evaluation is yet to utilize methodologies such as the study of user onboarding (Terres et al., 2019) or log data analysis (Greer & Mark, 2016). These new methods could improve understanding the behavior of users engaging with the ITS and would contribute to user-centeredness. Additionally, implementing them in formative evaluation iterations could provide actionable feedback to directly inform ITS development.

This study aims to address these shortcomings by introducing a new outlook on *learner experience*-centric evaluation, utilizing service design, a new user-centered approach. Service design is focused on not only the user's experiences with a service (or product), but it also considers the whole process of providing the service, also designing and evaluating the actions of service providers. Service design is applied mostly in the development and evaluation of commercial products, but the approach it represents suits the education domain particularly well (Wolfe, 2020), as it caters to teachers' and support teams' needs just as well as to students. Current ITS evaluation lacks this user-centeredness (Chughtai et al., 2016; Granić, 2008) and educational context-sensitivity which is why taking a service design approach in ITS evaluation could positively benefit the field.

All major shortcomings of the ITS evaluation literature identified could be addressed by the principles of service design. First, *user-centeredness* is a service-design principle in itself. Considering multiple stakeholders in the evaluation process can be addressed by the *co-creating* principle. These principles ensure that the needs of all major user groups and stakeholders are represented in the ITS evaluation. Second, the consideration of the

implementation context is addressed by service design offering a *sequencing* and a *holistic* approach (Stickdorn & Schneider, 2010). The *sequencing* principle means mapping the service user's activities onto a specific timeline. The timeline has three periods: *pre-service* (user onboarding), *service* (engaging with the ITS) and *post-service* (abandoning the ITS). This approach allows for a more detailed analysis in each service period while keeping a holistic contextual overview. The third shortcoming of the literature, i.e. the lack of iterations and utilizing user onboarding and log data analysis in ITS evaluations is addressed by applying the service design timeline (*sequencing*). The *pre-service period* can include the analysis of user onboarding; and studying the *service period* separately leaves space for using a multi-method approach that can involve log data analysis. Additionally, examining the different service periods in different studies leaves room for iterating the outcomes of previous studies.

Therefore, the service design approach carries the possibility of adding to the ITS evaluation literature but has not been utilized in the field before, which suggests the evaluation of its practical applicability. This leads to the main research question of this study.

Main Research Question: *How can the gaps in learner experience centered ITS evaluation be addressed by utilizing a service design approach?*

This study introduces a service design-based, iterative evaluation framework and a case study that tests the applicability of this framework to answer the main research question. The case study is built on evaluating Waterproof, an ITS with the aim to help students learn to prove mathematical statements. Waterproof is developed at Eindhoven University of Technology (TU/e) at the Department of Mathematics and Computer Science and it is currently implemented in context of the bachelor level university course, Analysis I. The well-defined context, stakeholders, user groups and direct access to log data collected by

Waterproof provide ideal conditions to perform a case study of evaluation with the service design approach.

The case study is organized around three sub research questions that divide Waterproof use into three periods according to the service design timeline.

Sub Research Question 1 (SRQ1): *How can the onboarding experience of Waterproof be improved?* – examining the *pre-service period*, i.e. when the user is in the process of getting to know Waterproof.

Sub Research Question 2 (SRQ2): *How do users engage with Waterproof*? – examining the *service period* when the users actively engage with Waterproof.

Sub Research Question 3 (SRQ3): *Which are the points where users abandon the process of proving mathematical statements in Waterproof?* – examining the circumstances of entering the *post-service period*, i.e. under what conditions do users stop using Waterproof.

For answering these research questions, four studies are conducted. Study 1 is a service blueprinting process where the blueprint of the user onboarding of Waterproof is created in a co-creation workshop. Study 2 includes two survey studies and log data collection. In Study 3, user tests and interviews are performed as a qualitative supplementation to the quantitative data collected in Study 2 and analyzed in Study 4. Finally, the aim of Study 4 is to create an activity sequence map based on the log data collected in Study 2.

The outcomes of these studies inform a list of recommendations for the Waterproof development team regarding potential improvements for Waterproof. The recommendations can target changes in the user onboarding process and the user interface and can suggest new features, all for improving Waterproof user experience and increasing retention.

In the following, the paper examines related studies in the ITS literature that help understand the different approaches taken and methods used for designing and conducting ITS evaluation studies and the gaps in the literature. Then, the new, service design-based evaluation framework is introduced together with the case study: Waterproof and the preparation for the evaluation is presented. Afterwards, the paper details the four studies of data collection, their results and conclusions followed by the aggregated results of the study, the general discussion of findings and the conclusion.

Related work

This overview of related work aims to showcase the approaches taken in conducting Intelligent Tutoring System (ITS) evaluation, with special attention to the variety of frameworks and focuses taken in the evaluation studies. Then, user-centered design and evaluation are examined along their connections to ITS evaluation. It is followed by a further elaboration of *learner experience* and *learner experience* measurement in the ITS evaluation literature. Subsequently, an analysis of ITS usability evaluation studies is included, with special attention to user-centered principles in usability studies. The overview is concluded by an outlook on the methods in ITS evaluation and the possibilities of service design addressing the shortcomings of the literature.

Evaluation of Intelligent Tutoring Systems

ITS evaluation is one of the main interests within the ITS research domain. The domain of ITS research being a multidisciplinary field (Guo et al., 2021), the goals, process and methods of evaluation can differ depending on the evaluation approach (education or software science). However, according to Mousavinasab et al. (2021), the educational field the ITS is applied in, the target group, purpose and the AI used to build the ITS also influence the evaluation process.

Due to the high significance of evaluation, it has been in the center of interest within ITS research throughout the history of the domain. Starting from the 1980s, several evaluation frameworks have been developed (Guo et al., 2021; Kabudi et al., 2021), most of them taking a slightly different approach to evaluation.

ITS evaluation approaches and frameworks

Shute and Regian (1993), focused on ITS efficacy, i.e. assessing whether the ITS reached its educational goals (how well they taught a certain part of material to students). They defined a seven-step process: (1) establishing the goal of the tutor; (2) identifying goals of evaluation; (3) developing evaluation design; (4) instantiating the evaluation design; (5) making logistical preparations for the evaluation study (6) pilot testing the system and (7) planning a primary data analysis for the study. This framework allows great liberty in the choice of methods but suggests that each ITS should be evaluated according to a specialized set of aspects applicable only to the ITS in question. With the increase in the importance of ITS evaluations, having to develop a specialized evaluation framework for each ITS is a very resource-intensive approach (Siemer & Angelides, 1998).

Siemer and Angelides (1998) designed a generally applicable evaluation framework for complete ITSs as a response to Shute and Regian's (1993) ITS-specific approach. They argued that systematic analysis of the relationship between the architecture and behavior of an ITS, and the educational impact of the ITS on students can provide a comprehensive evaluation approach that is not tailored to a specific system. They proposed a dual evaluation consisting of an *internal* and an *external* component. *Internal evaluation* was aimed at understanding how the internal architecture of the ITS yields the system's behavior and was concerned with the design and build of the system. *External evaluation* focused on the effect of the ITS on the student. Siemer and Angelides (1998) further divided *external evaluation* into two parts based on different ways the student can be affected by the ITS. *Learning*

achievement assessed the extent to which the ITS fosters learning, and *learning affect,* that was described as the combination of attitudes, motivation and emotions of learners caused by the ITS impacting the learning process and system adoption.

Similarly to Siemer and Angelides (1998), Woolf (2010) also argued that ITS evaluation (and all evaluation studies conducted in the field of Education Technology) should focus on both evaluating the system design, calling it *software evaluation*, that is similar to *internal evaluation* and the learning outcome, calling it *classroom evaluation*, that is similar to external evaluation. However, Woolf (2010) did not further differentiate between the different aspects of *classroom evaluation* according to the framework of Siemer and Angelides (1998).

A recent review paper by Mousavinasab et al. (2021) examined 53 ITS evaluation articles between 2007 and 2017 and identified three perspectives in ITS evaluation. First, *system performance* is measured along criteria such as accuracy, precision, sensitivity, adaptively, reliability, recognition rate, etc. This approach is similar to the *internal evaluation* of Siemer and Angelides (1998), and *software evaluation* defined by Woolf (2010).

Second, *learner performance*, usually measured in pretest-posttest experimental studies, builds upon the educational aspect of ITS evaluation and associates the quality of an ITS with the gained skills and knowledge it provides the student with. This approach is aligned with the *learning achievement* component of *external evaluation* of Siemer and Angelides (1998) and *classroom evaluation* of Woolf (2010). The third evaluation aspect was *learner experience*. Mousavinasab et al. (2021) also referred to it as user experience but did not clearly define the approach, only linked it to usability evaluations referring to Lawless et al. (2010).

Mousavinasab et al. (2021) drew the conclusions that about 43% of all reviewed ITS papers involved some form of evaluation along the lines of *learner experience* (together with

system performance and *learner performance* about 15%, combined with only *learner performance* about 23% and based on *learner experience* only about 6%). Thus, although *learner experience* is used as an aspect in several ITS evaluations, Mousavinasab et al., (2021) argued that it was not in the center of evaluation studies. Chughtai et al. (2016) supported this conclusion arguing that most of the development and evaluation studies in the domain of ITSs were focused on software science (*system performance* evaluations) and learning sciences (*learner performance* evaluations) and less attention was given to *learner experience*.

The findings of Lynch and Ghergulescu (2016) however contradicted the conclusion of *learner experience*-based measurements being neglected. Based on their review of 14 ITS evaluation frameworks published between 2010 and 2016, they found that user experience was, in focus of most ITS evaluations and they drew the attention to the lack of system performance evaluations in the field. It is important to mention that Lynch and Ghergulescu (2016) gave a precise definition of user experience by equalizing it completely with usability.

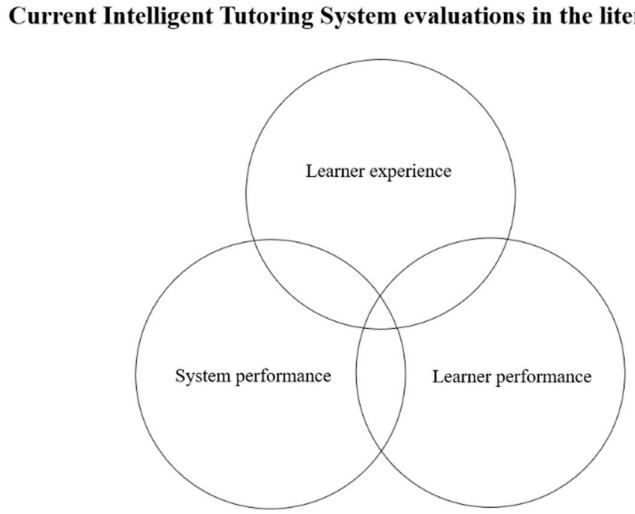
Chrysafiadi et al. (2022) summarized the current state of ITS evaluation by providing an overview. They claimed that even though performance is the main focus of ITS evaluation, usability is also a common evaluation perspective. Furthermore, they drew attention to the lack of a widely approved evaluation framework and best practice evaluation methodologies in the ITS literature.

The collection of evaluation perspectives and frameworks above shows the evolution of ITS evaluation. First, the generalizability of ITS evaluation was established by Siemer and Angelides (1998), and ITS evaluation was suggested to be conducted as a combination of evaluating the ITS from two aspects. A software science aspect (*system performance*) and an educational aspect assessing the effect of the ITS on students (*learner performance*) (Lynch & Ghergulescu, 2016; Siemer & Angelides, 1998; Woolf, 2010). Later, a third evaluation

perspective, *learner experience* gained prominence (Mousavinasab et al., 2021). However, there is confusion in the field about the definition and measurement of *learner experience*, which calls for further research.

Figure 1

Main evaluation approaches taken in the ITS literature (Mousavinasab et al., 2021)



Mousavinasab et al. (2021) equalized *learner experience* with user experience. Additionally, as learners are the primary user group of ITSs, *learner experience*-based ITS evaluation can be considered a user-centered approach.

User-centered design and evaluation

User-centered design (UCD) puts the user into the center of the design process. The approach places emphasis on directly involving users in all stages of the design process in order to understand their problems (design phase) and provide them with solutions that they can use easily and that fit their needs and motivations (evaluation) (Still & Crane, 2017a). Learners being users of ITSs, and ITSs supporting learner-centered goals, UCD is an approach that could help define *learner experience*-based ITS evaluation. UCD also places a great emphasis on examining the problem in context (Still & Crane, 2017b).

User-centered evaluation studies are either performed contextually, or users are placed into imaginary contextual situations, i.e. scenarios (Obendorf & Finck, 2008), while performing evaluation studies. User-centeredness thus advocates for creating specific solutions for specific groups of people that experience problems under specific conditions. This approach is similar to Shute and Regian's (1993) ITS-specific approach to evaluation and thus counters the universal ITS evaluation generalizability claim of Siemer and Angelides (1998).

The popularity and importance of UCD is underlined by the number of design and evaluation frameworks and methodologies developed, such as user experience design (Allanwood & Beare, 2019), participatory design (Kuhn & Muller, 1993) and service design (Stickdorn & Schneider, 2010). Service design is further detailed later in this section.

User-centeredness in ITS evaluation – learner experience

In the early years of the field, user-centeredness was considered in designing ITS evaluation studies and frameworks (Granić et al., 2002; Miller, 1988; Virvou & Tsiriga, 2000). Granić et al. (2002) even highlighted considering both teachers and learners as crucial contributors in designing and evaluating ITSs implemented in an educational context. They identified teachers as the ones controlling the learning process and responsible for the authoring within the ITS. (Granić et al., 2002) also emphasized the importance of communication between learners, teachers and knowledge (represented in the system) as a success determinant for ITSs. However, as mentioned in the section about evaluation frameworks, the focus of ITS evaluation was changed onto performance-related measures (Chrysafiadi et al., 2022; Chughtai et al., 2016), and parallel to this, the interest in user-centered evaluations declined.

Recently, with the appearance of *learner experience* as an approach to ITS evaluation (Mousavinasab et al., 2021), user-centeredness seems to gain more attention in the field of

ITS evaluation. However, there is significant confusion in the literature about the term *learner experience* which makes it difficult to identify and systematically assess *learner experience*-based ITS evaluation studies.

Learner experience is often used as an umbrella term, describing a general impression of the learner, their success and subjective evaluations of the ITS (e.g. Nwana, 1990). While Mousavinasab et al. (2021) equalized *learner experience* with user experience, Lynch and Ghergulescu (2016) used user experience as measurable construct and defined it as a synonym for usability. On the contrary, Hassan and Galal-Edeen (2017) identified a threefold relation between the terms user experience and usability based on their review of the use of these terms in the Human-Computer Interaction literature. They found that (1) usability is a part of user experience, (2) it is a quantifiable user experience measure, and (3) usability and user experience complete each other. Nevertheless, the approach of treating usability as equal to user experience has been used in several recently conducted empirical ITS evaluation studies (e.g. Chrysafiadi et al., 2022). This approach makes usability assessments the primary measure of *learner experience*-based ITS evaluation.

The next section assesses usability studies conducted in the ITS domain. It is important to note, that in this study, usability assessments are conducted, but *learner experience* is defined along Norman and Nielsen's (n.d.) user experience definition: "User experience encompasses all aspects of the end-user's interaction with the company, its services, and its products". Therefore, *learner experience*-based ITS evaluation in this study takes a more holistic approach than conducting only a usability study.

Usability

Usability is a quality attribute that determines the ease of use of a user interface. Usability testing usually involves assessing the system along quality components such as learnability, efficiency, memorability, errors and satisfaction (Nielsen, 1993). Usability issues

of ITSs are potential obstacles in the way of an individual user adapting an ITS. Insufficient system usability may therefore disrupt the *learner experience* (Chughtai et al., 2016; Lawless et al., 2010; Mulwa et al., 2011). Approaching usability not (only) as a system performance criterion but as a construct affecting user experience supports the growth in importance of user-centered evaluation of ITSs.

Usability evaluation has part of ITS evaluation since the early days of the field. Granić et al. (2002) found that scenario-based usability testing in ITSs provides a low-threshold opportunity to perform usability evaluations. They also argued that users' task performance is a good indicator of usability and thus usability should be assessed by user walkthroughs within the system interface, guided by predefined steps. They also highlighted the importance of contextual usability assessments in ITS evaluation that lead to performing scenario-based usability testing with real users of the system (Granić, 2008).

Granić (2008) defined scenario-based usability testing to have three steps. (1) a walkthrough usability test, assessing the system's learnability, efficiency and user errors through tasks of knowledge generation with the ITS; (2) a memo test assessing system memorability through asking users to recall effects of certain commands; and (3) a satisfaction questionnaire measuring the satisfaction usability component. This testing protocol is a combination of behavior and opinion-based measures, which contributes to understanding the studied phenomenon better according to the theory of triangulation (Thurmond, 2001).

Another approach to usability evaluations in the field of ITS evaluations is heuristic evaluation (Andone & Sireteanu, 2008). This approach is based on one or a few experts' assessment of a platform along a list of predefined aspects, such as visibility of system status, match between system and the real world, user control and freedom, consistency and standards, error prevention, recognition rather than recall, flexibility and efficiency of use,

aesthetic and minimalist design, help users recognize, diagnose and recover from errors, and finally, help and documentation. (Nielsen, 1993).

Empirical studies

The review of 13 ITS usability assessments conducted by Erümit et al. (2019) shows the preference for usability assessment methods in empirical studies. They compared user assessment-based (Granić, 2008) and expert-based (Andone & Sireteanu, 2008) methodologies. The results show that the user assessment-based approach was undoubtedly preferred. About 69% of the examined usability evaluations included some form of user assessment. The most often applied methods for user assessments were quantitative scales or questionnaires on usability, such as the System Usability Scale (SUS) (Bangor et al., 2009). Other user assessment studies applied task performance-based qualitative assessment, such as the scenario-based assessment with methods like the thinking-aloud protocol (Boren & Ramey, 2000). In about 31% of the studies examined by Erümit et al. (2019), expert-based, qualitative usability evaluation was used, but it is important to mention that almost all of these studies (23% of all studies) were also using a user-assessment based approach additional to the heuristic evaluation.

Additionally, the study of Erümit et al. (2019) showed that though it is not characteristic, a mixed methods research approach was taken in empirical ITS usability assessment studies combining usability questionnaires and task-performance based assessments. Mixed methods research refers to the collection and analysis of both qualitative and quantitative data in order to examine the research problem from multiple perspectives and thus build a better understanding of possible solutions. (Mertens, 2017). Although mixed methods studies are not characteristic to ITS usability assessments, they are a prominent method used in user-centered design and evaluation (Ivankova & Wingo, 2018).

Examining ITS usability evaluations from the aspect of user-centeredness, it can be concluded that they fail to consider multiple user-centered principles. First, usability assessments are centered around one user group of ITSs, learners (Erümit et al., 2019) and they do not consider teachers or the ITS support team. Second, studies are often conducted in lab environments, and therefore they do not consider the ITS implementation context. Furthermore, usability assessments are performed at an arbitrary point in time (Erümit et al., 2019), usually at the end of the development cycle, as a summative evaluation. Therefore, ITS usability evaluation results are not implemented in an iterative manner. Finally, ITS usability studies rarely utilize mixed methods research designs that would support rich data collection and better understanding of user behavior with the ITS. Consequently, although usability studies are the main measures of *learning experience*-based ITS assessment, they fail to adhere to user-centered principles. This creates a need for a new, user-centered approach for designing and conducting *learner experience*-based ITS evaluation studies.

ITS evaluation methods

In cases of *system performance* and *learner performance* focused ITS evaluations, (Mousavinasab et al., 2021) pretest-posttest experimental studies and questionnaires are researchers' main choices of ITS evaluation methods (Chrysafiadi et al., 2022; Greer & Mark, 2016; Mousavinasab et al., 2021). This clear preference for pretest-posttest experiments is interesting in the light of the recent development in data analysis techniques. Greer and Mark (2016) highlighted that ITS evaluation methods have not changed much since the early days of the field and pinpoint that the development in the design of ITSs is not mirrored by evaluation. Woolf (2010) argued that using automatically collected log data from ITSs can be used in ITS evaluations and Greer and Mark (2016) further suggested the use of "modern methodologies" such as learning analytics data collected by ITSs. They argued that this data can be used for recognizing patterns of use and learner activity. Log analyses can

also support the recognition of meaningful features which can contribute to more efficient ITS evaluation. Moreover, using automatically collected learning analytics data in ITS evaluations could also tackle the participation bias, pointed out by Greer and Mark (2016).

Log data analyses so far were used in ITS research in isolated studies for predicting student performance (Abu Naser, 2012; Cetintas et al., 2009; Haridas et al., 2020), but not for ITS evaluation. Abu Naser (2012) extracted log data from an ITS designed to provide learners with linear programming exercises of appropriate difficulty. They used low level logs about problem type and difficulty, student expertise, date and time, feedback options, number of attempts to solve the problem and user results. Subsequently, they trained a neural network with this data to improve upon the student model of the ITS. The evaluation of *system performance* showed that the model had a prediction accuracy of 92%.

Similarly, Cetintas et al. (2009) used logged performance features, problem features, time and mouse movement features to predict the likelihood of a learner providing a correct answer in an ITS. They built two different regression models based on performance and problem features; and performance, problem and time features. Then they used ridge regression correcting for data scarcity. For the comparative evaluation of *system performance*, they used the harmonic mean of precision and recall (F_1 metric). Evaluation results showed that using ridge regression provided better models.

The recent longitudinal study of Haridas et al. (2020) used an ITS aimed at providing general study material to learners regarding multiple school subjects. Logs were collected from the ITS; however, the type of data used is not shared. The logs were used to give formative and summative predictions of learner performance, identify students at the risk of failing and screen students with reading difficulties. They evaluated the prediction success via longitudinal data collection and concluded that prediction accuracy in all areas improved

due to the addition of logs to the prediction model. They also highlighted paying little attention to student errors as the main limitation of the study.

Another case of log data use in ITSs was conducted by Janning et al. (2016). Their aim was to use log data in order to predict learners' perceived task difficulty. They used logs related to hints available, data about learners skipping tasks, time between actions, results and the number of actions. They based their predictions on the assumption that the longer the time between actions, the more difficult learners perceive the task to be. They used F-measures to compare predictions based on different combinations of predictors but concluded that the volume of data they tested with was not sufficient to draw conclusions from.

The characteristics of log data analyses make this method a suitable candidate to include in user-centered, mixed-methods evaluation studies. Log data collection is low-threshold, yet analysis outcomes can provide rich data about users' behavior while interacting with an ITS.

Gaps in the literature

In summary of the overview of related work, it stands out that there is thorough theoretical grounding for the different ITS evaluation approaches, i.e. *system performance, learner performance* and *learner experience*. There are several proposed frameworks in the literature; however, the practical implementation of these frameworks and new methodologies in empirical ITS evaluations are not always implemented (Chrysafiadi et al., 2022). User-centeredness, although prominent in the early years of the field, lost attention due to the increase in the significance of *system* and *learner performance* related evaluation studies. Even though *learner experience* represents user-centeredness in newer ITS evaluation studies, there is unclarity in the literature regarding the definition of the term. Usability studies are the primary measures of *learner experience*-based ITS assessment, but

usability studies fail to consider user-centered principles. This creates the following gaps in the ITS evaluation literature regarding *learner experience*-based assessments.

First, contrary to the early ITS literature that emphasized the involvement of teachers (Granić, 2008; Virvou & Tsiriga, 2000) in evaluation studies, ITS evaluations do not consider the role of teachers, or the ITS support team in providing the *learner experience*, although their role is prominent in an educational environment.

Second, evaluation studies fail to consider the context of use for ITSs. In most cases, ITSs are implemented in educational contexts (Mousavinasab et al., 2021); however, the evaluations are often performed in lab environments, with a selected subset of participants, which can introduce participation bias. It also assumes that users behave the same, i.e. use the system in a lab environment, accompanied by an experimenter comparably to when they are working with the ITS at their preferred location, unobserved. This can introduce experimenter bias.

Another contextual aspect not considered is changes in use over a period of time. ITS usability evaluations are confined to an arbitrary point in time and fail to consider the use of the ITSs as a longitudinal process in time, which can lead to false assessments of usability.

Third, although there are methodologies available for studying ITSs that could increase the quality of evaluation data, approaches like log data analysis are rarely implemented in empirical ITS evaluation studies.

Service design – a new approach to *learner experience*-based ITS evaluation

Introducing a new approach to ITS evaluation based on the user-centered principles of service design could address the shortcomings of the ITS evaluation literature. It would allow for conducting user-centered, contextual evaluations of ITSs and also make room for the implementation of novel, user-centered methodologies in ITS evaluations.

Service design is a relatively new approach within user-centered design, usually applied in case of commercial product/service development (Catalanotto, 2018). Service design is an interdisciplinary design approach that combines different methods and tools from various disciplines such as psychology, marketing, design, etc (Stickdorn & Schneider, 2010). This is different from user experience design, which focuses on an end-to-end experience created for a user group; service design is centered around how this experience is created and thus involves multiple stakeholders (Stickdorn & Schneider, 2010). This is especially beneficial in an educational setting where multiple stakeholders are present (students, educators, student assistants, administrators, etc.). Wolfe (2020) argues that the service design approach provides "genuine co-creation with users to address their experience holistically, identifying the needs of all stakeholders" (Wolfe, 2020, p. 3). Moreover, ITSs can also be considered as services provided by their development team to stakeholders such as instructors, assistants, students as users in the context of a course or part of a course.

The service design methodology is organized along five principles: it is *user-centered, co-creative, sequencing, evidencing and holistic* (Stickdorn & Schneider, 2010). Only *evidencing* is strictly applicable for commercial products out of the five principles, as it is about creating a positive tangible memory (a gift) about the service experience that stays with the user. This is more difficult to apply to ITSs applied in an educational context, which is why it is not included in the current version of the evaluation framework. In the following, this section details how the other four service design principles (*user-centered, co-creative, sequencing and holistic*) support the applicability of the service design approach in *learner experience*-based ITS evaluations:

User-centered

Although service design represents a holistic approach, it still places the users in the center of the evaluation and prioritizes examining services along the users' actions. This approach also provides a stable lead for performing ITS evaluation studies.

Co-creative

Service design encourages involving all stakeholder and user groups in the design process as each group has different needs and expectations. As opposed to the "traditional" user-centered methodologies, such as surveys, user interviews, usability tests or focus groups (Allanwood & Beare, 2019), service design takes this one step further and expects all stakeholders to (1) be in one room for workshops and (2) actively participate in these sessions (Stickdorn & Schneider, 2010).

This approach is similar to the participatory design methodology, another branch of user-centered design that advocates for the democratization of the design process and promotes users participating actively in all stages (Kuhn & Muller, 1993). Involving users and stakeholders to this extent has not been done in the field of ITS evaluation before. This approach addresses biases that occur from performing evaluations with a selected group of users while neglecting another (students as opposed to teachers and the ITS support group).

Sequencing

Service design places emphasis on visualizing services as a sequence of interrelated actions. Therefore, understanding and mapping the timeline of these actions has great importance in service design methodologies. Service design uses an extended timeline to map the service process onto. This timeline is divided into three periods according to the user's involvement with the service. (1) The *pre-service period* includes the means through which the user gets to know the service and gets in touch with it (user onboarding). (2) The *service period* includes all the interactions while the user engages with the service and the (3) *post-*

service period that takes place after the interaction with the service and includes all follow-up interactions between the user and the service (Stickdorn & Schneider, 2010). All ITS evaluation studies are conducted in the *service period.* Examining the *pre-service* and *post-service* periods separately in terms of ITS evaluation is a novelty that would help place evaluation studies more into context by restricting the examined timeline and collect data answering the questions specific to the examined period. This would allow for more understanding about the longitudinal process of system-user interaction and iterative implementation of findings between studies examining different periods. Having more information about sequences of user actions allows closer user-centeredness via understanding the user pain points, their origin and potential remedies better (Norman, 2013).

One way of examining the *pre-service period* of an ITS is via focusing on user onboarding. User onboarding in case of commercial software products means the process of introducing the potential user to the capabilities of the software product in order to increase the chance of them becoming a user. Onboarding can also incorporate online or offline training, goal-setting, and the organization's customer success process, depending on the type of software product (Renz et al., 2014). Onboarding builds on the *first impression bias*, i.e. the quick and incomplete observations of the user based on the first piece of information perceived that lead to assumptions and judgements about the product (Lindgaard et al., 2011). First impressions and thus also onboarding is crucial in the success of commercial products, i.e. retention (Cascaes Cardoso, 2017).

The field of user onboarding research is under-studied (Terres et al., 2019), which explains the few onboarding-related studies in the ITS domain. Although less present in the literature, ITS user retention is an important factor. A user-centered onboarding process can contribute to the positive user evaluations of the ITSs. An example of an ITS onboarding study was conducted by Pian et al. (2020) and focused on helping new learners overcome the

lack of information about a complex ITS. The study found that designing a gamified onboarding process helps start to interact with the ITS. However, onboarding has not yet been studied in terms of ITS evaluations, even though onboarding evaluation could help improve the perceived usability of the system (Lindgaard et al., 2011).

Consequently, placing emphasis on *sequencing* in ITS evaluation would significantly improve the understanding of user experiences and would allow for more informed decision-making if applied in iterative evaluation studies. Examining ITS use on the service design timeline, could also open the opportunity to apply a variety of methodologies and create room for mixed methods iterative evaluation study designs.

Holistic

The holistic approach service design represents already appears in the inclusion of all stakeholders and the broad timeline, but it is also important that service design considers the entire service environment. Applying the holistic service design principle in ITS evaluations would help place the evaluation process into context, unhinging it from the previously characteristic evaluations performed in laboratory environments.

Application of the service design approach could be a beneficial addition to the approaches of ITS evaluation, as it would support user-centered *learner experience*-based ITS evaluation. This would include considering all stakeholders and the ITS implementation context. Additionally, it would make room for the utilization of user-centered methodologies in mixed methods evaluation studies. However, the service design approach was not yet used in the ITS evaluation literature, which leads to the main research question (MRQ) of the study.

Main Research Question: *How can the gaps in learner experience-centered ITS evaluation be addressed by utilizing a service design approach?*

This study

The aim of this study thus is to provide a proof of concept for a new, user-centered approach to *learner experience*-based ITS evaluation. The new approach addresses the gaps in the literature by utilizing the principles and tools of service design.

The service design based ITS evaluation framework

The service design-based evaluation framework (Figure 2) builds on the combination of the user-centered, iterative framework of design thinking (Allanwood & Beare, 2019) and the software development life cycle (SDLC) model (Radack, 2009). However, the framework was designed explicitly considering the needs of ITS evaluation.

User-centeredness suggests an ITS-specific evaluation approach, which calls for a preparatory phase of describing the ITS in question. This phase is followed by data collection and analysis in the *pre-service period*, the *service period* and the *post-service period.* The service design based ITS evaluation process ends with a summary of recommendations based on the evaluation and implementation of these recommendations. Just as the whole evaluation process, the data collection in different periods is also iterative according to the principles of user-centeredness providing a formative evaluation framework. The process is depicted by Figure 2.

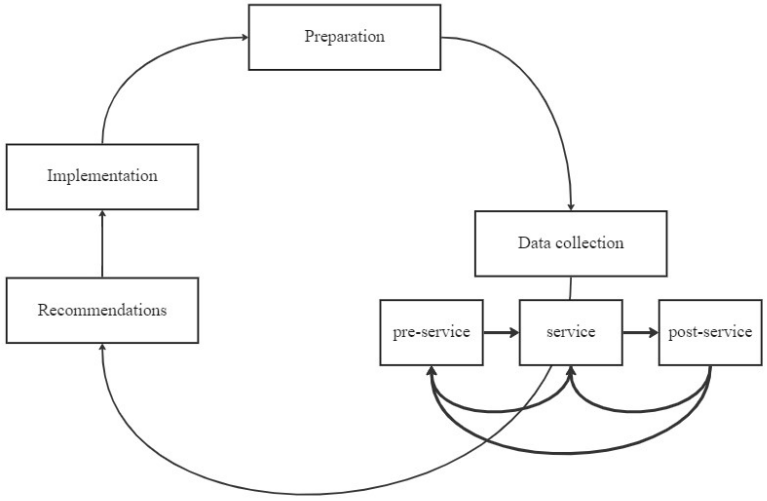
The Preparation phase provides the professional conducting the ITS evaluation with information about the ITS before starting data collection and supports creating an evaluation plan. The methods suitable to inform the recommendations and the evaluation timeline can also be determined in the Preparation phase.

Preparation is followed by iterative Data collection. Separating the *pre-service period* (learning to use the ITS) within data collection makes studying the user onboarding of ITS systems possible, while studying the *service period* yields the opportunity to perform the mixed methods research *learner experience* assessment including e.g. usability assessments.

Studying the *post-service period* systematically and separately allows for an understanding of the reasons why people abandon the ITS by examining the contextual circumstances. It also helps with design decisions to increase retention. Iteration within the Data collection phase provides opportunity to test the implemented changes based on evaluation studies conducted in earlier periods. The iteration-within-iteration structure provides instant feedback which allows multiple iterations of recommendations.

Figure 2

The service-design framework for learner experience-based ITS evaluation



Finally, after analyzing the data acquired in the Data collection process, a list of actionable recommendations is given about potential improvements of the ITS which are then considered and implemented by the ITS development team. The implemented changes can be evaluated in a further iteration.

The case study – Evaluating Waterproof using the service design framework

The service design framework is assessed in a case study including the evaluation of Waterproof. Waterproof is an ITS supporting students learning to prove mathematical statements (Waterproof Development Team, 2022), developed at the Department of Mathematics and Computer Science of Eindhoven University of Technology (TU/e).

Waterproof checks the logical soundness of each proof step and provides guidance and feedback to the user. It is built on the *Coq proof assistant*, which is a special computer program that checks the structures and correctness of proofs written in a syntax specific to Coq (The Coq Development Team, 2022). There are multiple factors that distinguish Waterproof from other *Coq proof assistant*s. First, Waterproof has a custom-developed Coq library, *coq-waterproof 2*. It extends the default Coq tactics allowing users to use natural language when formulating their proof, making Waterproof proofs more similar to handwritten ones. Second, the Waterproof editor has a unique design with a self-developed user interface (Wemmenhove et al., 2022). Waterproof has been implemented as a part of the bachelor university course for Applied Mathematics students. Analysis 1 at TU/e in the academic years 2020/21, 2021/22 and 2022/23. This educational setting provides a designated context and well-defined stakeholders for contextual ITS evaluation utilizing the service design framework. Teachers of Analysis I communicate with students using the Learning Management System Canvas (Instructure, 2023). Before the start of this Master Thesis Project (in September 2022), Waterproof was not subject to formal ITS evaluation.

For conducting an ITS evaluation with the service-design approach on Waterproof was to perform the Preparation, Data collection (Studies 1-4) and Recommendations (Appendix F) phases of evaluation. The following sections detail the outcomes of these phases.

Preparation

The Preparation phase of this project started with a series of introductory conversations with two members of the Waterproof development team. The aim of these conversations was to start thinking about Waterproof as a service provided by the team to the students and instructors, clarify the goals of Waterproof, explore who are the key stakeholders and define which aspects of the evaluation study could be beneficial for the

development team. Sticky notes and a flipchart were used for brainstorming in these initial sessions and. The outcomes can be found in Appendix A. In collaboration with the Waterproof development team, the following inquiries for the evaluation study have been identified. These inquiries are also sub research questions of this study.

Sub Research Question 1 (SRQ1): *How can the onboarding experience of Waterproof be improved? –* examining the *pre-service period*, i.e. the user onboarding of Waterproof. For answering SRQ 1, a service blueprinting session (Study 1) was organized, where the blueprint of the onboarding period of Waterproof was created in a co-creation workshop. Key stakeholders of Waterproof, i.e. students, instructors, members of the development team have attended the workshop.

Sub Research Question 2 (SRQ2): *How do users engage with Waterproof? –* examining the *service period* when the users actively engaged with Waterproof, i.e. performing a mixed methods *learner experience* assessment on Waterproof. Three studies contributed to answering SRQ2. Study 2 involved two survey studies with 29 voluntary students using Waterproof during the course Analysis I in 2022/23. Both surveys included questions related to usability and supplementary questions about Waterproof use. Additionally, the surveys provided a safe platform for students to share log data collected by Waterproof. Study 3 served as a qualitative supplementation to the quantitative data collected in Study 2. It involved conducting five user tests and some contextual questions about using Waterproof with students from the pool of the survey participants. Finally, Study 4 was the process analysis of user behavior in Waterproof conducted on the log data collected in Study 2.

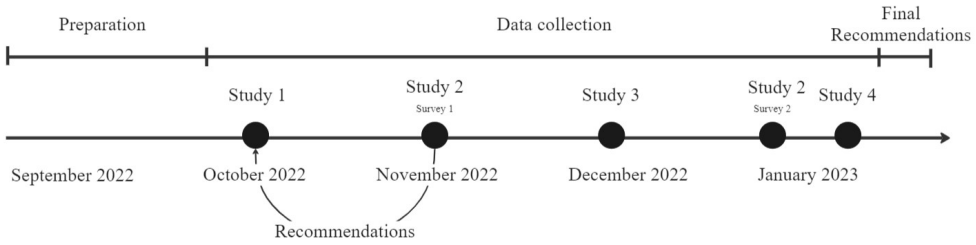
Sub Research Question 3 (SRQ3): *Which are the points where users abandon the process of proving mathematical statements in Waterproof? –* examining the circumstances of entering the *post-service period*, i.e. how and under which conditions users abandon

Waterproof. Study 3 included contextual questions about “giving up” on a problem in Waterproof and Study 4 examined the process and attributes of Waterproof use preceding closing the program after an unsuccessful work session.

The Preparation phase also included defining a timeline for all the studies conducted to provide recommendations and answer the sub research questions of this study. The timeline can be found in Figure 3.

Figure 3

The timeline of Waterproof evaluation case study



As a part of Preparation, the general course evaluation of Analysis I was examined. The following questions about Waterproof were included in the evaluation questions in the academic years 2020/21 and 2021/22: (1) Have you tried Waterproof (Y/N), (2) If yes: How did you like working with Waterproof? (1-5 Likert-scale where 1: I did not like it at all and 5: I liked it very much) and (3) If yes (to (1)): What suggestions of improvement for Waterproof would you have? (open-ended question). An inductive thematic analysis (Guest et al., 2011) was conducted using the answers to the open-ended question, which provided insights about previous student experiences of Waterproof. Thematic analysis included snippets from the answers containing information about the topics in interest were copied onto virtual sticky notes and clustered in Miro(Miro, 2023), an online whiteboard tool. Themes, subthemes and links between themes were identified and user quotes were collected to support themes. The following persistent issues emerged. (1) Students found it time-consuming to get into

Waterproof and did not find it equally helpful to work with throughout the course. (2) They found Waterproof a nice aid to get into “mathematical thinking” but found it more of a burden towards the end of the course, as the exam was on paper. (3) They also voiced that study materials additional to the Waterproof Tutorial and more explanations about the tool would be helpful, especially regarding error messages and syntax. These insights drew attention to the pain points of previous users of Waterproof and thus helped build an understanding about earlier students’ needs before diving into Data collection. The detailed outcomes of the thematic analysis can be found in Appendix A.

Finally, the user interface of Waterproof was examined as part of Preparation. Figures 4 and 5 showcase the main elements of the interface.

Figure 4

Waterproof user interface elements 1.

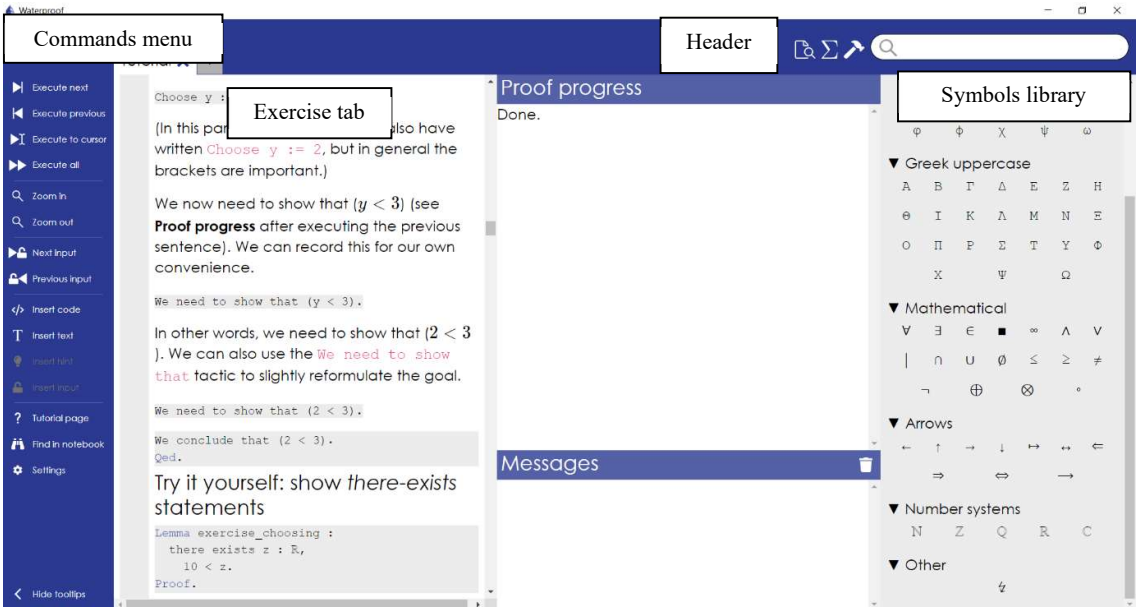
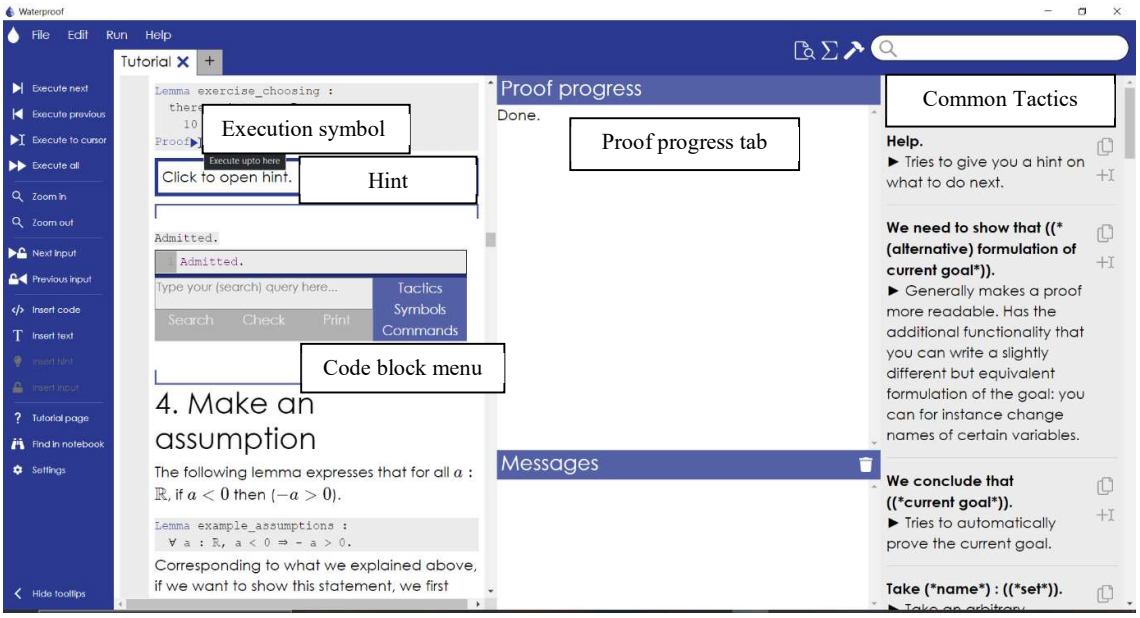


Figure 5

Waterproof user interface elements 2.



Data collection - Study 1 – Service blueprinting of Waterproof user onboarding

Study aim(s)

Study 1 was aimed at examining the user onboarding process of Waterproof by drawing up a service blueprint of onboarding, co-created by all stakeholders of Waterproof. Subsequently, Study 1 identified the pain points of all involved users and provided a first round of recommendations on how to solve the identified pain points and improve onboarding. Therefore, Study 1 had the purpose of answering SRQ1 (How can the onboarding experience of Waterproof be improved?).

The onboarding period has been defined together with the Waterproof development team as the interval between the students reading the first announcement of the course (including an introduction to Waterproof) until completing the Waterproof Tutorial, a series of exercises provided to familiarize new users with the syntax, features and tips and tricks of Waterproof.

Method

Study 1 consisted of a service blueprinting co-creation session, which was followed by synthetizing and organizing the collected data into a service blueprint. Service blueprints are the most often utilized methods in the service design toolkit (Gibbons, 2017). Service blueprints are visual maps centered around the actions users perform while interacting with the service, but also consider and map out the actions of all actors and stakeholders involved. Service blueprinting is a suitable method for evaluating ITS user onboarding as it gives a comprehensive understanding of how all user groups cooperate to prepare new users of the system for working with Waterproof. Service blueprints also visualize the relationships (links, dependencies, causalities, etc.) between *service components* such as people, physical or digital props (e.g. touchpoints) and processes. Mapping out the interactions allows for identifying pain points that can be obstacles of using an ITS and helps identify opportunities for optimization. Detailed explanation about the service blueprint framework and its elements can be found in Appendix B, in the service blueprinting session script.

Participants

The six participants of Study 1 were carefully invited so that all stakeholder groups, i.e. students, teachers and the members of the Waterproof development team were represented. Two students participated who formerly used Waterproof during the Analysis I course. One of these participants is also part of the Waterproof development team. Two other members of the development team were present, who are also current instructors of Analysis I. Two additional instructors attended. They were responsible for Waterproof instruction groups in the past, but they are not familiar with Waterproof on a development level themselves. The session was led by one researcher. Participation was voluntary.

Materials

In the co-creation session participants worked on a sheet of brown packaging paper and used five different colors of sticky notes and two different colors of voting dots for pain points (red – prioritized, yellow – not prioritized). The basic structure of the service blueprint was pre-drawn for the session by the researcher. The session setup can be found in Figure 6. For the digitalization, organization and analysis of the collected data, Miro and a service blueprint template (Digital Design Agency, 2022) were used.

Figure 6

Service blueprint co-creation session setup

Procedure.

The service blueprinting co-creation session took place on the TU/e campus, in the Atlas building. It is important to mention that the session took place before Analysis I started in the 2022/23 academic year. All participants attended physically, one participant had to leave 45 minutes into the session and another participant arrived 50 minutes late. The session took about 105 minutes and consisted of three parts. First, in Part A, the participants learnt about the service blueprinting framework, and the aim of the session, i.e. drawing up a preliminary service blueprint for the user onboarding of Waterproof. Following, in Part B,

participants collaborated in gathering all the steps new users perform during onboarding. These steps defined the phases of onboarding to be examined separately. Then, participants were asked to form two groups with mixed members from each stakeholder group and work out the frontstage actions, backstage actions and support processes for the identified onboarding phases. These activities were followed by Part C, where participants worked together in one group again, reviewed each other's work, defined the onboarding timeline, identified pain points of all stakeholder groups and prioritized these pain points with dot voting according to the urgency of solving them. Concluding the session, there was space for participants to give feedback and ask questions. The detailed script for the co-creation session can be found in Appendix B.

For data analysis, the outcomes of the blueprinting session have been digitalized in Miro and the researcher added the dependencies between actions based on the discussion in Part C of the session. The digital blueprint was sent to all participants for feedback. Two participants gave feedback which was implemented. Finally, the researcher collected the pain points and gave suggestions for solutions.

Results

First, Part A of the blueprinting session resulted in six main user actions that defined the timeline and phases of Waterproof onboarding: (1) Reading (Canvas) announcement (about Waterproof), (2) Join Canvas group with Waterproof support, (3) Going to GitHub page (of Waterproof), (4) Trying to install Waterproof. Here, depending on the success of this step, students either (5) Go to the (first) instruction session to install Waterproof or arrive at the instruction session with Waterproof installed and (6) do the Tutorial during the instruction session. It is important to add that students were required to finish (and submit) the Tutorial even if they could not finish it during the first instruction session.

The outcomes of Part B and C, i.e. the service blueprint supplemented with the timeline and the dependencies of activities can be found in Appendix B. Three insights were derived from the blueprint after aggregating the data collected at the session. First, regarding the timeline, onboarding consists of a few days only. It is important to note though that all the preparation activities performed by teachers and the development team take place between the end of the course in a year and the beginning of the onboarding in the next year. These activities can be performed over the course of a whole year. Second, most of this longer-term preparation is the responsibility of the Waterproof development team. Third, the onboarding of Waterproof is linked to three digital platforms (besides the physical (first) instruction session): the Canvas Learning Management System, GitHub (the platform from which students can download Waterproof) and a dedicated channel for Waterproof on Microsoft Teams (where students can reach out for help online).

Table 1 contains the pain points that were identified during the co-creation session that got at least one red voting dot, indicating priority. One pain point of the development team, namely non-availability of hardware to test Waterproof solutions on both Windows and Mac operating system was not considered, as it falls out of the scope of this study.

Table 1 shows that all but one prioritized pain point was linked to student actions and thus affects students primarily. The pain point of instructors was related to late involvement into working with Waterproof. Instructors did not feel competent enough to provide the needed help to students struggling with Waterproof. Three of the student pain points were related to the Tutorial. Students of previous years noted that not everyone completed the Tutorial as it was difficult to complete on time. Instructors also noticed that students did not execute code in the Tutorial and in later exercise sheets. This is a crucial step in Waterproof use, since execution tests code correctness. Two further student pain points were related to the GitHub platform. Several students interacted with GitHub for the first time for

downloading Waterproof. Some of them did not find their way around the platform and found it overwhelming at first. They also did not find the information on GitHub sufficient to install Waterproof successfully.

Table 1

Descriptions of prioritized pain points and solutions for them based on the service blueprint of Waterproof user onboarding

	Pain point	Student/Instructor	Suggested Solution
1	Students do not complete the Tutorial	Students	Make a video explanation for the installation and tutorial
2	The Tutorial is too difficult to complete on time	Students	
3	Students do not execute code (in the Tutorial)	Students	
4	Students fail to install Waterproof based information on GitHub only	Students	
5	GitHub looks scary for students	Students	Create a Waterproof information page on Canvas Manage installation using this page (instead of GitHub)
6	Instructors have no time to learn to work with Waterproof	Instructors	Let (new) instructors test the newest version of Waterproof

Conclusion

First, having an iterative structure of preparation for onboarding allows for the assessment of implemented improvements of the onboarding iteratively. This supports the application of a formative evaluation framework. Furthermore, the development team being

responsible for most of the preparation allows for great control over the planning of preparation.

Furthermore, results of Study 1 show that the most pain points of the user onboarding process of Waterproof affect students, that is the largest user group of ITSs. As Table 1 shows, the advice of creating video explanations for installing Waterproof and starting the Tutorial solves multiple pain points combined. Moreover, this suggestion does not only solve these pain points but also the pain points identified based on the course assessment of Analysis I in the *Preparation* phase. Namely, the videos provide more explanation and reduce the time to start working with Waterproof. Additionally, leaving GitHub out of the list of platforms by migrating the Waterproof installer files to Canvas can simplify the installation process.

In conclusion, Study 1 answered SRQ (How can the onboarding experience of Waterproof be improved?) of the study by providing a list of suggested solutions for pain points identified in the service blueprint co-creation session of Waterproof user onboarding. Suggestions regarding student pain points were implemented and students' reactions on the onboarding were measured in Study 2.

Data collection - Study 2 – Survey studies

Study aim(s)

Study 2 answered research questions directly and indirectly via informing Studies 1, 3 and 4. Study 2 contributed to answering SRQ 2 (How do users engage with Waterproof?) specifically regarding changes in perceived usability. It also provided supplementary data about aspects of *learner experience* such as user motivation, satisfaction and cognitive load. Study 2 also indirectly helped answer SRQ1 (How can the onboarding experience of Waterproof be improved?) by providing feedback on the implemented changes after Study 1.

Method

Surveys allowed for gathering quantitative data about Waterproof use early in using the tool (Survey 1) and later, when students built up approximately a month of experience in using Waterproof (Survey 2). Survey 1 was administered in the beginning of the course Analysis I, with a deadline dating to two weeks after the first instruction session which meant the end of the onboarding process. Students had a week's time to fill in their responses. Survey 2 was administered eight weeks after the onboarding. Respondents had, again, a week to fill in the survey. This double data collection allowed quantitative comparison of usability scores recorded in these two timeslots.

Participants

Students of Analysis I in the academic year 2022/23 that were part of the three instruction groups using Waterproof were invited to participate in Study 2. Participation was completely voluntary and had no relation to the grading of the course. 29 respondents completed both surveys, eight students completed only Survey 1 and two students completed only Survey 2. As participation included filling in both surveys and only Survey 1 included the informed consent form, 29 participants' data was used in the analyses.

Materials

For assessing the perceived usability in both surveys, the System Usability Scale (SUS) was used (Bangor et al., 2009). It consists of ten questions (e.g. I found Waterproof very cumbersome to use.) assessing usability as an inert concept, meaning that subscales or individual questions of SUS separately do not carry valuable information. SUS scores are measured on a five-point Likert scale (1 – Strongly disagree; 5 – Strongly agree) and scores can range from 0 to 100. According to the acceptance scale for SUS worked out by Bangor et al. (2009), a score below 50 means that the system usability is not acceptable, a score between 50 and 70.5 means marginally acceptable (between 50 and 62.5 low-marginal and

between 62.5 and 70.5 high-marginal) and above 70.5 the system usability is deemed acceptable.

User motivation was measured by eight questions assessed on a five-point semantic differential scale. The questions were adapted from the motivation questionnaire developed by Guay et al. (2000) and the scale had an internal consistency of α=0.81 in this study. Satisfaction was measured with a scale of four questions assessed by a five-point Likert-scale, similar to SUS scores. The questions were adapted from the extended Technology Acceptance Model defined by Dasgupta et al. (2002). Regarding satisfaction scores, the internal consistency of the four subscales was α=0.6. Both motivation and satisfaction can be reported with one score. Finally, cognitive load was measured by three questions rated 1-10 (1 - very low load, 10 - very high load) adopted from the NASA Task Load Index (Hart & Staveland, 1988). The cognitive load scores were analyzed along the NASA TLX analysis guidelines. Outcomes of cognitive load can be: low (0-9), medium (10-29), somewhat high (30-49), high (50-79) and very high (80-100). The surveys were administered via LimeSurvey (LimeSurvey GmbH, 2022), which is a GDPR compliant online survey tool.

Procedure.

Survey 1 had the following structure: (1) SUS scale, (2) open-ended questions about the installation process, the Tutorial, the Waterproof Canvas page and generally about the initial experiences with Waterproof, (3) log data submission and (4) email address collection for matching surveys and contacting participants. The structure of Survey 2 was similar to Survey 1: (1) SUS scale + open-ended question about changes in the general impression about Waterproof, (2) questions about satisfaction, cognitive load and motivation + open-ended questions about the changes in constructs, (3) log data submission and (4) email address. Descriptive statistical analysis of the quantitative outcomes of the survey were performed in Microsoft Excel and qualitative data analysis, performed with counting and

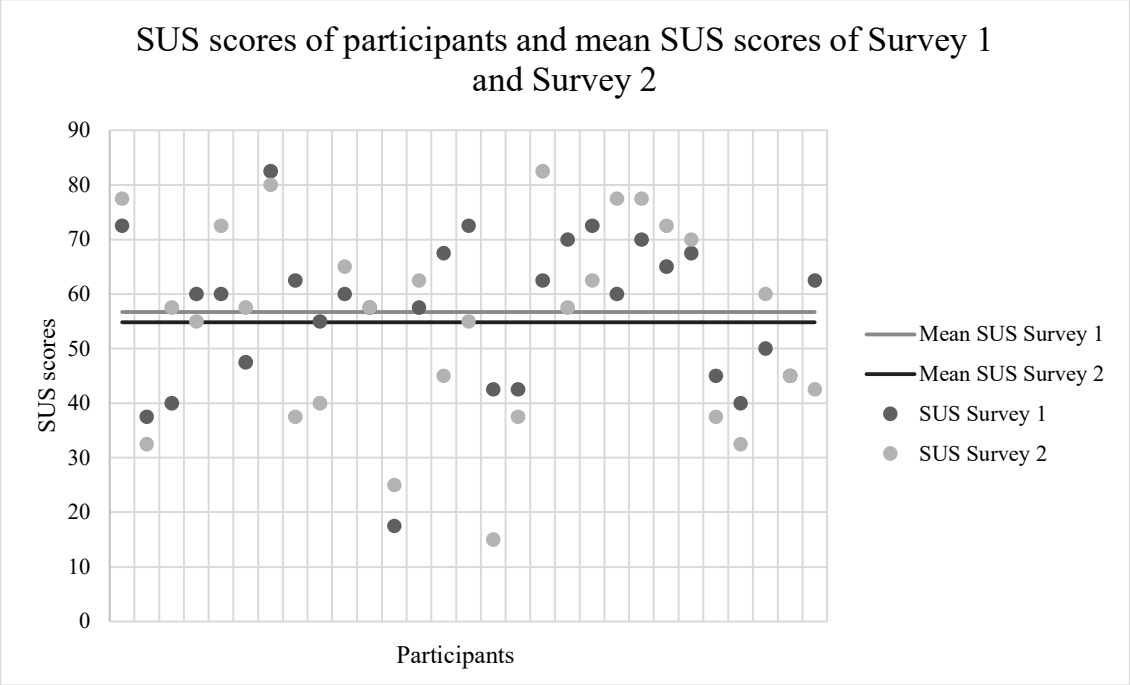
thematic analysis, took place in Miro. Thematic analysis was performed with an inductive approach meaning that all the theories presented in the results are originated from the patterns in the data (Guest et al., 2011). In case of counting, only explicit mentions of constructs have been counted.

Results

The SUS scores in both surveys show high variation (Figure 7). The scores ranged from 17.5 to 83 in Survey 1. The mean score was $M_1 = 56.72$ with a standard deviation of $SD_1 = 13.76$. The mean SUS score decreased by 1.98 points between Survey 1 and Survey 2. The SUS scores ranged from 15 to 82.5 in Survey 2 with a mean $M_2 = 54.74$ and a standard deviation of $SD_2 = 17.6$. The system usability measured in both surveys falls thus in the category of low-marginal.

Figure 7

The individual and mean SUS scores measured in Survey 1 and Survey 2



Note. Overlapping datapoints mean that the participant's SUS score was unchanged. Each column represents one participant.

Detailed outcomes of the thematic analysis of feedback on the onboarding can be found in Appendix C. In summary, respondents described the installation process with words like "easy" (21 mentions), "smooth" (6 mentions) and "quick" or "fast" (6 mentions altogether). Participants also described the installation process as "not more difficult than any other application".

The Tutorial was subject to more diverse judgement by respondents. Students described it as "useful" and "helpful" (16 mentions altogether), and "clear" (6 mentions) but also as "long" (6 mentions). Other students found the Tutorial "necessary" to understand Waterproof (5 mentions). Students mentioned unclarities, e.g. "[after completing the tutorial] I still do not understand some functions." and "[the Tutorial] was easy to follow but difficult to implement". The comments also showed conflicting information about the expectations regarding the Tutorial. While some students lacked detail saying: "…the tutorial only scratched the surface." and "a more comprehensive tutorial with more practice of combining several concepts would be useful"; others found it unnecessary lengthy: "…for some things I can imagine I would have figured them out from looking at the tactics." The inconsistency in expectations about the Tutorial was also reflected in the general suggestions for improvement regarding Waterproof. On the one hand, some students mentioned the need for an "easier and longer" Tutorial, more practice exercises with the general mindset of "initially, it's better to keep it simple". On the other hand, others suggested adding "a more difficult exercise in the tutorial" "with hints". The need for more explanation about the syntax and errors were also suggested by statements such as "…it would be useful if I knew what I was doing wrong." Altogether four students mentioned more elaborate error explanations and four respondents suggested to provide more aid for troubleshooting. It is important to mention that these

comments were made by students two weeks after the completion of the user onboarding of Waterproof.

The results also revealed that the Canvas page created for Waterproof was complete by students saying nothing was missing (27 mentions). However, three students mentioned that they did not remember the page at all.

The quantitative *learner experience* scales administered in Survey 2 yielded the following insights. The mean score of motivation was M = 3.2 with a standard deviation of SD = 1.01. The satisfaction score was M = 3.7 with a standard deviation of SD = 0.74. Participants' cognitive load was measured with three subscales and yielded the result of participants considering the mental effort (M = 27.09) and difficulty of using Waterproof (M = 23.1) "medium" and the frustration associated with Waterproof use (M = 29.48) as "somewhat high".

The outcomes of qualitative analyses performed on answers to open-ended questions in Survey 2 resulted in the following insights. 11 students indicated that their general opinion about Waterproof changed for the better since the beginning of Analysis I. They elaborated on this claim by stating that Waterproof became easier to use (5 mentions) and that they grew to understand Waterproof better (3 mentions). Five students stated that their opinion about Waterproof changed from positive to negative and two of them mentioned that they switched to completing proofs on paper. They named the increasing difficulty of proving exercises as a reason: "Once my understanding of proofs became blurry, it also became harder to use Waterproof…". Six respondents mentioned that their general opinion about Waterproof did not change throughout the period of using it.

Regarding satisfaction, ten participants mentioned that they became more satisfied with Waterproof since the beginning of Analysis I, 6 students mentioned that they became less satisfied with Waterproof mentioning reasons such as "Waterproof cannot keep up with

the material.” and giving up using Waterproof altogether as “it sometimes took way too much time to figure out what Waterproof wanted”. Six other respondents claimed that their satisfaction with Waterproof did not change since starting to use it.

With regards to cognitive load, 14 respondents mentioned that using Waterproof costs less mental effort than in the beginning, due to having “learnt techniques to prove different statements” and “habits arising” about Waterproof use. Only one student mentioned that the cognitive load was unchanged or increased, respectively.

Finally, 6 participants mentioned that their motivation to use Waterproof increased mentioning e.g. “the satisfaction of seeing that your proof is correct is motivating as the exercises get more difficult”. 7 students claimed that their motivation decreased mentioning reasons such as “it took much more time to solve exercises using waterproof than on paper, but it wasn't any more rewarding”. 9 students mentioned that their motivation did not change since starting to use the tool.

Conclusion

Study 2 served as an iteration to gather feedback about the changes in onboarding. This contributed to answering SRQ1 (How can the onboarding experience of Waterproof be improved?). Results showed that replacing the Waterproof installers to the Canvas page and adding the video about installation made the process clear. Students expressed conflicting opinions on the Tutorial, which suggests different expectations. Therefore, these expectations could be examined further and different versions of tutorials with different purposes could be developed. The feedback also suggests that the visibility of the Waterproof Canvas page should be increased and that more explanation about the syntax, errors and troubleshooting tactics could be provided in the form of handouts or as a part of one of the tutorial versions. These points of feedback were incorporated in the final list of recommendations (Appendix F).

Additionally, Study 2 contributed to answering SRQ2 (How do users engage with Waterproof?). The large range of Waterproof SUS scores and large standard deviation in motivation scores supplemented with conflicting qualitative comments about changes in motivation, satisfaction and cognitive need further exploration. Research should be conducted focusing on finding the reasons for these differences by e.g. trying to distinguish groups of different Waterproof users.

Data collection - Study 3 – User testing/interview

Study aim(s)

The aim of Study 3 was to gain more contextual information about how students use Waterproof. Although logs provide information about unmoderated tool use under preferred conditions by students, logs do not contain data about the interaction aspects of Waterproof use and the reasons behind logged user actions. Interaction aspects can range from triggers that induce using some functionalities to users' preference for a computer mouse, the touchpad of a laptop or hotkeys for interaction. The interaction aspects can impact how users prefer to use the user interface and thus inform design decisions (Reimann et al., 2014). Interaction data is usually collected via observation, which also provide an environment and circumstances for collecting data about an uninterrupted journey, that is not guaranteed in log files. Pairing observation-based research with log analysis in a mixed methods study thus helps understand usage patterns better by explaining reasons behind choices and explanation for user behavior.

Understanding contextual information in case of an ITS is especially important due to the possibility of ITS implementation in educational contexts. Different educational contexts provide different circumstances that result in different interactions and thus different tool use. It is therefore of great help to examine how the ITS is used in the defined context so that it can be adjusted to best support users of the system.

Together with the Waterproof team the following areas of inquiry for Study 3 were identified: (1) contextual information about how students use the following features: Common Tactics, Search functionality, Symbols, Hints, Tutorial; (2) interaction data regarding the usage of hotkeys and code execution; (3) errors and (4) steps taken during proving. Study 3 also gave the opportunity to ask users a series of short questions which were directed to error fixing strategies and circumstances of giving up on an error in Waterproof.

Study 3 thus helped find answers to SRQ2 (How do users engage with Waterproof?) and SRQ3 (Which are the points where users abandon the process of proving mathematical statements in Waterproof?).

Method

Study 3 consisted of two parts. First, a task-based part (Part 1) was an adaptation of the scenario-based usability testing of Granić et al. (2002) performed with the thinking aloud methodology (Erümit et al., 2019). Part 1 catered to answering the inquiries defined by the Waterproof team. Part 1 was followed by Part 2, that included a series of short, contextual questions that contributed to answering SRQ2 and SRQ3.

Part 1 of Study 3 consisted of two tasks and a practice task for thinking-aloud. The practice task for thinking-aloud was performed so that participants get used to commenting on their actions while interacting with Waterproof. As this method could be a novelty for some people, participants were nudged to think aloud throughout the study if it was necessary. The nudging involved the interviewer repeating user actions and asking about what they did. The aim of Task 1 was for participants to complete a proof process uninterrupted, following through their usual steps, while in Task 2 participants were presented with a proof scattered with typos and the aim was to examine how they perform error-correction. All tasks together with the questions for Part 2 can be found in Appendix D. The interviewer in all sessions also used a guide that included the analysis aspects for Part 1

for both tasks and the questions for Part 2 (Appendix D). The interviewer's guide allowed the interviewer to take notes on paper for follow-up questions.

Answers given in Part 2 were analyzed via inductive thematic analysis. The analysis focused on error fixing strategies mentioned by participants and the circumstances of giving up on an error statement in Waterproof and giving up on completing an exercise sheet in Waterproof.

Participants

Study 3 had five participants. Participants were selected based on a first come-first serve basis from the respondents of Survey 2 (Study 2) that indicated interest in participating in an in-person study session. Participants gave their informed consent for participation before the session.

Materials

The tasks for Part 1 were performed in Waterproof in an exercise sheet, *Proofs in analysis* that students already worked on previously during the course. Screen- and voice recordings sessions in Study 3 were made with Microsoft Teams (Microsoft, 2022) and the recordings got transcribed automatically with Whisper (Radford et al., 2022). Data analysis was performed in Microsoft Excel and Miro.

Procedure

All sessions for Study 3 were conducted in the course of one week, between the 12th and 16th of December 2022. This way, all participants had about a month of experience with using Waterproof. The sessions were conducted physically with both the participant and the interviewer present. The sessions took place in the Atlas building on the TU/e campus using a borrowed laptop from the TU/e, with a USB mouse attached. The sessions took 45 minutes. Between sessions the interviewer took at least 30 minutes to organize their notes and reset Waterproof for the tasks. The detailed script for the sessions can be found in Appendix D.

Metrics and sub questions were identified for Part 1 to operationalize the inquiries defined by the Waterproof team. The summary of these metrics can be found in Table 2.

Table 2

Description of metrics in user tests along the areas of inquiry

	Area of inquiry			
	(1) Contextual data about features	(2) Interaction data	(3) Errors	(4) Proof steps
Metrics (objective and subjective)	how the feature was accessed why the feature was accessed successful use of the feature	the hotkeys used means of code execution means of input	overall task completion severity of error (1 - severe; 2 - medium; 3 - mild) number of occurrences number of participants affected by the error task at which error occurred how the error was handled	-

Regarding data analysis, the grading scheme used in usability testing was utilized (Nielsen, 1993). Metrics for (1) contextual data about features, (2) interaction data and (3) errors were quantified via counting and supplemented with qualitative notes. Error severity is defined according to the user's capability to recover from the error. (1) Severe error means that the user cannot recover from it, (2) medium error means that the user can recover from

the error, but not immediately and (3) mild error means that: user can recover from it immediately.

(4) Proof steps were identified qualitatively based on examining interaction patterns during task completion. Interaction patterns were observed on the screen recordings made during the sessions. Follow-up questions about interaction patterns that were unclear were asked in Part 2.

Results

First, results of Part 1 are introduced. Table 3 summarizes the contextual use of features determined in the inquiry by the Waterproof team.

The Search functionality and the Tutorial were not used during the sessions, although two participants mentioned using the Tutorial as a reference for Syntax in Part 2. Common Tactics and Hints were used by one participant each, due to confusion or uncertainty about the syntax or proof structure to follow. Common Tactics were accessed by clicking its icon in the header and were used in Task 1; Hints were accessed in both Task 1 and Task 2. While Tactics were only browsed by the participant, and no other interaction (e.g. inserting a Tactic in the code field) happened, Hints helped the participant start the proving process in Task 1. Symbols were the most often used functionality. They were accessed via hotkeys most often, by three participants, one participant copied them from the local menu of the code block they were working in, and one participant copied them from the proof progress tab.

In summary, only Symbols were used by more than one participant. The reason for feature use was confusion or difficulties with syntax. Features (Table 3) were used in case of both tasks, thus both for writing a proof and fixing errors and finally, feature use helped one participant to proceed with a proof. Consequently, there was variety in feature using behavior of participants, but the user test did not provide enough data to draw general conclusions.

Table 3

Summary of contextual feature use analysis of user tests

Waterproof features	Number of participants engaged in it	Reasons for feature use	How featrure was used: task, access (number of participants)	The outcome of feature use
Common Tactics	1	uncertainty about syntax (1)	for Task 1 (1); accessed pressing the hammer icon (1)	Just browsing (1)
Search functionality	0	-	-	-
Symbols	5	-	for both Task 1 (5) and Task 2 (5) accessed via hotkeys (3) or the code block menu (1); copied from the Proof Progress tab (1)	-
Hints	1	confursion about proof structure (1)	for both Task 1 (1) and Task 2 (1)	Starting Task 1 successfully (1)
Tutorial	0	-	-	-

Regarding interaction data, all five participants used typing as a means of input, two participants used copy-paste, either their own code, or from the proof progress tab and one participant used the Symbol library. Furthermore, hotkeys were used for adding symbols by three participants, for code execution by three participants and for adding a code block by one participant. At last, code execution happened either as a result of clicks or by using hotkeys. Two participants clicked on the symbols at the end of sentences for sentence execution and two participants used the execute command from the commands menu. Three participants that used hotkeys used *alt + down* to attempt to execute the next sentence and one participant used *alt + end* to execute all code blocks in the exercise sheet. It is important to mention that participants used multiple different means of interaction in the sessions. Furthermore, symbols are vital parts of proofs, which is why their reasons for use and outcomes of use are

not detailed; and Hints can only be accessed from the exercise tab, which is why their access is not represented in Table 3.

In summary, all participants used typing as the primary means for entering code into Waterproof. All participants used hotkeys, mostly for code execution and inserting symbols. Code execution was most frequently done by executing the next sentence with a hotkey, as this is the fastest way to quickly check a line of code written.

Table 4

Summary of interaction analysis of user tests

Means of input (number of participants)	Aims of hotkey use (number of participants)	Means of code execution (number of participants)	
		By clicking	By using hotkeys - *hotkey*
typing (5) copy-pasting (2) insert from Symbol Library (1)	inserting symbols (3) code execution (3) adding code block (1)	clicking on the execute symbol at the end of the sentence (2) using the "execute next" command from the commands menu (2)	attempting to execute next sentence - *alt* + *down* (4) attempting to execute all code in the exercise sheet - *alt* + *end* (1)

Regarding task success, all participants completed the practice task for thinking aloud successfully and all but one participant finished Task 1 and Task 2, as well. Six different kinds of errors were registered altogether, that are summarized in Table 5.

All errors were of mild severity meaning that participants could recover from them on their own. Only one error occurred in case of more than one participant: all but one participant ran a single focus goal error in Task 2. The other cases of errors are detailed in Table 5.

While analyzing each participant's journey of completing tasks, several patterns emerged. These patterns were compared manually with the logs collected during the sessions. The outcome of this process, i.e. a preliminary map of an uninterrupted proving sequence is shown in Figure 8. The process shows a general pattern aggregated from all five participants' interactions with Waterproof. A step was only added to the map if at least three participants performed it. If an interaction or step was unclear, a follow-up question was asked for clarification after completing the tasks. E.g. in Task 2 a participant varied their means of interaction during the task, which was later explained as a consequence of using a different operating system than what they are used to. The aim of drawing up the process map was to draw identify logs corresponding to these steps. The explanation for logged activities can be found in Study 4, Table 7.

In general, participants starting to work on exercises can be marked with *focusing-block* in logs. Participants also mentioned that they sometimes switch between exercises, which can be marked by executing *Admitted.* in a block and focusing another block.

Focusing more on steps performed during working on the exercises, the following results unfolded. Participants executed sentences immediately after editing them, which either resulted in a successful sentence execution, Waterproof being stuck (no error message) or an error case. In case of a successful sentence execution, a *work cycle* proof step can be defined as the pair of logs: *coq-exec-next* and *coq-success-sentence*, Waterproof being stuck (*stuck cycle*) is indicated by *coq-next-beyond-sentence* in logs and errors (*error cycle*) can be indicated by multiple logs depending on the type of error, which is detailed in Study 4.

Other learnings from the sessions were that participants often trail back multiple sentences (*coq-exec-prev*) in the proving process to check the proof progress tab for more information to continue the proof. This step can be called *check proof progress*. Using various combinations of Waterproof features, such as Common Tactics, Search functionality,

Symbols, Hints and the Tutorial can be used to get out of a *stuck cycle* or an *error cycle.* A proving process can result in either successful exercise completion (in logs: *successfully executing Qed.*), switching between exercises or giving up (not *successfully executing Qed.*).

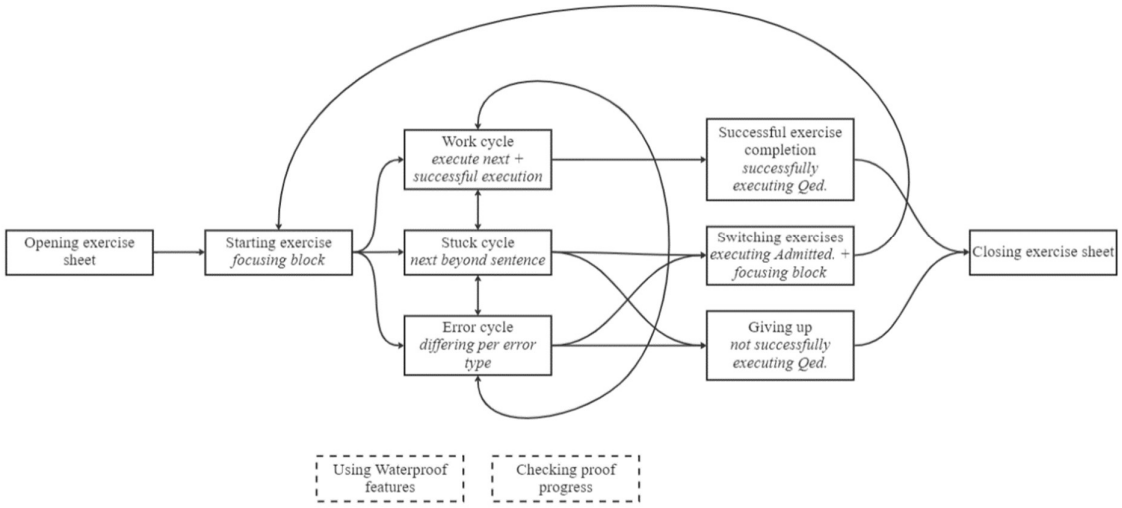
Table 5

Summary of error analysis of user tests

Error	Error severity - 1 - severe; 2 - medium; 3 - mild (number of participants)	Total number of occurrences	Occurrence (number of participants)	Task	Handling
Uncaught Ltac exception: TakeError	3 (1)	1	1	Task 1	Change “take” to “assume”
Expected a single focused goal but 2 goals are focused	3 (3) 2 (1)	5	4	Task 2	Added missing signs.
Uncaught Ltac exception: BothDirectionsError	2 (1)	2	1	Task 1	Browsed tactics and changed direction to continue the proof
[Focus] Wrong bullet-: current bullet- is not finished	3 (1)	1	1	Task 2	Finishes the statement
Expected a single focus goal but 0 goals are focused	3 (1)	1	1	Task 2	Finishes the statement
Uncaught Ltac 2 exception: TakeError	3 (1)	1	1	Task 2	Deletes sentence

Figure 8

The preliminary process map of working on an exercise sheet in Waterproof based on user test data and manual analysis of the corresponding log files



This section details the results of thematic analysis of participants' answers to the questions in Part 2. First themes that are related to error fixing strategies are introduced. The main themes include *checking for typos* in proofs, which is mostly directed at fixing syntax errors and usually include checking brackets, periods, symbols, spacing, capitalization and signs. All participants mentioned brackets when thinking aloud, one participant even said while fixing a typo in Task 2 "… so, it's probably the brackets, cause normally it's the brackets…". Another main theme was *using external help* in proofs. which includes the *using written materials* subtheme. Written materials include the Waterproof Tutorial, previously written code by the participant or information from the proof progress tab. The other subtheme that emerged within using external help was *asking for help from other people*, such as instructors and groupmates. A participant said: "sometimes I also come back to the tutorial of Waterproof to find whether there is a syntax that I can use". Two further themes were *rewriting* and *expanding on the proof*, which refers to some participants not focusing on

avoiding errors in Waterproof but concentrating more on using Waterproof as a platform to show their proof structure. One of the participants explained this train of thought the following way: “if the proof is right, Waterproof is okay with it”. Thus, they prefer to start the proof from scratch choosing another approach instead of checking their work line-by-line, looking for errors. The last main theme was *trying “random stuff”*, which was referred to by participants as the approach to follow when error messages were not clear enough to act on them directly. This theme is related to the *using materials* subtheme, and all main themes due to its versatility and ambiguity. A participant said “sometimes the errors are not very clear, so I just try random stuff and it will work. But that's not a very good comment for research stuff.” Later, the participant was assured that it is indeed a good comment for “research stuff”.

In this section the themes regarding “giving up” in Waterproof are discussed. First, it is important to note that the question about “giving up” was initially directed to giving up completing the exercise sheet in Waterproof. However, as one of the main themes also shows, participants were very persistent about Waterproof use, saying “we stick to Waterproof as much as we can”. They also claimed that even if they cannot solve an exercise for the first try, they ask for help and return to it later. Main themes include giving up if participants know that *the overall proof process* is right and when there is a *significant difference between what Waterproof accepts and how the proof on paper looks like*. Some participants voiced their thoughts regarding this issue the following way: “it wasn't a problem of me not knowing how to solve the exercise, it was me not knowing what to write in Waterproof so it will accept it”. Another main theme was that participants give up on an error when the *same one repeats even after trying the error-fixing strategies.* A participant said “[I give up]… if there's one error and it's the same error and I have rewritten it maybe three times and it's almost a deadline”. The final main theme was that students gave up on an error when

it took too much effort to follow through with solving it, knowing that finding a solution would not change the *grade* they get for the exercise. A participant phrased it like "[people] think Waterproof just wastes so much time, and from the efficiency perspective, they choose: I can write it in the hand proof [sic]." and another participant said "since you have the knowledge, you're not going to just get a lower grade because you cannot do it on Waterproof." It is important to note that participants also included their groupmates' views in answering the questions.

Conclusion

Results of Study 3, Part 1 showed that contextual feature use was difficult to examine due to low engagement with features. This can be due to participants being asked to work on an exercise sheet they were already familiar with and thus them remembering the proving process, and not needing to use the features as crutches. Another reason could be that participants had a month of experience of Waterproof use at the time of the user tests. One of the participants even highlighted that initially they used the symbols library, but once they became more comfortable with Waterproof, they switched to using hotkeys, as that way it is faster to work with the tool.

Results about interaction data suggest that users make use of the provided palette of means of interaction. They have more than one preferred way of interacting with Waterproof, regarding hotkeys, code execution and input. However, observations showed that users who used clicks for testing code had to click back-and-forth between panels to continue coding, which might result in less efficient use of Waterproof.

Participants' reactions to errors, rare occurrence of most errors and their low severity suggest that errors do not mean large obstacles for participants in Waterproof. Furthermore, the systematic approach to general error-fixing that unfolded from the thematic analysis shows that errors rarely caused participants to stop working on the exercise sheet. However,

during the user test, some participants showed the behavior of not reading error texts and thus not acting on errors specifically. This approach corresponds with the attitude of concentrating on the proof process and not on errors in Waterproof, that was an outcome of the thematic analysis on giving up on errors.

In conclusion, Study 3 contributed to answering SRQ2 (How do users engage with Waterproof?) via providing the findings related to interaction between users and Waterproof. Study 3 also provided a preliminary process map of working on an exercise sheet in Waterproof. These findings about proof steps were further used in Study 4. Furthermore, Study 3 helped answer SRQ3 (Which are the points where users abandon the process of proving mathematical statements in Waterproof?), as well. The general attitude of Study 3 participants showed that they are very persistent in using Waterproof and they return to working on the exercise sheet even if they failed the first time.

Data collection - Study 4 – Process mapping based on log data

Study aim(s)

The aim of Study 4 was to analyze the large quantities of unmoderated log data that were collected in Study 2. The aim of the analyses was to answer SRQ2 (How do users engage with Waterproof?) and SRQ3 (Which are the points where users abandon the process of proving mathematical statements in Waterproof?). More specifically, the aim of the study was getting a general overview about the interaction sequences with Waterproof. Special attention was given to studying how often and how four different kinds of errors occur and the consequences of these errors. Additionally, the conditions of leaving Waterproof, i.e. the steps taken before closing the program were examined.

Method

Participants

Participants of Study 4 are the same as participants of the survey studies, i.e. Study 2. As mentioned earlier, only data of participants who filled in both surveys were included in the analyses.

Materials

The list of all logged information can be found in Appendix E. The definition of activities and data preparation for process mapping was performed in R (The R Foundation, 2023). Fluxicon Disco (Fluxicon BV, 2023), a process mining tool was used for data analysis. Disco builds the process maps along a set of predefined attributes. The definition of these attributes in the collected log data can be found in Table 6.

Procedure

Logs were collected from Waterproof between the 14th of November 2022 and 16th of January 2023 with the exception of one participant, who sent their logs later. This participant has logs dating until the 23rd of January 2023.

Table 6

Explanation of the attributes required for a Disco process map in the present study

Disco attributes	Meaning of disco attributes in this study
Resources	Participants
Time	Time of the activity (described as a point in time)
Case	a work session: defined between starting and closing Waterproof
Activity	interaction with Waterproof

Following data collection and aggregation, data cleaning was performed. It included removing sessions where respondents did not work on exercise sheets and activities during the instruction sessions where users could get help from instructors. Interactions during instruction hours were removed to ensure unmoderated data collection. Data cleaning also involved aggregating multiple logs of e.g. closing, removing logs without information value (e.g. *heartbeat*), defining sessions and defining an absolute time stamp for all logs. The details of data cleaning can be found in Appendix E.

As a next step, labels were defined for actions related to directly working on the exercises (e.g. working on a proof, successfully completing an proof, etc.) and feature use (e.g. using common tactics, hints, etc.). Additionally, the four error categories were also distinguished in the logs. The explanation for activity labeling can be found in Table 7.

Once the activities were defined, they could be fed into Disco. The logs were summarized in a process map depicting session frequencies and session coverage (Figure 9). Session frequencies mean the number of occurrences of an activity in a session on average. They appear in Figure 9 as the numbers in activity boxes (e.g. *working on proof* - 46). Session coverage means the percentage of cases that included the activity (e.g. *working on proof* – 97.9%). For the sake of readability of the process map, only the most prominent connections between activities are included in Figure 9. The depicted paths were set to 25% (0% meaning only the most dominant connections depicted and 100% meaning all paths shown).

Table 7

Explanation of activity labeling, data preparation for the process map

Activity	Meaning of activity	Type of activity
Working on proof	the user successfully executes the next sentence in the proof	Proving activities
Exercise switch	the user switches between exercises (only if Exercise qed does not happen)	
Exercise stuck	the user is stuck on an exercise, i.e. cannot execute the next sentence, but does not get an error message	
Exercise qed	the user finishes a proof in the exercise sheet, i.e. executes "Qed."	
Execute all	the user executes all code in the exercise sheet	
Error syntax	the user encounters a syntax error	Errors
Error add	the user encounters an add error	
Error nested proof	the user encounters a nested proof error	
Error other	The user encounters an error that is different from the three above	
Hint	the user clicks a hint panel in an exercise sheet	Using Waterproof features (Troubleshooting)
Search	the user clicks the search button, i.e. searches a term (in the rightside panel on the Waterproof ser interface)	
Tutorial	the user opens the Waterproof Tutorial while working on an exercise sheet	

Note. The activity of clicking hints by definition does not distinguish between different hints and whether the hint was open or closed when clicked (clicking on an open hint also closes it in Waterproof).

Results

Figure 9

The Disco process map of users engaging with Waterproof

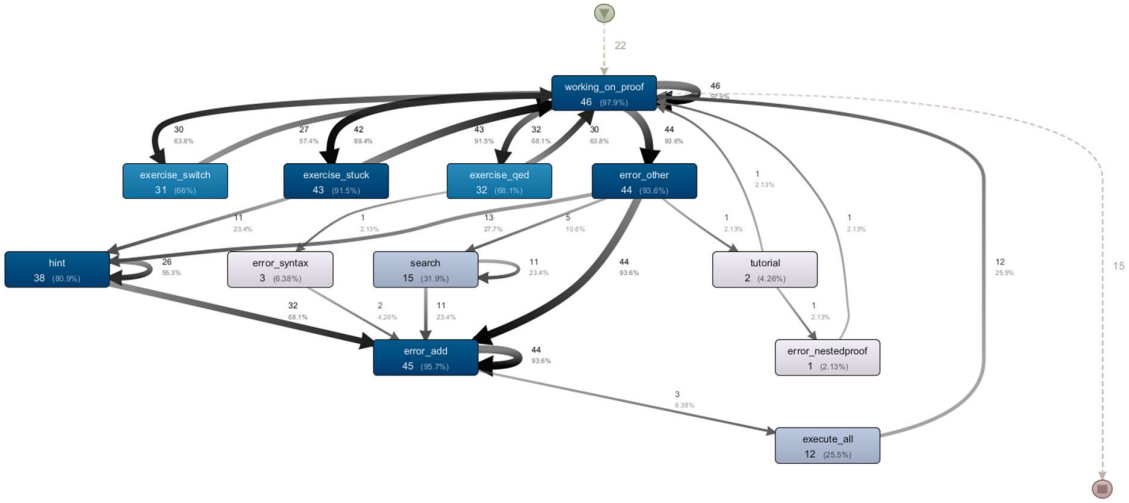


Figure 9 depicts the process map created based on the log data collected from students. First, the following main sequence of actions can be identified. Upon starting to work on an exercise sheet, students immediately start *working on proofs*, by writing and executing sentences. As a result of *working on a proof,* students can either successfully finish the proof (*exercise_qed*) or run into an obstacle. These obstacles can take several forms. Students either encounter an error (*error_other*), get stuck with coding (*exercise_stuck*) or move on to another proof (*exercise_switch*). The process map also shows that students sometimes fall from one type of error to another.

According to the process map, students characteristically solve these issues independently and return to *working on the proof*. However, in case of being *stuck,* students turn to the features of Waterproof, such as *hints.* According to the process map, once a student turns to hints, they are likely to utilize hints multiple times, creating a hint cycle.

Hints, however, also often lead to errors (*error_add*) instead of helping students return to *working on proofs*.

Errors interrupting *working on the proof* usually lead to different kinds of errors (*error_add*) which induce an error cycle difficult to escape. Other strategies to overcome errors include using the *search* functionality, the *tutorial* or *hints*. The process map shows that using the *tutorial* is most likely to lead back to *working on the proof* although not directly. *Search* activity can lead to a search cycle or is transferred to the error cycle (*error_add*) mentioned above.

The process map does not carry valuable information about different ways of finishing a(n unsuccessful) session. It shows only that students close Waterproof while *working on a proof*.

It is important to mention two further sequences depicted in the process map. First, after completing a proof (*exercise_qed*) the map shows students either running into errors (*syntax_error* and *add_error*) or returning to *working on proofs.* Second, leaving the *add_error* cycle, students tend to execute all code in the exercise sheet. These sequences should be further examined by refining activities and diving deeper into the sub-sequences in a further study to build more understanding about the reasons of these patterns.

Conclusion

Results of Study 4 show that although students are stopped by different obstacles in the process of working on completing a proof in Waterproof, they most often recover from the errors or being stuck with the proof not using any Waterproof features for troubleshooting. The process map shows after which actions students use the features offered by Waterproof (hints, tutorial, search) and the outcomes (actions) of using these features. Additionally, the process map depicts how different errors lead to other errors and an error-cycle and suggests that recovery from errors does not happen characteristically. Finally, the

process map does not provide valuable information about the steps preceding closing Waterproof.

In conclusion, Study 4 contributes to answering SRQ2 (How do users engage with Waterproof?) by showing sequences of interactions of students with Waterproof and providing information about encountering errors and disruptive events of working on exercises. However, contrary to expectations, the process map does not provide conclusive information to contribute to answering SRQ3 (Which are the points where users abandon the process of proving mathematical statements in Waterproof?). Refining activities could help gain more insights about the activity sequences following completing a proof and returning to working on a proof after errors.

Aggregated results

This section contains additional results that are not direct outcomes of a data collection studies but contribute to answering sub research questions of this study. Table 8 shows the number of student groups that submitted assignments in Waterproof during Analysis I. It is important to note that groups were included if they handed in at least a partial submission; and that failed submissions are also included. The data shows a monotonous increase in the submission of each assignment over the years. Table 8 also shows how in the latest year more groups kept submitting assignments throughout the course than in the previous years. One of the reasons for this increased retention can be the change in Waterproof user onboarding, as clear user onboarding was found crucial for high retention rates of e.g. commercial products (Lindgaard et al., 2011). The changes could also be a reason for assignments slightly changing over the years or the improvements in Waterproof. Direct causes are difficult to pinpoint. The high retention rates are supported by qualitative data: feedback provided by students in Study 2, Survey 1 described the installation process as

"smooth" and "clear". Additionally, informal feedback from the instructors shows that they explicitly noticed students' persistence with Waterproof.

Table 8 also suggests that student participation first drops at Assignment 4 and there is another significant drop at Assignment 7. This pattern is present in both 2021 and 2022 and suggests that it is worth examining study load or the material included in these assignments and implement alterations or provide students with more support in these periods.

Table 8

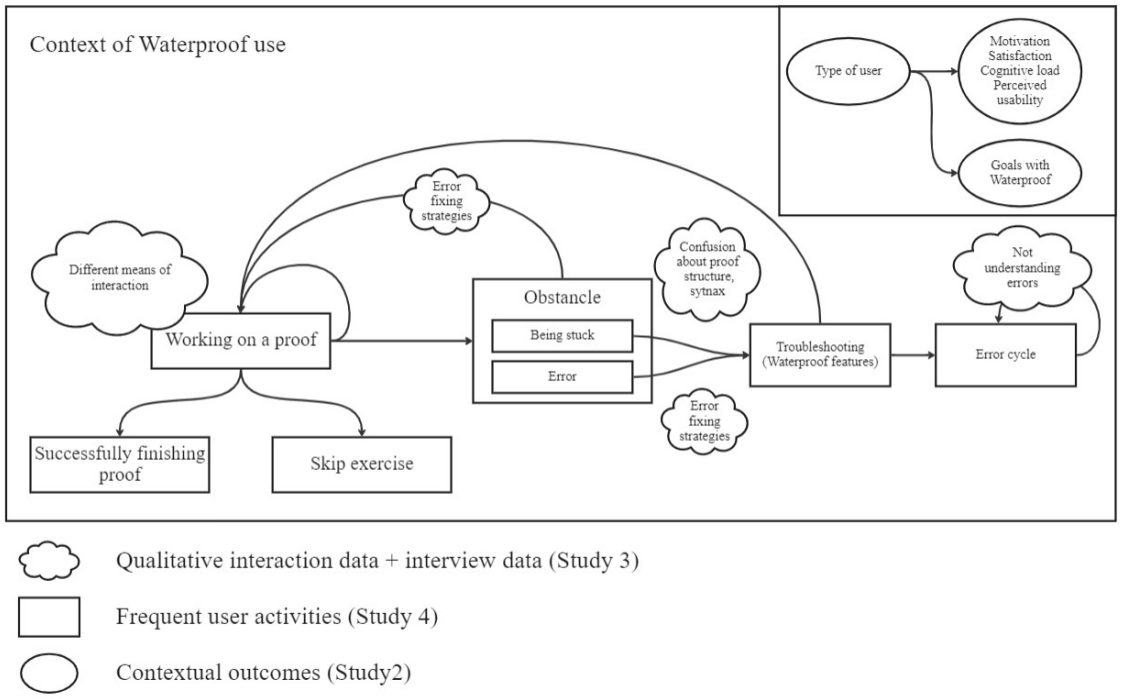
Waterproof assignment submissions in 2020/21, 2021/22 and 2022/23.

Assignment number	2020/21	2021/22	2022/23
1	15	16	25
2	6	14	25
3	4	14	25
4	3	10	19
6	2	11	20
7	0	3	12
9	0	11	17
10	0	9	19

Figure 10 aggregates the results of Studies 2, 3 and 4 and has the purpose to answer SRQ2. The foundations of the map in Figure 10 are the outcomes of Study 4: frequent user activities. These activities are supplemented by observations of interaction with Waterproof, which are outcomes of Study 3. Finally, the contextual additions resulting from the surveys in Study 2 are shown in ovals in the top right of the map.

Figure 10

The answer of SRQ2: a map of user engagement with Waterproof



The main activity users exhibited while engaging with Waterproof was working on proofs by writing code, and executing sentences using hotkeys, mouse clicks and other means of interaction detailed in Study 3. The aim of exercise sheets is for students to practice proving in Waterproof, which explains working on a proof being the most frequently logged activity. Experimenting with different formulations and testing code by executing it is a cyclical process which explains the working on a proof-cycle in the process map. This behavior was also observed during the user testing sessions in Study 3.

The working on a proof cycle can be ended by successfully finishing a proof or skipping to another exercise in the same exercise sheet (in case the proof was not finished). Users can encounter different obstacles while working on a proof. Being stuck on an exercise and bumping into errors can be fixed individually, without any external help by error fixing strategies (listed by participants in Study 3), such as looking for typos in the code and reading

the code over and over. Another way of overcoming these obstacles is using Waterproof features like Hints, Common Tactics or the Tutorial. These troubleshooting practices are present both in the process map of Study 4 and the qualitative outcomes of Study 3. Users can also run into error cycles. The reason for this might be not understanding errors, which can stem from not reading the error messages thoroughly, a user behavior observed in Study 3. Errors can also signal uncertainty about the syntax, in which case students turn to the Tutorial. If they cannot find the right syntax, they repeat the same error while trying to find a working solution, which results in an error cycle.

Besides answering SRQ2, Studies 3 and 4 had some unexpected findings. First, with regards to interaction with Waterproof examined in Study 3, students used the proof progress tab to copy expressions from and aid their thinking about the proof structure. Although this is invited in some Common Tactics, users also went back in their code (*execute_previous*) to take another look at the proof progress tab. The behavior of using the proof progress tab was not reflected in logs, since contents and interactions in the proof progress tab are not logged. Examining the logs collected during the user testing sessions, executing the previous sentence multiple times could be a sign of this behavior but more elaborate logging is necessary to identify it as an activity with certainty.

Other unexpected observations are related to the process map in Study 4. One of the unclear processes was users characteristically continuing to work on a proof (although not necessarily the same proof) after finishing an exercise. The expected step after finishing an exercise would be to close Waterproof.

Another outstanding finding in Study 4 was related to executing all code in the exercise sheet after an error loop that leads back to working on a proof. Executing all code occurred during Study 3 only in case of one participant, who used it before starting to work on exercises and after finishing everything to check if the exercise sheet is intact.

General discussion

This study was conducted to explore the possibility of using a service design approach to increase the prominence of user-centered principles in *learning experience*-focused ITS evaluation. The study introduced an ITS evaluation framework built on service design principles and presented a case study of evaluating Waterproof using this framework as a proof of concept. This approach was aimed at addressing the gaps identified in the ITS evaluation literature, namely the lack of user-centeredness, contextuality and the reluctance to make use of user-centered methodologies that allow mixed method study designs (log data analysis and examining user onboarding).

This study is organized around a main research question: *How can the gaps in learner experience-centered ITS evaluation be addressed by utilizing a service design approach?* The case study is supported by three inquiries that are considered sub research questions of this study. As certain studies contributed to answering several research questions, Figure 11 summarizes the outcomes of each study and how the studies contributed to answering the SRQs and the MRQ.

Answering SRQ1 (How can the onboarding experience of Waterproof be improved?)

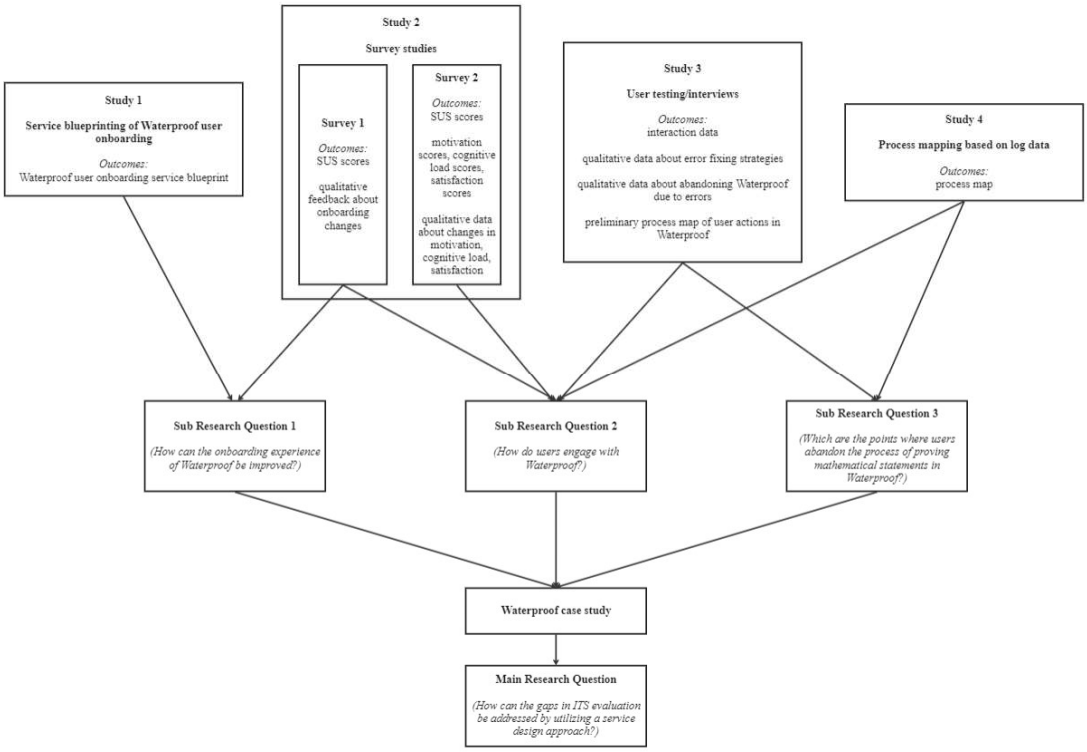
Answering SRQ1, as a result of Study 1, a list of recommendations to improve onboarding were determined based on pain points identified in the service blueprinting session (Table 1). A part of these recommendations was implemented in the 2022/23 academic year. The final list of recommendations, including feedback from Study 2 can be found in Appendix F.

Examining the user onboarding of Waterproof using a service blueprint created in a co-creation session thus resulted in successfully identifying pain points and a list of actionable suggestions to improve onboarding. The improved onboarding process places

emphasis on aiding the completion of training (Renz et al., 2014) via additional video instructions to the Tutorial and utilizes the *first impression bias* to work *for* Waterproof. This is ensured by removing a potentially unfamiliar platform (GitHub) from the onboarding process and presenting Waterproof as a part of the course (via Canvas) (Lindgaard et al., 2011). Furthermore, the high user retention rates (Table 8) can also partially be attributed to the new onboarding process (Cascaes Cardoso, 2017).

Figure 11

The structure and outcomes of this study



The success of Study 1 supports the applicability of the *sequencing* service design principle (Stickdorn & Schneider, 2010) in the ITS evaluation framework. Examining the onboarding process separately helped identify the pain points in the first impression that also affect engagement (Lindgaard et al., 2011) and *co-creation* contributed to identifying the pain points successfully (Stickdorn & Schneider, 2010).

Answering SRQ2 (How do users engage with Waterproof?)

Figure 10 depicts all the information collected in Studies 2, 3 and 4 combined and so is the answer to SRQ2. Results of Study 3 and 4 provide a picture of user engagement with Waterproof, while results of Study 2 suggest great differences in users' perceived usability of Waterproof (Figure 7). SUS scores of Waterproof dropping slightly over time contradicts the literature, stating the tendency of increasing usability scores due to familiarity with the tool (Hassenzahl, 2007). Interestingly, the qualitative outcomes of Study 2 support increasing familiarity with Waterproof over time. Similar qualitative contradictions are reflected in the changing motivation and cognitive load of using Waterproof.

Answering SRQ2 was focused on the *service period*, i.e. when users actively engaged with Waterproof. A *user-centered* mixed methods study design was successfully applied to map the *holistic* process of using Waterproof for working on exercise sheets (Stickdorn & Schneider, 2010). The process map supplemented with qualitative data from in-person user testing sessions (Granić et al., 2002), usability data (Lynch & Ghergulescu, 2016) and studying user motivation, cognitive load and satisfaction constructs shed light on Waterproof use from several different angles. The outcome of this mixed methods study design (Mertens, 2017) helped identify obstacles users encounter and highlight new directions to further refine the understanding of user behavior.

Answering SRQ3 (Which are the points where users abandon the process of proving mathematical statements in Waterproof?)

First, it is important to define the meaning of abandoning the process of proving mathematical statements in Waterproof. Abandoning Waterproof can mean stopping to use it to submit assignments throughout the course Analysis I. This definition allows answering SRQ3 via examining retention rates of Waterproof (Table 8).

Another definition of abandoning the proving process in Waterproof can mean giving up on solving an exercise. The points of abandonment in this definition are meant as reasons and circumstances of abandonment and were studied in Studies 3 and 4; however, the data collected does not yield conclusive answers to SRQ3 (Figure 10).

Answering SRQ3 was targeted at examining the circumstances and reasons for entering the *post-service period.* The same mixed methods, user-centered approach was utilized to answer this question as answering SRQ2 (Mertens, 2017; Stickdorn & Schneider, 2010). Refining sessions in the process map could contribute to having more information about abandoning Waterproof and successfully answering SRQ3 in future work.

MRQ (How can the gaps in ITS evaluation be addressed by utilizing a service design approach?)

First, this study places users in the center of the evaluation along the *user-centered* service design principle. This principle is an overarching theme in the ITS evaluation framework. Examining user onboarding is a user-centered method (Cascaes Cardoso, 2017; Terres et al., 2019), that follows users in getting to know a software program. Additionally, the mixed methods mapping of user engagement is also performed with a strong user-centered approach. User actions in Waterproof are mapped based on log data analyses which are supplemented with information resulting from interaction with real users of the system (Granić et al., 2002). This approach paints a picture about Waterproof from the users' perspective, which helps understand the needs, frustrations and aims of users (Still & Crane, 2017a). Designing an evaluation framework with a service design outlook contributed to reintroducing user-centeredness into ITS evaluation studies (Granić et al., 2002; Miller, 1988). However, only a subset of Waterproof users were involved in these evaluation studies, which introduced participation bias into the results of user-centered evaluation.

Second, including students, instructors and representatives of the Waterproof development team in Study 1 is an example of utilizing the service design principle, *co-creation* (Granić et al., 2002; Virvou & Tsiriga, 2000). It adds to the examination of user onboarding by providing information from multiple angles about pain points. Providing an opportunity for students and instructors to talk to each other in the blueprinting session helped clarify reasons behind user actions immediately and without involving the interpretation of the evaluator. Involving instructors and the development team via the service design approach also helped quickly highlight solution directions by having stakeholders, knowledgeable about with course organization and Waterproof development possibilities, on location. Student pain points however could be withheld due to instructors being present which could cause intimidation in students.

Third, the *sequencing* service design principle contributes to building the structure of the evaluation framework by introducing the distinct examination of *pre-service and service periods* (Stickdorn & Schneider, 2010). This approach led to improving onboarding based on the outcomes of Study 1 and a detailed analysis of user engagement of Waterproof. This resulted in recommendations for solving user frustrations and understanding how users interact with an exercise sheet. Besides distinguishing between periods, *sequencing* also placed using Waterproof on a timeline that highlighted the connections between different periods of Waterproof use. The initial frustrations about unclarities in the Tutorial (regarding syntax and errors) were reflected in tool use mapped in Study 4 and appeared in observations in Study 3. However, these patterns can also be the result of the activities being defined along observations of Waterproof use in Study 3. This approach could introduce a confirmation bias in the results.

Finally, the *holistic* service design principle adds to ITS evaluation by including contextuality, another overarching theme in the framework (Stickdorn & Schneider, 2010).

The case study did not examine Waterproof as an entity floating in vacuum, but considered it as a part of the Analysis I course. This approach promoted interpreting the outcomes of the data collection studies in the educational context (Wolfe, 2020), considering e.g. the eight-week timeline of the course in log data collection. Contextuality also put emphasis on examining students working on exercise sheets and needing to submit assignments for the course. This is why exercises were considered in both log data analysis (Study 4) and in user tests (Study 3). Furthermore, most of the recommendations, such as the onboarding suggestions and student frustrations are *holistic*, i.e. applicable to Waterproof *in the context of* Analysis I. *Holistic* evaluations however should also consider groups and means of collaboration on Waterproof exercises, which were not examined this study.

Consequently, including service design principles in the design of ITS evaluation successfully addresses the identified gaps in the literature. *User-centeredness* appeared in all studies conducted and supplemented with *co-creation* addressed the neglect of instructors and the ITS development team in ITS evaluation. *Holistic* evaluations introduced contextuality in evaluation and the *sequential*, iterative structure of the evaluation framework promoted using a mixed-methods evaluations. This design allows for the utilization of user-centered methods such as examining user onboarding and log data analysis. Additionally, the new approach to ITS evaluation provides opportunities for divergence and exploration highlighting possibilities of further research. This underlines the scientific relevance of this study, as it reintroduces user-centeredness into the literature (Granić et al., 2002; Miller, 1988) and provides the scientific community with an entirely new ITS evaluation framework addressing *learner experience*-based ITS evaluation utilizing service design principles.

The practical relevance of the study is directed towards practitioners designing and evaluating ITSs. The introduction of a service design-based framework opens opportunities to involve different user groups and stakeholders into ITS evaluation. In this study, this

involvement was reflected in the service blueprinting co-creation session, which allowed for successful collection and in-depth understanding of user pain points in onboarding from multiple viewpoints (Wolfe, 2020). The focus on contextuality introduced by the framework adds to the understanding of ITS use in education, considering course structures, student goals, study materials and educational assessment (Mark & Greer, 1993; Mousavinasab et al., 2021). This approach helps to understand student interactions with ITSs and prioritize functionalities in ITS design and evaluation that contribute to successfully completing courses. Furthermore, user-centeredness promotes design and evaluation for specific user groups in specific contexts (Still & Crane, 2017b), which is in line with the ITS-specific evaluation approach of Shute and Regian (1993). This specific approach requires more resources than a generally applicable ITS evaluation framework (Siemer & Angelides, 1998) but caters to in-depth ITS analysis.

Additionally, due to the framework being a novelty, only tested in one case study, it has several limitations that reduce its current applicability in empirical ITS evaluation studies. The next section lists these limitations and possible ways to address them to lower the threshold of the application of the framework in the educational context.

Limitations and future work

The first limitation of the framework is that it requires high involvement of the evaluating professional in the ITS project. It is first reflected in the preparatory phase that includes learning about the ITS in detail, which requires time and effort from the professional performing the evaluation. This high involvement throughout formative evaluation work including multiple iterations can also introduce experimenter bias in the results of the evaluation by e.g. asking leading questions or subconsciously misinterpreting participants' answers. Due to the five-month time limitation of this project and the framework being iterated only once, the evaluator's involvement was limited. Additionally, the evaluation was

performed by a researcher outside the Waterproof development team, which again ensured impartiality. In future work, the involvement of the researcher could be controlled for by different researchers evaluating different periods of ITS use. Additionally, further refinement of the framework might automatize high involvement phases of evaluation. Such a refinement combined with learnings from further case studies would also contribute to higher generalizability of the framework (Siemer & Angelides, 1998).

The second limitation of the study is participation bias. In case of ITS evaluation, especially performed in an educational context, participation in evaluation studies is voluntary. This can lead to a selection of participants already having a deep interest in the subject and thus evaluation results could be overly positive. In case of this study, data collection was combined with filling in a survey, which introduced a higher threshold of log data collection. Additionally, participants of Study 3 were survey respondents further filtered by interest in a high involvement, in-person session. Results confirm the bias introduced by this double filtering as while Study 3 participants characteristically expressed very high persistence of Waterproof use, Study 2 participants expressed changing to complete proofs on paper due to time constraints or frustration. Regardless of participation bias, this study was able to provide a broad range of recommendations based on participant data. Participation bias is difficult to address, but in future work, more attention could be given to controlling for this type of bias by e.g. comparing participants' course grades and to the average in the year. This way it could be visible if participants received higher grades, a sign for higher involvement in the course.

The final major limitation of the study is related to log data analysis. In this report, the results are outcomes of the first iteration of session and activity definition due to time constraints of this project and performing only one case study for assessment. Log data analysis resulted in unexpected findings in the process map and could be addressed in a

future study. Unexpected findings include users executing all code in an exercise sheet after breaking out from an error cycle, users returning to working on a proof after finishing an exercise instead of closing Waterproof and using hints leading to errors. These findings suggest that the definition of activities for inclusion in the process map could be revised and that redefinition of activities might affect other parts of the process map, as well. This might change the conclusions drawn from the process map.

Future work should consider multiple versions of activity definition to compare process maps and identify possible mis-definitions. The following considerations could be implemented in future work: (1) redefining sessions by opening a new session after e.g. 10 minutes of user inactivity to examine real sessions and avoid biased results caused by users leaving Waterproof running for weeks. (2) Reiterating whether sessions during instruction hours should still be included as the help of the instructor cannot be identified from the logs. (3) Adding the beginning and end times of activities would allow assigning a length to activities and examine e.g. how long users are stuck in error cycles. (4) Distinguishing between exercises when working on a proof could explain the unexpected finding about user behavior after finishing an exercise and (5) separate analysis of sequences for troubleshooting different types errors could explain how users fall from one kind of error into another or why they would execute all code after an error cycle. In addition to refinements in the log data preparation, trying to create clusters of users and match the users with different goals with the ITS could help understand personalized ITS use better contributing to achieving ITS goals (Mousavinasab et al., 2021).

Conclusion

Intelligent Tutoring Systems (ITSs) have learner-centered goals, such as providing educational guidance when a human tutor is not present and personalizing education (Chughtai et al., 2016; Mousavinasab et al., 2021) Evaluation is crucial to assess whether

ITSs meet these goals. There are three main approaches distinguished in ITS evaluation: *system performance, learner performance* and *learner experience*-based assessment. This study focused on *learner experience*-based assessments. *Learner experience* is not well-defined in ITS evaluation research and is mostly measured by usability assessments. However, usability is only a part of *learner experience* (user experience) (Hassan & Galal-Edeen, 2017), which creates the need for more comprehensive *learner experience*-centric ITS evaluations. In order to address this need, this study introduced a user-centered ITS evaluation framework utilizing service design principles. The framework supports user-centered ITS evaluation designs that consider contextual ITS implementations and support the utilization of user-centered methods such as the examination of user onboarding and log data analysis.

The framework was applied to a case study: the evaluation of Waterproof, an ITS aimed at helping students learn to prove mathematical statements. The results of the evaluation contributed to improving the experience of using Waterproof by addressing pain points related to the onboarding process, i.e. starting to use Waterproof. The outcomes of the evaluation also helped understand the sequences of user actions in Waterproof via information about what frustrates users, what they struggle with and lack from Waterproof. For example, the activity sequence map pointed out error cycles in the activity sequences which was in line with the lack of a comprehensive collection of errors and a troubleshooting guide mentioned by students. The contents of the map contributed to improving Waterproof by clarifying students' frustrations, how they are related to patterns in using Waterproof and giving directions on how to address them. Using the service design-based ITS evaluation framework for Waterproof had several advantages over conducting solely a usability evaluation. The iterations in the framework provided the opportunity to implement feedback and test suggestions for improvement instead of conducting evaluation only once. Utilizing

service design in the evaluation also provided a holistic map of Waterproof use supplemented by contextual information, specific to the university course Waterproof is implemented in. Learning more about Waterproof in the implementation context allowed for targeted, actionable recommendations that contribute to improvements more directly than a usability score would.

The advantages of using the service design-based evaluation framework are not specific to Waterproof. Utilizing service design presents an entirely new outlook on ITS evaluation. Broader *learner experience* assessment can uncover new depths of user engagement with ITSs, provide more information about users and therefore it can support the personalization of education better than usability evaluations. Furthermore, service design introducing contextuality into evaluation studies provides information about how an ITS works in practical education. Contextual assessments allow examining ITS use on a specific educational timeline e.g. over the course of a semester and distinguish and examine different periods within this timeline. Contextuality also places emphasis on aims of students with regards to a course, which can help prioritize ITS features in different contexts. Utilizing a service design-based approach adds to the literature of ITS evaluation by redirecting attention to the learners and help achieve ITS goals. Consequently, conducting ITS evaluations focused on the learners in the context of learning can inform ITS designs that fit education contexts better and thus better support learners' needs, even when a human tutor is not available.

References

Abu Naser, S. S. (2012). Predicting Learners Performance Using Artificial Neural Networks in Linear Programming Intelligent Tutoring System. *International Journal of Artificial Intelligence & Applications*, *3*(2), 65–73. https://doi.org/10.5121/ijaia.2012.3206

Allanwood, G., & Beare, P. (2019). *User Experience Design: A Practical Introduction*. Bloomsbury Publishing.

Andone, I., & Sireteanu, N.-A. (2008, April). Heuristic Evaluation of Web-Based Intelligent Tutoring Systems. *The 4th International Scientific Conference ELSE, Bucharest*.

Bangor, A., Kortum, P., & Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, *4*, 114–123.

Boren, M. T., & Ramey, J. (2000). Thinking aloud: Reconciling theory and practice. *IEEE Transactions on Professional Communication*, *43*(3), 261–278. https://doi.org/10.1109/47.867942

Cascaes Cardoso, M. (2017). The Onboarding Effect: Leveraging User Engagement And Retention In Crowdsourcing Platforms. *2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 263–367. https://doi.org/10.1145/3027063.3027128

Catalanotto, D. (2018). *A Tiny History of Service Design* . Blurb.

Cetintas, S., Si, L., Xin, Y. P., & Hord, C. (2009, July). Predicting Correctness of Problem Solving from Low-level Log Data in Intelligent Tutoring Systems. *International Conference on Educational Data Mining (EDM)*.

Chrysafiadi, K., Virvou, M., Tsihrintzis, G. A., & Hatzilygeroudis, I. (2022). Evaluating the user's experience, adaptivity and learning outcomes of a fuzzy-based intelligent tutoring system for computer programming for academic students in Greece. *Education and Information Technologies*. https://doi.org/10.1007/s10639-022-11444-3

Chughtai, R., Zhang, S., & Craig, S. D. (2016). Usability evaluation of intelligent tutoring system. *Http://Dx.Doi.Org/10.1177/1541931215591076*, *2015-January*, 367–371. https://doi.org/10.1177/1541931215591076

Dasgupta, S., Granger, M., & McGarry, N. (2002). User acceptance of e-collaboration technology: An extension of the technology acceptance model. *Group Decision and Negotiation*, *11*(2), 87–100. https://doi.org/10.1023/A:1015221710638/METRICS

Digital Design Agency. (2022). *Service Blueprint Template*. https://miro.com/miroverse/service-blueprint-template/

Erümit, A. K., Çetin, İ., Kokoç, M., Kösa, T., Nabiyev, V., & Aygün, E. S. (2019). Designing a Usibility Assessment Process for Adaptive Intelligent Tutoring Systems: A Case Study. *Turkish Online Journal of Qualitative Inquiry*, 141–179. https://doi.org/10.17569/tojqi.506439

Fluxicon BV. (2023). *Fluxion Disco* (3.3.7). https://fluxicon.com/disco/

Gibbons, S. (2017, August 27). *Service Blueprints: Definition*. https://www.nngroup.com/articles/service-blueprints-definition/

Granić, A. (2008). Experience with usability evaluation of e-learning systems. *Universal Access in the Information Society*, *7*(4), 209–221. https://doi.org/10.1007/s10209-008-0118-z

Granić, A., Glavinić, V., & Kluev, V. (2002). An Approach to Usability Evaluation of an Intelligent Tutoring System. *Advances in Multimedia, Video and Signal Processing Systems*, 77–84. https://www.researchgate.net/publication/268199851

Greer, J., & Mark, M. (2016). Evaluation Methods for Intelligent Tutoring Systems Revisited. *International Journal of Artificial Intelligence in Education*, *26*(1), 387–392. https://doi.org/10.1007/S40593-015-0043-2/METRICS

Guay, F., Vallerand, R. J., & Blanchard, C. (2000). On the assessment of situational intrinsic and extrinsic motivation: The Situational Motivation Scale (SIMS). *Motivation and Emotion*, *24*(3), 175–213. https://doi.org/10.1023/A:1005614228250/METRICS

Guest, G., MacQueen, K. M., & Namey, E. E. (2011). *Applied Thematic Analysis*. SAGE Publications.

Guo, L., Wang, D., Gu, F., Li, Y., Wang, Y., & Zhou, R. (2021). Evolution and trends in intelligent tutoring systems research: a multidisciplinary and scientometric view. In *Asia*

Pacific Education Review (Vol. 22, Issue 3, pp. 441–461). Springer Science and Business Media B.V. https://doi.org/10.1007/s12564-021-09697-7

Haridas, M., Gutjahr, G., Raman, R., Ramaraju, R., & Nedungadi, P. (2020). Predicting school performance and early risk of failure from an intelligent tutoring system. *Education and Information Technologies*, *25*(5), 3995–4013. https://doi.org/10.1007/S10639-020-10144-0/FIGURES/7

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*, *52*(C), 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

Hassan, H. M., & Galal-Edeen, G. H. (2017). From usability to user experience. *Conference on Intelligent Informatics and BioMedical Sciences*, 216–222.

Hassenzahl, M. (2007). The hedonic/pragmatic model of user experience. *Towards a UX Manifesto*, 16–20.

Instructure. (2023). *Canvas LMS*.

Ivankova, N., & Wingo, N. (2018). Applying Mixed Methods in Action Research: Methodological Potentials and Advantages. *American Behavioral Scientist*, *62*(7), 978–997. https://doi.org/10.1177/0002764218772673/ASSET/IMAGES/LARGE/10.1177_0002764218772673-FIG2.JPEG

Janning, R., Schatten, C., & Schmidt-Thieme, L. (2016). Perceived Task-Difficulty Recognition from Log-file Information for the Use in Adaptive Intelligent Tutoring Systems. *International Journal of Artificial Intelligence in Education*, *26*(3), 855–876. https://doi.org/10.1007/s40593-016-0097-9

Kabudi, T., Pappas, I., & Olsen, D. H. (2021). AI-enabled adaptive learning systems: A systematic mapping of the literature. *Computers and Education: Artificial Intelligence*, *2*. https://doi.org/10.1016/j.caeai.2021.100017

Kuhn, S., & Muller, M. J. (1993). Participatory design. *Communications of the ACM*.

Lawless, S., O'connor, A., & Mulwa, C. (2010). *A Proposal for the Evaluation of Adaptive Personalized Information Retrieval*.

LimeSurvey GmbH. (2022). *LimeSurvey* (5.3). https://www.limesurvey.org/

Lindgaard, G., Fernandes, G., Dudek, C., Brown, J., Lindgaard, G., Fernandes, G., Dudekx, C., & Brown, J. (2011). Attention web designers: You have 50 milliseconds to make a good first impression! *Behaviour & Information Technology*, *25*(2), 115–126. https://doi.org/10.1080/01449290500330448

Lynch, T., & Ghergulescu, I. (2016). An Evaluation Framework for Adaptive and Intelligent Tutoring Systems. *In E-Learn: World Conference on e-Learning in Corporate, Government, Healthcare, and Higher Education*, 1385–1390.

Mark, M. A., & Greer, J. E. (1993). Evaluation Methodologies for Intelligent Tutoring Systems. *Journal of Artificial Intelligence in Education*, *4*, 129–153.

Mertens, D. M. (2017). Mixed Methods in Evaluation: History and Progress. In *Mixed Methods Design in Evaluation* (1st ed., pp. 1–30).

Microsoft. (2022). *Microsoft Teams* (1.5.00.33362). Microsoft. https://www.microsoft.com/en-us/microsoft-teams/group-chat-software

Miller, J. R. (1988). The Role of Human-Computer Interaction in Intelligent Tutoring Systems. In *Foundations of Intelligent Tutoring Systems* (1st ed.). Psychology Press.

Miro. (2023). *Miro*. https://miro.com/

Mousavinasab, E., Zarifsanaiey, N., R. Niakan Kalhori, S., Rakhshan, M., Keikha, L., & Ghazi Saeedi, M. (2021). Intelligent tutoring systems: a systematic review of

characteristics, applications, and evaluation methods. In *Interactive Learning Environments* (Vol. 29, Issue 1, pp. 142–163). Routledge. https://doi.org/10.1080/10494820.2018.1558257

Mulwa, C., Lawless, S., Sharp, M., & Wade, V. (2011). A web-based framework for user-centred evaluation of end-user experience in adaptive and personalized e-Learning systems. *Proceedings - 2011 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2011*, *3*, 351–356. https://doi.org/10.1109/WI-IAT.2011.203

Nielsen, J. (1993). *Usability Engineering*. Academic Press.

Norman, D. (2013). The Design of Everyday Things. In *academia.edu* (17th ed.).

Norman, D., & Nielsen, J. (n.d.). *The Definition of User Experience (UX)*. Retrieved January 31, 2023, from https://www.nngroup.com/articles/definition-user-experience/

Nwana, H. S. (1990). Intelligent Tutoring Systems: an overview. *Artificial Intelligence Review*, *4*, 251–277.

Obendorf, H., & Finck, M. (2008). Scenario-based usability engineering techniques in agile development processes. *Conference on Human Factors in Computing Systems - Proceedings*, 2159–2166. https://doi.org/10.1145/1358628.1358649

Pian, Y., Lu, Y., Huang, Y., & Bittencourt, I. I. (2020). A Gamified Solution to the Cold-Start Problem of Intelligent Tutoring System. *Artificial Intelligence in Education* , 378–381. http://www.springer.com/series/1244

Radack, S. (2009). *The system development life cycle(SDLC)*. https://csrc.nist.gov/csrc/media/publications/shared/documents/itl-bulletin/itlbul2009-04.pdf

Radford, A., Kim, J. W., Xu, T., Brockman, G., Mcleavey, C., & Sutskever, I. (2022). *Robust Speech Recognition via Large-Scale Weak Supervision [Manuscript submitted for publication]*. https://github.com/openai/

Reimann, R., Cooper, A., Cronin, D., & Noessel, C. (2014). *About Face: The Essentials of Interaction Design* (4th ed.). John Wiley & Sons.

Renz, J., Staubitz, T., Pollak, J., & Meinel, C. (2014). IMPROVING THE ONBOARDING USER EXPERIENCE IN MOOCS. *EDULEARN14 Proceedings*, 3931–3941. http://library.iated.org/view/FELIP2014MON

Shute, V. J., & Regian, W. (1993). Principles for Evaluating Intelligent Tutoring Systems. *Jl. of Artificial Intelligence and Education*, 245–271.

Siemer, J., & Angelides, M. C. (1998). A comprehensive method for the evaluation of complete intelligent tutoring systems. In *Decision Support Systems* (Vol. 22).

Stickdorn, M., & Schneider, J. (2010). *This is Service Design Thinking: Basics, Tools, Cases* (1st ed.). John Wiley & Sons.

Still, B., & Crane, K. (2017a). Introduction to User-Centered Design. In *Fundamentals of User-Centered Design* (1st ed., pp. 1–17). CRC Press.

Still, B., & Crane, K. (2017b). UCD Principles. In *Fundamentals of User-Centered Design* (pp. 43–65). CRC Press.

Terres, P., Klumpe, J., Jung, D., & Koch, O. (2019, May 15). DIGITAL NUDGES FOR USER ONBOARDING: TURNING VISITORS INTO USERS. *ECIS 2019*.

The Coq Development Team. (2022). *The Coq Proof Assistant* (8.16.1).

The R Foundation. (2023). *R*. https://www.r-project.org/

Thurmond, V. A. (2001). The point of triangulation. *Journal of Nursing Scholarship*, *33*(3), 253–258. https://doi.org/10.1111/J.1547-5069.2001.00253.X

Virvou, M., & Tsiriga, V. (2000). International Forum of Educational Technology & Society Involving Effectively Teachers and Students in the Life Cycle of an Intelligent Tutoring System. *Source: Journal of Educational Technology & Society*, *3*(3), 511–521. https://doi.org/10.2307/jeductechsoci.3.3.511

Waterproof Development Team. (2022). *Waterproof* (6.1). Eindhoven University of Technology. https://github.com/impermeable/waterproof

Wemmenhove, J., Beurskens, T., McCarren, S., Moraal, J., Tuin, D., & Portegies, J. (2022). *Waterproof: educational software for learning how to write mathematical proofs*. https://doi.org/10.48550/arxiv.2211.13513

Wolfe, K. (2020). Service design in higher education: a literature review. *Https://Doi.Org/10.1080/13603108.2020.1792573*, 121–125. https://doi.org/10.1080/13603108.2020.1792573

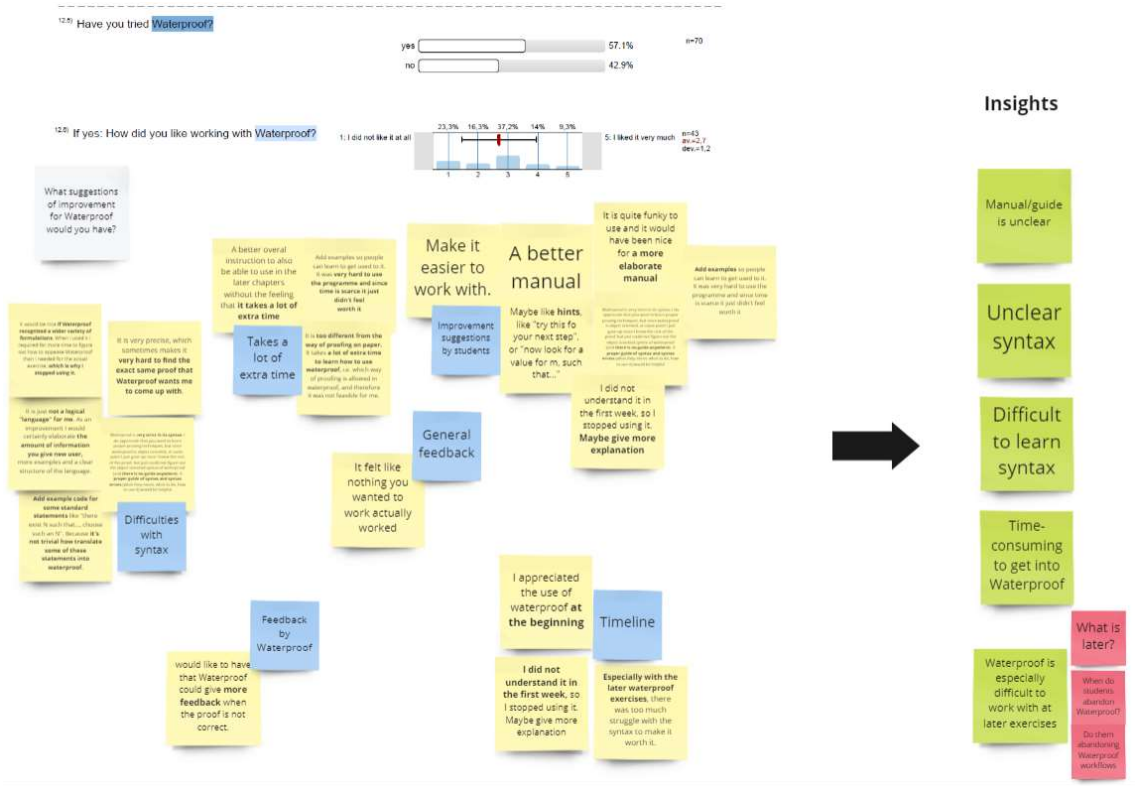
Woolf, B. P. (2010). Evaluation. In *Building Intelligent Interactive Tutors* (pp. 183–220).

Appendices

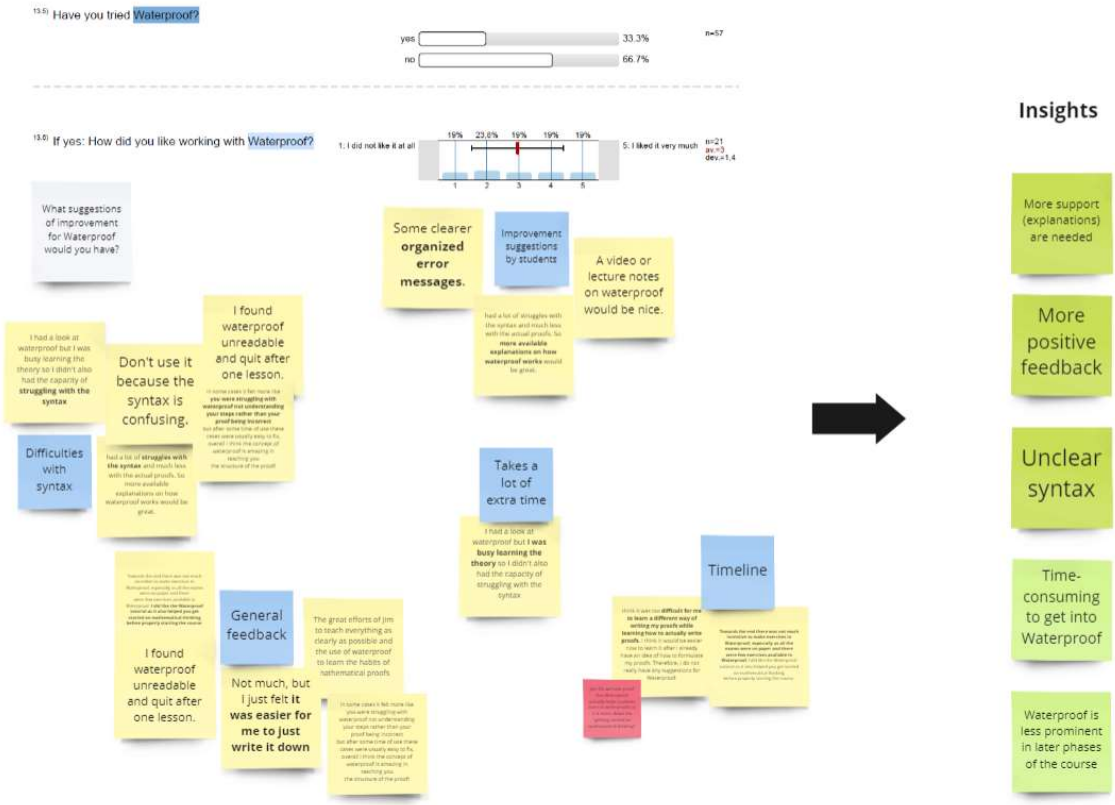
Appendix A

Outcomes of the initial conversations with the Waterproof development team.

Thematic analysis of Waterproof-related questions in the Analysis I. course assessments 2020/21



Thematic analysis of Waterproof-related questions in the Analysis I. course assessments 2020/21



Appendix B

Service blueprinting materials.

Blueprinting script.

Goal(s):

- Draw up a low fidelity service Blueprint for Waterproof onboarding
- Identify bumps and pain points in the process
- Brainstorm on points of improvement

Agenda

1) **Introduction – 10 min**
2) **Part A – The framework + user actions – 20 min**
3) **Part B – Blueprinting – 40 min**
4) **Part C - Taking a step back – broader picture – 30 min**
5) **Wrap up and feedback moment – 5 min (if we have time)**

Script

1) **Introduction – 10 min**
    a. Introducing Dorina and Rianne + Thank you for coming + How is everyone doing?
    b. Introduction round
        i. Name
        ii. how they are connected to Waterproof
        iii. any fun plans for the weekend?
    c. **Aim of this session:** draw up a higher level, preliminary service blueprint **of the service Waterproof provides (as is, i.e. current way)** – later there will be room for discussing points of improvement
        i. What is service design? – Why do we examine Waterproof as a service?
        To get a broader picture and help improve the service by understanding what each stakeholder group (also represented here) needs for the service for it to work smoothly and provide a good user experience
        ii. What is a service blueprint and why it is good to draw up together?
        Helps us understand how the service can be best delivered, if changes need to be made, and whether the people involved understand their respective parts
    d. About this workshop
        i. The workshop will be interactive.
        ii. We will work both all, together and in groups of 3.
        iii. Every group member will contribute to drawing up the blueprint.
        iv. We won't map every single interaction in the scenario — just the most important ones (we're telling a story, not making a comprehensive list).
        v. There may be knowledge gaps and open questions at the end of the workshop, which is okay.
        vi. We won't leave the workshop with a polished or complete blueprint.
    e. **"Rules"**
        i. honor each other's time
        ii. honor time box times
        iii. if you need a break, please let me know
        iv. trust us with the process
        v. be open for FRIENDLY nudging (mostly because of the time constraint)
        vi. if you have any questions, please feel free to interrupt me

Before we get started, I just want to check, how strict that 1.5 hours is, I know that … needs to leave, but I am also asking the others.

f. Agenda (on the big brown paper on the wall)

2) **Part A - The framework + user actions – 20 min**

a. Service scenario: read out loud

Imagine the following: a student enrolled in the course Analysis 1 read the course page on Canvas and read about Waterproof. They are enthusiastic and want to try the tool. Please think about how the student will proceed into getting to know Waterproof. We are especially interested in the steps until the student starts working on their first exercise sheet in Waterproof.

b. Explain the blueprinting framework (on the big brown paper on the wall) – 5 minutes
   i. "rows" – separating lines: line of interaction, line of visibility, line of internal interaction; - student actions, frontstage actions, backstage actions, support processed
   ii. "columns" – defined by the process

Key elements

- Customer actions
  - Basically a customer journey map
  - Steps, choices, activities, and interactions that customer performs while interacting with a service to reach a particular goal.
- Frontstage actions
  - Actions that occur directly in view of the customer
  - Can be human-human (interaction with employee) or human-computer (interaction with self service technology)
  - there is not always a parallel frontstage action for every customer touchpoint.
  - If customer interacts with the service: a moment of truth happens: customers judge your quality and make decisions regarding future purchases
- Backstage actions
  - Steps and activities that occur behind the scenes to support onstage happenings.
  - Performed by backstage employee OR frontstage employee
- Processes
  - Internal(!) steps and interactions that support the employees in delivering the service

Lines:

- Line of interaction: direct interactions between the customer and the organization
- Line of visibility: separates all service activities that are visible to the customer from those that are not visible (frontstage: above, backstage: below)
- line of internal interaction: separates contact employees from those who do not directly support interactions with customers/users

c. First, all together, please write up the student actions (end user actions) on a blue sticky note – 15 min

3) **Part B – Blueprinting – 40 min**

a. Make groups and divide user actions ("columns"). – **5 min**
b. Each group works on multiple (2-3) user actions separately. – **35 min**

Tutorial: what is that? It is a file
-be more clear on support processes

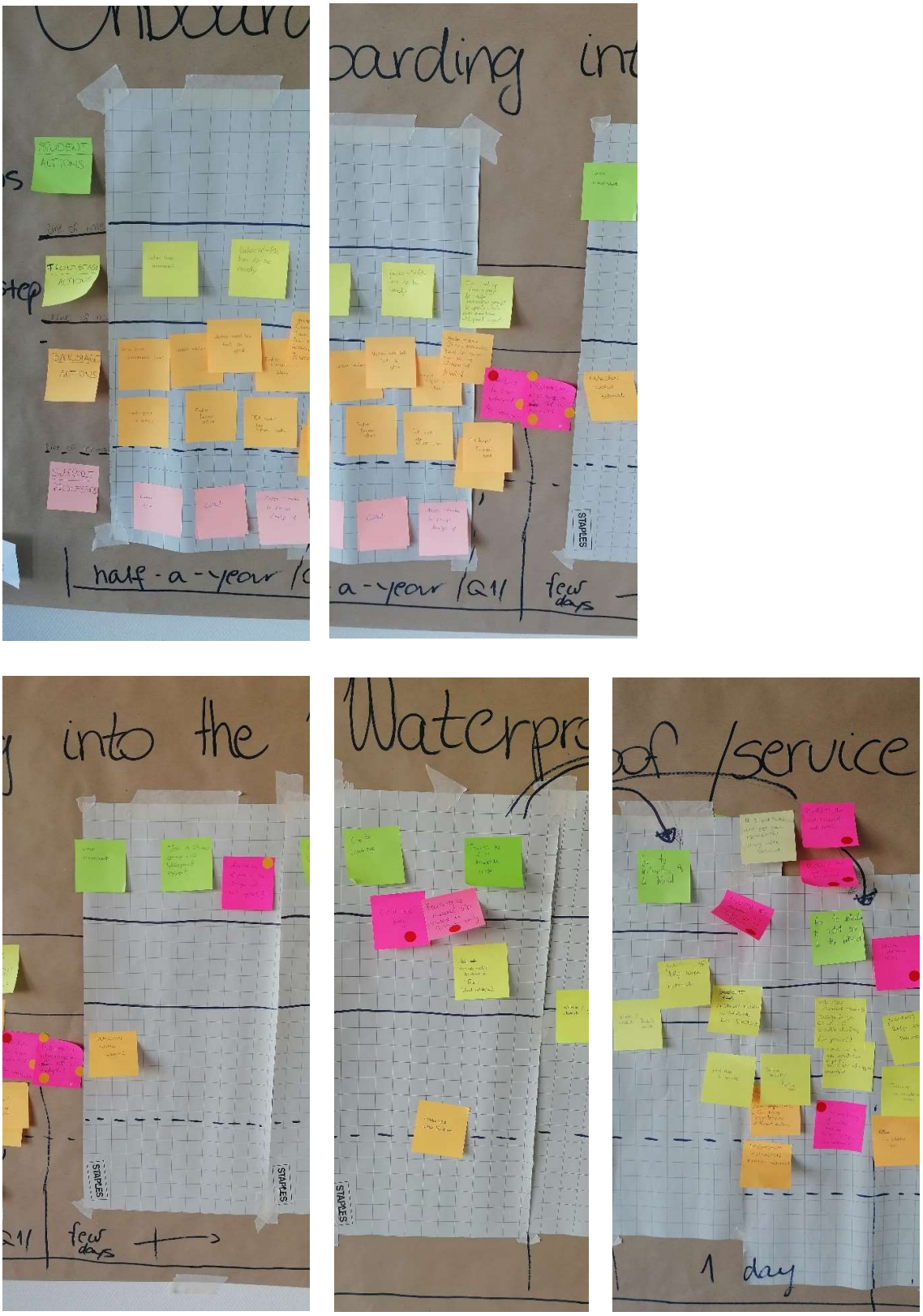
4) **Part C – Taking a step back – broader picture – 30 min**

a. Assemble the blueprint and take a look together – **5 min**
b. Add arrows, timeline,– **10 min**

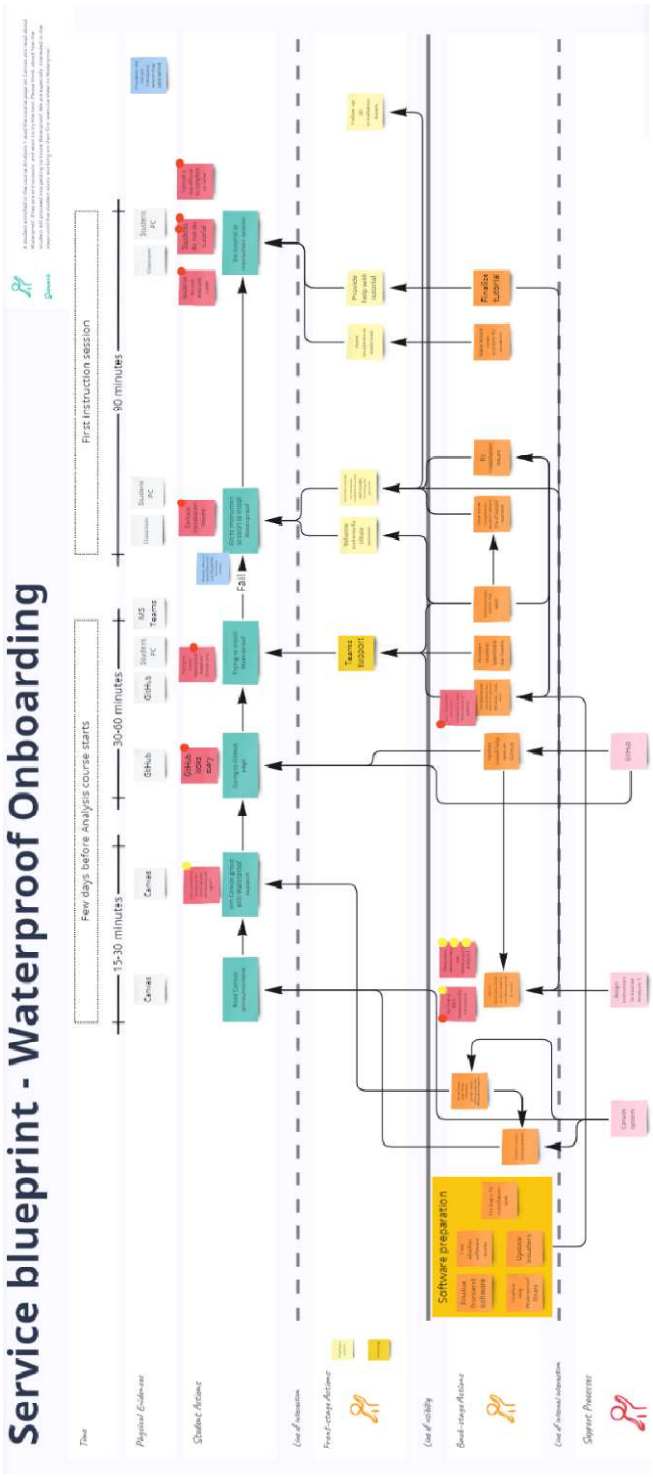
a. Questions: - **15 min**
    i. when is the journey successful? – **3 min**
    ii. What are the bumps, pain points – **5 min**
        1. vote on pain points (dot voting) – **2 min**
    iii. what could we do to make the process smoother? – **5 min**

5) **Wrap up and feedback moment – 5 min**
6) **Next steps**
    a. "Clean up" the blueprint and put it in Miro
    b. Share with participants and ask them for feedback (comment) – what kind of feedback? – missing, incorrect, misunderstood

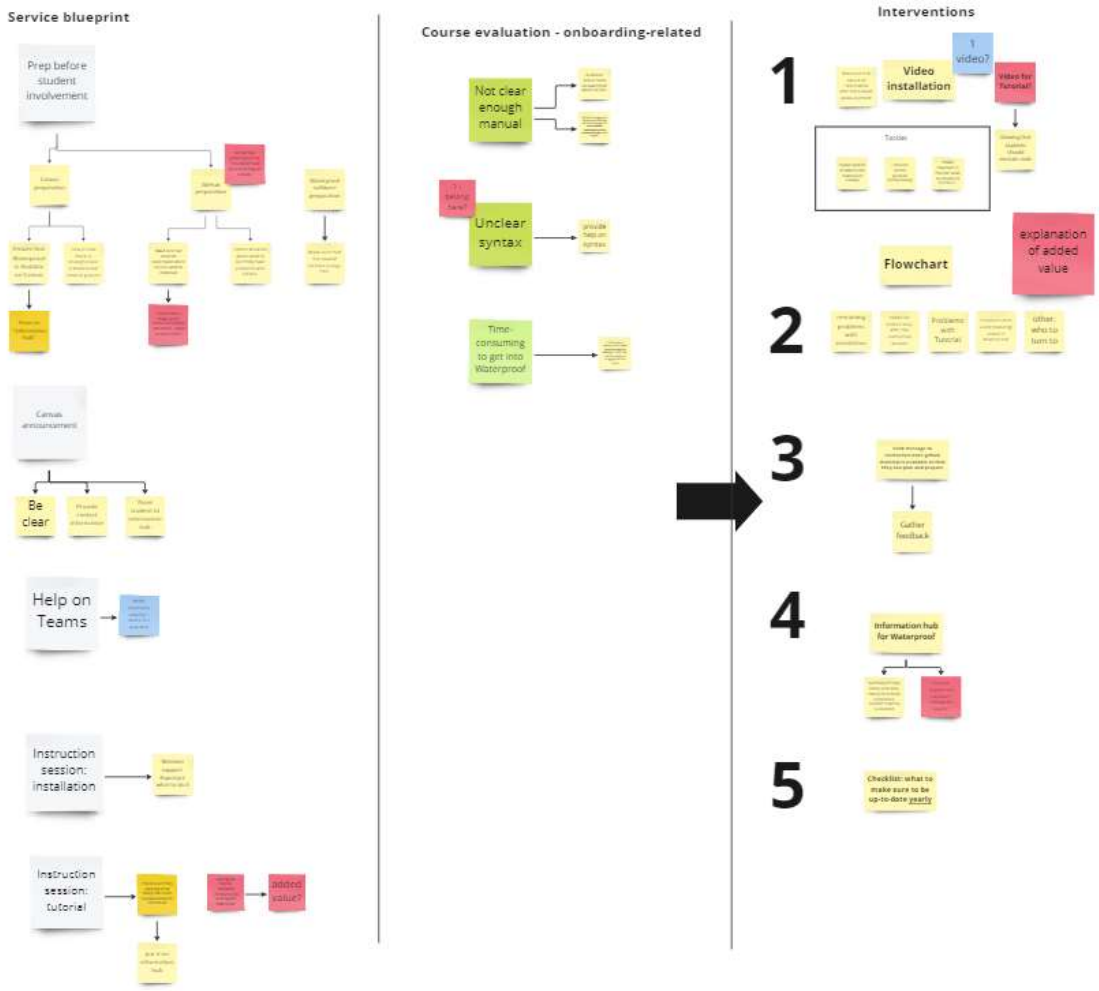
Outcomes of the blueprinting session on paper.



The digitalized and organized service blueprint of Waterproof user onboarding.

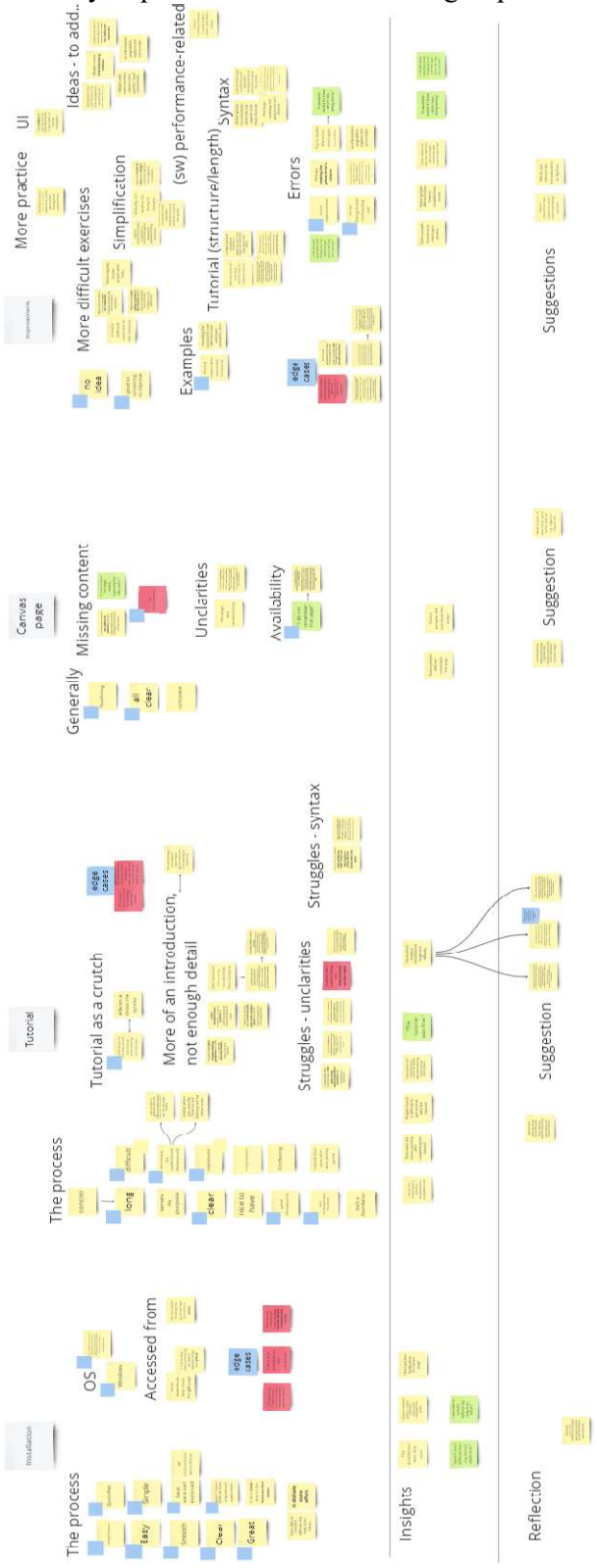


Working out suggested user onboarding improvements.

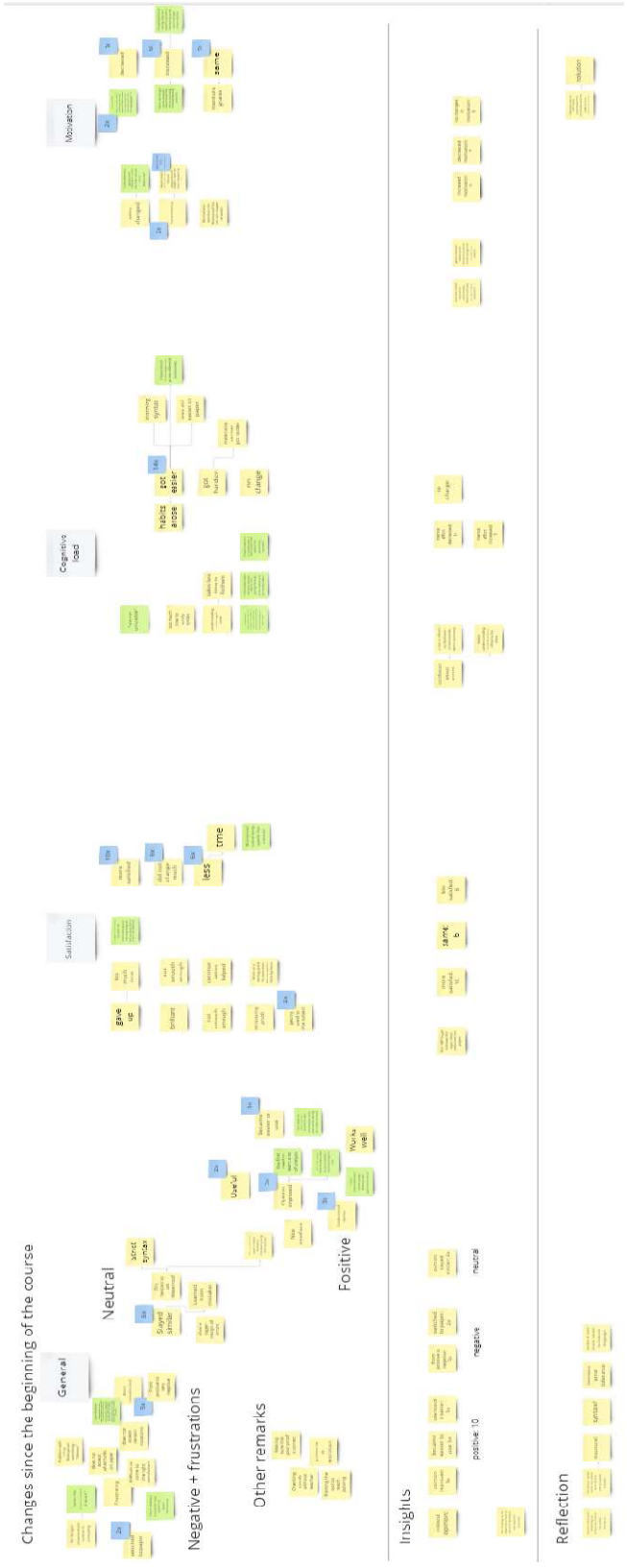


Appendix C

Qualitative analysis of Survey 1 questions about onboarding implementations



Qualitative analysis of Survey 2 questions about changes in user-centered constructs over the time using Waterproof



Appendix D

User test/interview materials.

Interview script.

Intro
Hi! Welcome to this in-person session! Thank you very much for being here! The goal of this 45-minute session is for me to learn even more about how you use Waterproof. This time, as opposed to the survey, the focus will be on your direct interaction with Waterproof.
Do you have a question about this? 😊
Let's quickly go through what is going to happen today 😊
First, I am going to ask you to complete a list of tasks with Waterproof and then, I will ask you to answer some questions. I will make a screen-and-voice recording of this session in Microsoft Teams **(show).** If you have any questions during this session, please feel free to ask me; however, I cannot always give you an answer, as I am curious about how YOU would go about solving the tasks.
In this session it is not YOU are being tested but Waterproof, so you cannot do anything wrong, there are no wrong answers. I am not here to judge you at all, it is completely fine to make mistakes in the proving process. We are super happy with all the negative critique and faults you find as it helps us further improve Waterproof!
If you want to have a break, please just say so!
Do you have any questions before I tell you more about the tasks?
Tasks
I'll ask you to use this laptop to work on two specific tasks with Waterproof. For this, I will ask you to work on an exercise sheet from Waterproof you are already familiar with about Proofs in Analysis. I will ask you not to use anything else but Waterproof to work on this task.
While working on the tasks, I will ask you to think aloud. This means saying out loud what you are doing in Waterproof. This might be a bit unusual at first, so I might nudge you to continue if you stop mid-task if you are silent for a while.
We will first do a quick practice task for the thinking aloud method so that you get into it.
***Here are your tasks.**

PRACTICE TASK (for the thinking aloud methodology): Please change the theme of Waterproof to light and the zoom to 100% while you are thinking aloud.

TASK1: Please solve Exercise 3.11.1.

TASK2: Please execute the sentence written in the text field of Exercise 3.11.2 and execute the sentence. What do you think happened? How would you proceed? These were all the tasks! 😊 Shall we proceed to the questions?

Questions + potential follow-ups

- How do you think this session went? How did you feel?
- Can you tell me how you usually go about working on an exercise sheet?
    - Solving errors?
    - When do you give up? When do you stop?
- How does your homework group usually work?
    - Do you work in sessions or in one go?
    - Do you have roles in your group?

- What Operating System do you usually use?
- Do you usually use your mouse or keyboard for navigation?
- Follow-up questions based on the tasks.

These were all my questions for now 😊
Do YOU have any questions?

Debriefing
Thank you so much for participating in this session. You helped this project immensely!
This time as opposed to the survey earlier, I observed your interaction with Waterproof, e.g. did you use any hotkeys, what caught your attention, so more contextual information.
If in the future you have any questions, please shoot me an email.
Do you have any questions left?
If not, there is only one thing left to do: please fill in your bank account and sign this form for payment. 😊 (Normally, I would transfer you the money, but you can also send a Tikkie, if you prefer that.) Thank you!

Interviewer's guide

Interviewer's guide

TASK1

Input: Type / Tactics

Tactics:
Used? Y / N
Accessed from: text field menu / tactics menu
Help tactic used? Y / N
Why?

Search:
Used? Y / N
Accessed from: text field menu / tactics menu
Why?

Hints:
Used? Y / N

Symbols:
Input: autocomplete / from symbols menu

Tutorial:
Used? Y / N

Hotkeys used? Y / N

Bumps:

Questions asked:

Sentence execution (at which points):

TASK2:

Tactics:

Used? Y / N
Accessed from: text field menu / tactics menu
Help tactic used? Y / N
Why?

Search:
Used? Y / N
Accessed from: text field menu / tactics menu
Why?

Hints:
Used? Y / N

Tutorial:
Used? Y / N

Hotkeys used? Y / N

Bumps:

Questions asked:

Questions

- How do you think this session went? How did you feel?
- Can you tell me how you usually go about working on an exercise sheet?
    - Solving errors?
    - When do you give up? When do you stop?
- How does your homework group usually work?
    - Do you work in sessions or in one go?
- Do you have roles in your group?
- What Operating System do you usually use?
- Do you usually use your mouse or keyboard for navigation?
- FOLLOW UPS?

Data analysis Part 1.

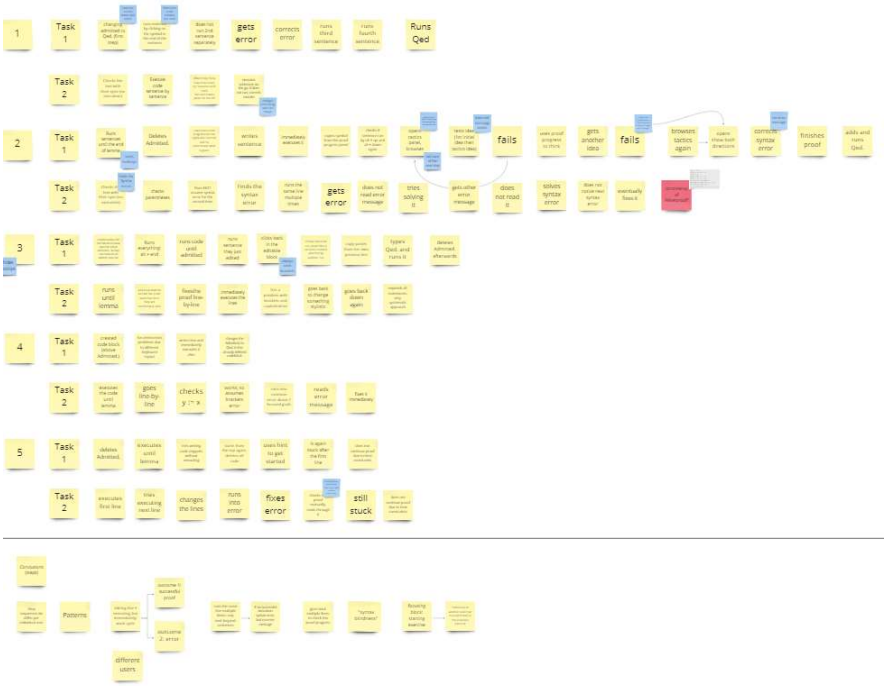
Use of functionalities and Interaction data.

	Functionalities						Interaction			Study-specific
	Tactics	Search	Symbols library	Hints	Tutorial	Other	Input	Hotkeys	Execution	Usually used OS
1	NO	NO	hotkeys: \leq	NO	NO	NO	type	symbols	clicking on the execute symbol at the end of the sentence trying the commands menu on the left	Windows
2	opening tactics panel browsing it (no copying from it)	NO	copies symbol from the text shown in the "proof progress" tab	NO	NO	NO	type copy-paste	execution	alt + down hotkey	Mac
3	NO	NO	hotkeys: \R; \leq; \and	NO	NO	NO	type copy-paste	execution symbols	alt + end hotkey clicking on the execute symbol at the end of the sentence alt + down hotkey	Windows
4	NO	NO	hotkeys: \reals	NO	NO	NO	type	adding code block (alt + c while hovering the blue line) execution symbols	alt + down hotkey	Mac
5	NO	NO	opening from the code block menu: inserts element; reals, less than equal, implication arrow	YES Task 1 Task 2	NO	NO	type symbols from symbol menu	execution	commands menu on the right alt + down hotkey	Mac AND Windows

Errors.

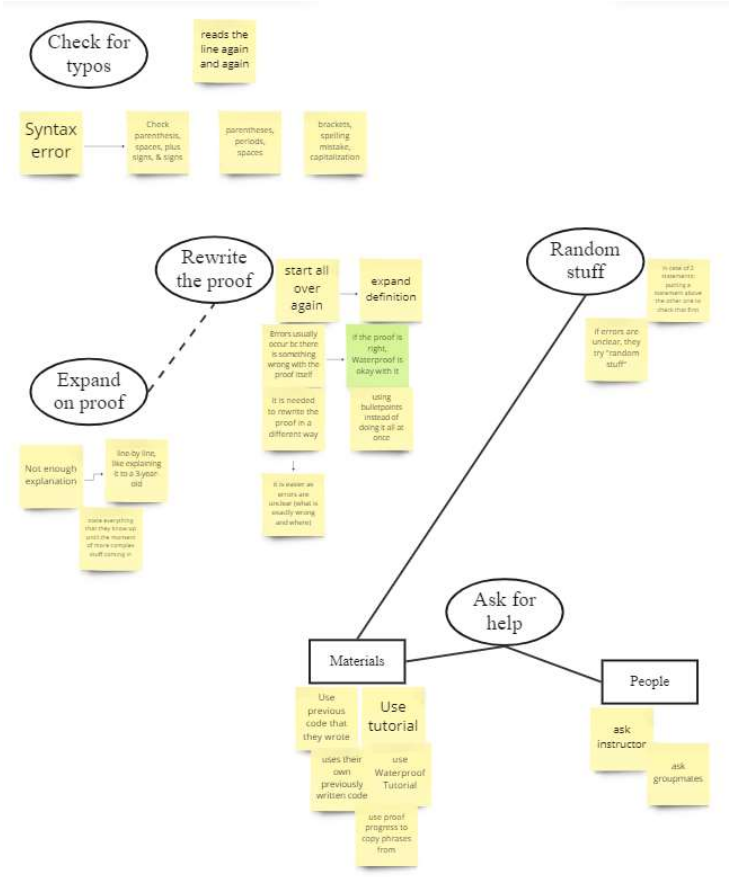
	Errors	Def: severe: user cannot recover from it, medium: user can recover from it, but not immediately, mild: user can recover from it immediately					
	Error	Severity (1 - severe; 2 - medium; 3 - mild)	Total # occurrence	Occurrence (#participants)	Task	Handling	Type
1	Uncaught Ltac exception: TakeError	3	1	1	Task 1	Change take to assume	Other error
2	Expected a single focused goal but 2 goals are focused.	3 (3); 2 (1)	5	4	Task 2	Added missing signs	Other error
3	Uncaught Ltac exception: BothDirectionsError	2	2	1	Task 1	Browsed tactics and cha	Other error
4	[Focus Wrong bullet-: current bullet- is not finished	3	1	1	Task 2	Finishes the statement	Other error
5	Expected a single focus goal but 0 goals are focused	3	1	1	Task 2	Finishes the statement	Other error
6	Uncaught Ltac 2 exception: TakeError	3	1	1	Task 2	Deletes sentence	Other error

Preliminary process mapping.

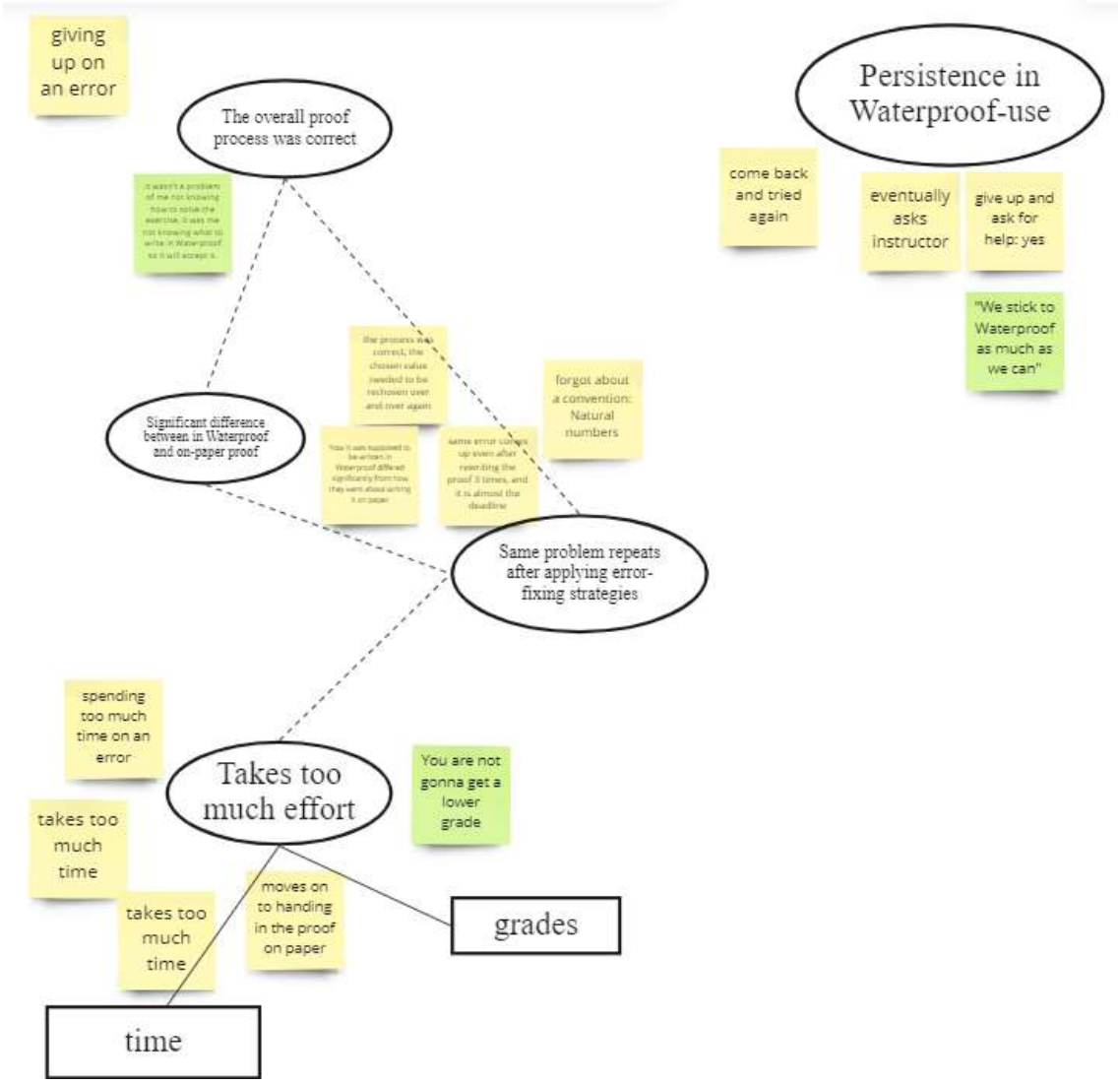


Data analysis Part 2.

Error fixing strategies



"Giving up" circumstances



Appendix E

Logs collected from Waterproof.

All messages have "type": name of type, "sinceBoot": millis since boot

* = not in release v0.6-RC1
! = behind setting

Type | trigger | properties
boot | when log is started up | time: startup time stamp
heartbeat | to show the application is still active |
startup | to show the internal app has loaded |
closing | when the application is being closed | running: whether the application is still running so we ask if there is unsaved progress
navigation | when the page of the app changes | to: the name of the page we are going to, location: a file if directly opening a file

* open-new-tab | when already in edit mode and adding a new tab | file: the file url of the new tab or null if empty, tabIndex: the index of the new tab
* switch-tab | when switching between already open tabs | file: the file url of the tab or null if empty, tabIndex: the tab index of the tab
loaded-file | when a notebook is loaded | file, tabIndex, isExercise: whether the file is an exercise

focusing-block | when clicking on a block | file, tabIndex, blockIndex: the index of the block in the notebook, exerciseIndex: before or in which exercise is the block being focused
inserting-block | when a new block is inserted | file, tabIndex, blockIndex: the index of the new block, blockType: the type of the new block, insertingBlockInBlock: whether this block was inserted within another block
removing-blocks | when blocks are removed | file, tabIndex, blocksRemoved: array of block indices which are being removed

side-window-change | when the side window is changed | sideWindowName: the name of the side window now opened or null if none, openedSideWindow: whether a window is open, should be exactly true if sideWindowName is not null

coq-exec-to | when executing to | file, tabIndex, targetIndex: the target index to which to execute
coq-exec-next | when (attempting to) execute the next sentence | file, tabIndex
coq-exec-prev | when (attempting to) execute the last sentence | file, tabIndex
coq-exec-all | when (attempting to) execute all sentences | file, tabIndex
coq-exec-to-cursor | executing to a where the cursor is in the code | file, tabIndex, targetIndex: to where to execute
coq-search | when searching via serapi | file, tabIndex: the active file and tab since commands are "executed" there, searchQuery: the query, fromExample: whether the search is from an example

! coq-success-sentence | when we have successfully executed a sentence | file, tabIndex, exerciseIndex, coqID: the coq id of the sentence, text: the source text of the sentence, blockIndex: the index of the block containing (the end of) the sentence, indexInBlock: the end index of the sentence in the block
! coq-execute-error | when we hit an execute error | file, tabIndex, exerciseIndex, coqID: the coq id of the sentence, error: the message of the error, beginIndex: the begin index in the

source of the error, endIndex: the end index of the error in the source, blockIndex: the index of the block where the error occurred, indexInBlock: the index within the block of the start of the error
coq-add-error-shown | when an add error was hit and after some time shown to the user | file, error: the message of the error ! DOES NOT HAVE TABINDEX !
coq-next-beyond-sentence | when the user attempt to execute the next sentence but their is none | file, tabIndex, executedIndex: to where we have already executed, addErrorIndex: the index of any active add errors or -1 if none

Variable types:
For the meaning see above
type: string, name of type
time: string, a timestamp
sinceBoot: number, milliseconds since boot
running: boolean
to: string, internal page name one of "home", "edit"
location: null | string, a file path
file: null | string, a file path of the respective file
tabIndex: number, index of the tab, so the number starting at 0
isExercise: boolean
blockIndex: number, the index of the block in the notebook
exerciseIndex: number, at/before which exercise (input block) the block is. So before the first input block, gives 0. After that but before the second input block gives 1, etc.
blockType: string, the type of the block one of "code", "text", "input", "hint"
insertingBlockInBlock: boolean
blocksRemoved: Array<Number>, indices of all the blocks removed
sideWindowName: null | string, the name of the sideWindow one of "Mathematical Symbols", "Common Tactics", "Commands", "Search Results"
openedSideWindow: boolean, should be true if sideWindowName is not null
targetIndex: number, a target index in the coq text, the exact character, the is not related to the blockIndex
searchQuery: string
fromExample: boolean
coqID: number, the serapi/coq id of the sentence
indexInBlock: number, a character index within the block
beginIndex: number, a character index within the coq text
endIndex: number, a character index within the coq text
error: string
executedIndex: number, a character index within the coq text
addErrorIndex: -1 | number, a character index within the coq text

R code for data cleaning and activity classification

```
file2 <- read_csv("data_out/data_clean.csv")
       # indicate sessions (closing switching to boot)
file_ext <- file2 %>%
  filter(type != "heartbeat") %>%
  group_by(filename) %>%
  mutate(session_start = ifelse((lag(closing type) == "" & type == "boot" )|
                                  row_number() == 1,
                                1, 0),
         session_no = cumsum(session_start),

         # indicate exercisesheet
         exercisesheet_start = ifelse(!is.na(file) & !grepl("Tutorial", file) &
                                        (lag(file) != file | is.na(lag(file))),
                                       1, 0),

         # indicate working in tutorial or elsewhere
         tutorial = ifelse(grepl("Tutorial", file), 1,
                    ifelse(exercisesheet_start == 1, 2, NA)),
         workgroup_time = (real_time > "2022-11-14 13:30" &
                           real_time < "2022-11-14 15:30") |
                           (real_time > "2022-11-16 08:45" &
                            real_time < "2022-11-16 10:45") |
                             (real_time > "2022-11-21 13:30" &
                              real_time < "2022-11-21 15:30") |
                               (real_time > "2022-11-23 08:45" &
                                real_time < "2022-11-23 10:45") |
                                 (real_time > "2022-11-28 13:30" &
                                  real_time < "2022-11-28 15:30") |
                                   (real_time > "2022-11-30 08:45" &
                                    real_time < "2022-11-30 10:45") |
                                     (real_time > "2022-12-05 13:30" &
                                      real_time < "2022-12-05 15:30") |
                                       (real_time > "2022-12-07 08:45" &
                                        real_time < "2022-12-07 10:45") |
                                         (real_time > "2022-12-12 13:30" &
                                          real_time < "2022-12-12 15:30") |
                                           (real_time > "2022-12-14 08:45" &
                                            real_time < "2022-12-14 10:45") |
                                             (real_time > "2022-12-19 13:30" &
                                              real_time < "2022-12-19 15:30") |
                                               (real_time > "2022-12-21 08:45" &
                                                real_time < "2022-12-21 10:45") |
                                                 (real_time > "2023-01-09 13:30" &
                                                  real_time < "2023-01-09 15:30") |
                                                   (real_time > "2023-01-11 08:45" &
                                                    real_time < "2023-01-11 10:45") |
                                                     (real_time > "2023-01-16 13:30" &
                                                      real_time < "2023-01-11 15:30")
       )   %>%
  fill(tutorial) %>%
  mutate(
    tutorial = ifelse(tutorial == 2, 0, tutorial)) %>%
  group_by(filename, session_no) %>%
  mutate(
         exerciseheet_no =  cumsum(exercisesheet_start),
         # indicate exercises
         exercise_start = type == "focusing-block") %>%
  group_by(filename, session_no, exerciseheet_no)%>%
  mutate(
         exercise_no = cumsum(exercise_start),
         activity =
           ifelse(tutorial == 1, "tutorial",
           # exercise activities
           ifelse(text == "Admitted." & grepl("success", type),
                                  "exercise_switch",
           ifelse(grepl("Qed", text),
                  "exercise_qed",
           ifelse(grepl("search", type), "search",
           ifelse(type == "coq-next-beyond-sentences",
                  "exercise_stuck",
           ifelse(type %in% c("coq-exec-next", "coq-exec-prev",
                              "coq-success-sentence",
                              "coq-exec-to", "coq-exec-to-cursor"),
                              "working_on_proof",
```

```
            # errors
            ifelse(type == "coq-add-error-shown" &
                     grepl("Syntax error", error_message),
              "error_syntax",
            ifelse(type == "coq-add-error-shown" &
                      grepl("Nested proofs", error_message),
                     "error_nestedproof",
            ifelse(type == "coq-add-error-shown",
                      "error_add",
            ifelse(type == "coq-execute-error",
                           "error_other",
            ifelse(type == "coq-exec-all",
                   "execute_all",
            # hint
            ifelse(type == "hint-opened",
                   "hint",
            ifelse(
              openedSideWindow == TRUE & sideWindowName == "Common Tactics",
              "common_tactics",
            ifelse(openedSideWindow == TRUE & sideWindowName == "Search Results",
                      "search",
            ifelse(openedSideWindow == TRUE & sideWindowName == "Mathematical Symbols",
                   "symbols",
            ifelse(openedSideWindow == TRUE & sideWindowName == "Commands",
                   "commands",
                 NA
            )
            ))))))))))))))) %>%
  filter(!type %in% c("boot", "startup", "closing", "navigation", "switch-tab",
                      "loaded-file", "open-new-tab", "side-window-change",
                      "focusing-block", "inserting-block", "removing-blocks")) %>%
  ungroup()

# summarize sessions
sessions <- file_ext %>%
  group_by(filename, session_no) %>%
  summarize(
    exercise_session = sum(exercisesheet_start,na.rm = T) > 0,
    workgroup_session = sum(workgroup_time)
    )

file_filt <- file_ext %>%
  left_join(sessions, by = c("filename", "session_no"))  %>%
  # only include sessions with at least one exercise
  filter(exercise_session == 1, workgroup_session == 0) %>%
  select(filename, session_no, activity, real_time)

no_activity <- file_filt %>%
  filter(is.na(activity))

## write to file
write.csv(file_filt, "data_out/final_dataset.csv")

group_file <- file2 %>%
  group_by(filename, session_no) %>%
  mutate(hint = sum(activity == "error_add"))
```

Appendix F

Recommendations for the improvement of Waterproof UX

1. **User onboarding**

   Second round of recommendations for the improvement of user onboarding (based on the qualitative feedback on implemented recommendations – Study 2)

   - Split the Tutorial into two:
     - Creating a tutorial that is short, easy (no challenging proofs), focuses on syntax and functions as an interactive guide. Has almost no tips and is designed to be completed once, as part of the onboarding.
     - Creating a tutorial that is more of a practice document with more difficult exercises that require the combination of functions, more tips and generally more challenge. This document could also serve as a collection of different errors with explanation and troubleshooting possibilities.
   - Increase the visibility of the Waterproof Canvas page: increase its hierarchy within the menu points on Canvas (e.g. same level as Pages or People, etc.) if possible
   - Have more, easier exercises in fist assignments so that people practice and get used to the syntax better.
   - Provide guidance on explaining and troubleshooting errors in a guide. Break down troubleshooting techniques per type of error. Communicate troubleshooting via different means: e.g. creating troubleshooting videos, including a short explanation in first assignments.

2. **Contextual recommendations – Waterproof in Analysis I**

   - Check the complexity of proofs in assignments (especially Assignment 4 and 7) and how the complexity compares to the previous assignment. Make students

aware of more difficult assignments and provide extra guidance in the form of pointing them to specific tactics or exercises of the Tutorial

- Emphasize and increase the satisfaction of completing a proof in Waterproof conveys by e.g. adding a congratulations pop up with animation – this also provides feedback to students about successful proof completion

3. New features

- Recommend students different troubleshooting techniques in case of different errors in the platform (based on success routes on the refined process map) – recommendations should be able to be turned on and off in the settings. Means of recommendations should prototyped and tested by A/B testing (e.g. pop up, notification bell)
- Introduce difficulty levels for proving in exercise sheets/assignments. This way students that are looking for a challenge can get less guidance and students that are struggling with the material can still use Waterproof for checking their proof structure.
- Expand the range of words Waterproof accepts when students use natural language: e.g. “We conclude that..” can be expanded to “We *can* conclude that…”, “I conclude that”, etc. analysis of unsuccessfully executed sentences can inform these changes.
- Support groupwork within Waterproof (more research needed) by building a collaboration module – implement “share” button OR Overleaf export
- Reporting bugs within the platform → add them to a bug-log for the Waterproof Development Team

4. Suggestions for improving the user interface and current functionalities

- Let the proof progress tab show the whole proof structure as a summary instead of only the last executed sentence.

For access to the data on the Miro board please email d.bor@student.tue.nl or bor.dorina98@gmail.com.