Fall 11-12-2021

Facilitating heuristic evaluation for novice evaluators

Anas Abulfaraj
DePaul University, AABULFA2@depaul.edu

FACILITATING HEURISTIC EVALUATION FOR NOVICE EVALUATORS

BY

Anas Abulfaraj

A DISSERTATION SUBMITTED TO THE SCHOOL OF COMPUTING,

COLLEGE OF COMPUTING AND DIGITAL MEDIA OF DEPAUL

UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

DOCTOR OF PHILOSOPHY

DEPAUL UNIVERSITY

CHICAGO, ILLINOIS

November, 2021

DePaul University

College of Computing and Digital Media

Dissertation Verification

This doctoral dissertation has been read and approved by the dissertation committee below according to the requirements of the Computer and Information Systems PhD program and DePaul University.

Name: Anas Abulfaraj

Title of dissertation:

FACILITATING HEURISTIC EVALUATION FOR NOVICE EVALUATORS

Date of Dissertation Defense:

11/12/2021

Adam Steele
Dissertation Advisor*

Xiaowen Fang
1st Reader

Peter Hastings
2nd Reader

Kevin Buffardi
3rd Reader

** A copy of this form has been signed, but may only be viewed after submission and approval of FERPA request letter.*

FACILITATING HEURISTIC EVALUATION FOR NOVICE EVALUATORS

ABSTRACT

Heuristic evaluation (HE) is one of the most widely used usability evaluation methods. The reason for its popularity is that it is a discount method, meaning that it does not require substantial time or resources, and it is simple, as evaluators can evaluate a system guided by a set of usability heuristics. Despite its simplicity, a major problem with HE is that there is a significant gap in the quality of results produced by expert and novice evaluators. This gap has made some scholars question the usefulness of the method as they claim that the evaluation results are a product of the evaluator's experience rather than the method itself.

In response, the goal of this thesis is to bridge the gap between expert and novice evaluators. Based on interviews with 15 usability experts, which focused on their experience with the method, the difficulties they faced when they were novices, and how they overcame such difficulties, it presents a comprehensive protocol called Coherent Heuristic Evaluation (CoHE). This step-by-step protocol guides novice evaluators from the moment they decide to conduct an evaluation until the submission of their evaluation report.

This protocol was verified by conducting an experiment to observe the difference between novices using the CoHE protocol and novices using Nielsen's 10 usability heuristics without the guidance. The experiment involved 20 novices performing two sessions; the first was an understanding session where the novices read and understood the heuristics and the second was an inspecting session

where they inspected a system. The findings show that, while evaluators take more time to read and evaluate a system using CoHE, they tend to identify more problems. The experiment also demonstrated that CoHE can improve the thoroughness, effectiveness, and f-measure of evaluation. However, the validity of CoHE was comparable to that of HE.

ACKNOWLEDGEMENTS

I would like to thank my parents, Dr. Waleed Abulfaraj and Dr. Najlaa Kurdi for their unconditional love and support, not only throughout the PhD journey, but throughout my entire life. I would like to thank my advisor, Dr. Adam Steele, who guided and supported me all along. He is the best advisor anyone could ask for. I would like to thank the committee members: Dr. Xiaowen Fang, Dr. Peter Hastings and Dr. Kevin Buffardi for accepting to be part of the committee and taking the time to read my dissertation and provide suggestions and recommendation to improve the quality of it. I would like to thank Dr. Habib Fardoun, Dr. Cesar Collazos, Dr. Huizilopoztli García, Dr. Jaime Arteaga and Jessica Westbrook for their help in reaching participants. Without their help, this work would have not been possible. I would like to thank my aunt Dr. Heba Kurdi for her continues support and encouragement. Finally, I would like to thank my cousin Mohannad Abdulghani and my friend Saif Alshammari with whom I always exchanged ideas and talked about my work. They were always there for me. From the bottom of my heart, thank you all.

CONTRIBUTIONS

The contributions of this work are the following:

- It provides a detailed list of the problems novices face when conducting Heuristic Evaluation (HE). This is of a great importance as in the literature, no one, to the best of our knowledge provided any comprehensive list of issues that novices face when using HE. This list will help other researchers in understanding the challenges of HE and coming up with different solutions to these issues that might differ from the ones we provided.
- It provides a complete, step-by-step protocol to help novices perform HE. Novices are left to wonder when asked to perform HE. They don't exactly know where to start, or how to approach the system. Therefore, this protocol helps in facilitating the process since its beginning until its end. It guides them through understanding the usability heuristics and the significance of using them, it guides them through the process of evaluating the system, and it guides them through the process of preparing a meaningful and effective report of the usability issues.
- Novices usually get overly concerned with the usability heuristics to the point where they forget the main goals of the evaluation. Usability heuristics are means to an end. Therefore, the goals of the evaluation should be always kept in mind while doing the evaluation. Thus, this work provides a mapping between usability heuristics and the goals of the

evaluation to make the evaluation more aimful and purposeful and to let evaluators know how the usability heuristics contribute to these goals.

- Each usability heuristic is general and could be open to different interpretations. Moreover, it includes many concepts under it. This makes it difficult to novices to fully grasp the heuristics. This is problematic, since understanding anything is the first step to apply it correctly. Therefore, this provides a detailed usability heuristics that are not only explain each heuristic in detail but also explains why it matters, how it works and when not to use it. This helps in enhancing the understandability of the heuristics for novice evaluators.
- Usability heuristics don't lend themselves to be easily applied during the evaluation to detect usability issues. To address this issue, this work provides a list of operational usability heuristics which could be used during the evaluation to facilitate the process of detecting usability issues. The list follows a question-based approach in which the evaluators are provided with questions that they could ask themselves when evaluating any given system. To answer these questions, they would have to examine specific parts of the system, leading them to detect usability problems more easily
- This work provides the results of comparing traditional, unguided usability heuristics against detailed usability heuristics and operational usability heuristics. The results show the quantitative and qualitative differences. Quantitative differences include the time it took to read the heuristics and

evaluate the system, number of problems found, the thoroughness, validity, effectiveness and f-measure. Qualitative differences include the difficulties faced during the process, the confidence levels in the results produced, and the opinions about the methods. This would help in illustrating the advantages and disadvantages of using any of the two methods.

- Finally, this work gives clear directions to what to be followed in the future to further enhance the work of facilitating the use of HE for novice evaluators.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

In the past, technology tended to be used by a specific group of people, most of them either tech-savvy or in possession of enough knowledge and persistence to overcome technical challenges. This tended to be accepted because of the nature of technology then and because the technology tended to be too complicated for ordinary users. The biggest challenge for developers back then was to develop functional systems, and making systems easy to use was considered a luxury. In a time where personal computers had not yet been invented, the pervasiveness of technology was very much lower than today.

After the emergence of personal computers, technology became a key player in everyday life, making systems more usable became an urgent need. Usable systems can significantly reduce costs, by diminishing costs related to training, maintenance and development, errors and support, by reducing employee turnover and task-completion time, and by increasing sales and customer satisfaction (Aydin et al., 2012, pp. 632–637).

The following list offers some real-world examples of such cost reductions:

Reducing training costs. American Express was able to reduce 12 hours of training to 8 hours by redesigning their system (Gibbs, 1997).

Reducing maintenance and development costs. American Airlines was able to reduce maintenance costs by 60–90% by fixing a number of usability problems early in the design phase (Harrison et al., 1994, pp. 203–241).

Reducing errors. A study with the aim of improving the usability of a large wireless carrier showed that errors per screen dropped by 67% (Cope & Uliano, 1995, pp. 263–267).

Reducing support costs. Breastcancer.org redesigned their system to make common tasks easier to perform, reducing the number of help desk support requests by 80%. On average, they previously received 15 requests weekly and, after the redesign, the number of requests dropped to three (Usability First, 2007).

Reducing employee turnover. Usability helps to reduce employee turnover, because when employees work on systems that meet their needs and are easy to use, they are less likely to resign. Two divisions of a hotel corporation were able to reduce employee turnover by 10%, exceeding the overall profit of the two divisions (Karat, 2005, pp. 103–141).

Reducing task time. Bond (2007) described a study in which a call center was able to reduce the length of each call by about one minute by improving usability, saving around $250,000 in a year.

Increasing sales and customer satisfaction. The La Quinta hotel chain improved the usability of their website, thereby increasing their profits by 83% and customer satisfaction by 28% (Peterson, 2007).

Although decreasing costs and increasing profit and customer satisfaction are very important, usability benefits go beyond these outcomes. A lack of usability can have serious consequences, such as health problems and, in the worst cases, even death. Kushniruk et al. (2005) have reported a relationship between certain types of usability problems and errors in prescription of medications. Moreover, a study found unexpected increased mortality after implementation of a computerized physician order entry system, stating that one potential reason for this was "human-machine interface flaws" (Han, 2005). In 1992, 87 people died in an aircraft crash reportedly due to usability issues (Sripathi & Sandru, 2013).

One of the main objectives of the field of Human-Computer Interaction (HCI), which emerged during the 1980s, was addressing usability issues (Issa & Isaias, 2015, pp. 19–36; Lazar et al., 2017; MacKenzie, 2012). HCI is a multidisciplinary field that draws expertise from a broad range of areas including computer science, psychology, sociology and anthropology, aiming at understanding and facilitating interaction between humans and computers (Preece & Rombach, 1994). While the importance of usability is clear, we must ask: "What is usability?"

1.1. Usability

There are various definitions of usability. Bennett (1984), among the first to define usability (Han et al., 2001), described it as follows: "For a PRODUCT to be claimed as usable, it must enable some required percentage of the appropriate range of USERS to carry out some required percentage of the range of TASKS for some required range of usage ENVIRONMENTS." The International Organization for Standardization (ISO) defines usability as the "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use" (ISO, 2018).

One of the most popular definitions of usability is Nielsen's (2012) definition, which states that usability is assessed by five quality components:

Learnability. "How easy is it for users to accomplish basic tasks the first time they encounter the design?"

Efficiency. "Once users have learned the design, how quickly can they perform tasks?"

Memorability. "When users return to the design after a period of not using it, how easily can they reestablish proficiency?"

Error. "How many errors do users make, how severe are these errors, and how easily can they recover from the errors?"

Satisfaction. "How pleasant is it to use the design?"

This suggests that, for a system to be considered usable, it should be not only easy to use but also efficient, effective, memorable, and satisfying. Since usability is an important quality with multiple facets, it should ideally be measured using more than one method (Gray & Salzman, 1998; Hartson et al., 2003), so we must ask: "What are the different usability evaluation methods?"

1.2. Usability evaluation methods

There are many different methods for evaluating the usability of a given system; the most used methods are usability testing and heuristic evaluation

(Liljegren, 2006; Muller et al., 1998; Tan et al., 2009; van Dijk et al., 2007). Usability evaluation methods (UEM) can be divided into two categories: user-based methods (Dumas, 2003, pp. 1093–1117) and inspection-based methods (Nielsen, 1994, pp. 413–414) (Figure 1).

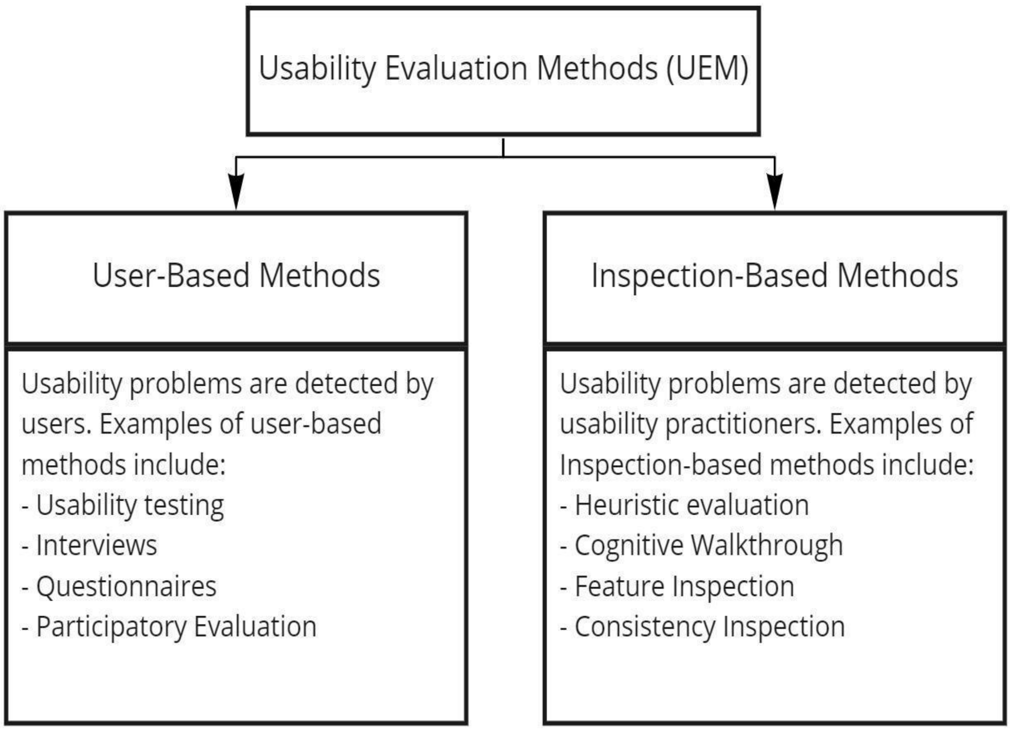

Figure 1. *UEM categories*

The main difference between the two methods is the extent of involvement of actual potential users during evaluation. The former category depends on actual potential users' participation, while the latter category relies on usability experts to perform the evaluation without actual users' involvement. Ideally, both categories should be used in evaluating a system. Inspection-based methods are usually applied earlier in a product development cycle to ensure that all major problems are detected, while user-based methods tend to be applied later to ensure that real users could effectively use the system prior to launching the product.

1.2.1. User-based methods

User-based evaluation methods aim to detect usability problems by gathering real user data and analyzing the performance and the opinions of actual users. User-based methods are mostly summative and take place when the system is ready to be launched (Tullis & Albert, 2008).

The main advantage of user-based methods is that they yield more accurate insights into the problems that users may encounter. However, such methods are relatively expensive and require considerable time and resources. User-based methods include usability testing, interviews, satisfaction questionnaires, and participatory evaluation (Maguire, 2001).

Usability testing. Considered the gold standard method, usability testing typically involves bringing potential users into a lab and assigning them a number of realistic tasks to perform using the system, hoping to reveal as many usability issues as possible. The number of participants can vary depending on the budget and the schedule, but Virzi (1992) suggests that five participants tends to reveal 80% of the usability problems.

Interviews. Designed to obtain direct feedback from the stakeholders, interviews help to understand opinions, preferences, and attitudes with respect to a specific system. Interviews can be structured, semi-structured or unstructured (Robson & McCartan, 2016). While the main advantage of this method is that it can allow an interviewer to obtain deep insights about the system, it can be time-consuming to conduct interviews and analyze results.

Questionnaires. Across all fields of studies, questionnaires are one of the most used research methods (Babbie, 2013). Questionnaires are often used in the field of human-computer interaction because they help by collecting data from a large number of people within a short period of time and can usually be conducted at a minimal cost. However, the results of questionnaires are often of little depth.

Participatory evaluation. Users are given specific tasks or scenarios to accomplish using the system and encouraged to verbalize their activities throughout the process. A moderator asks the participants questions during the process to clarify anything unclear. There are multiple participatory evaluation variations, such as evaluation workshops (Robinson & Fitter, 1992) and evaluation walk-throughs (Maulsby et al., 1993, pp. 277–284).

1.2.2. Inspection-based methods

Inspection-based evaluation methods aim to uncover usability problems via system inspection by usability experts. Usability experts draw on usability guidelines, heuristics, personal experiences, and imagining themselves as a user to discern usability problems. Inspection-based methods are usually formative, taking place during the early stages of the development life cycle, and iterative (Tullis & Albert, 2008). While inspection-based methods are generally less expensive, require fewer resources, and are not excessively time-consuming, they are subject to the expertise effect and could produce false-positive results or false-negative results. Inspection-based methods include heuristic evaluation, cognitive walkthrough, guideline review, and consistency inspection (Nielsen, 1994, pp. 413–414).

Heuristic evaluation. Introduced by Nielsen and Molich in the 1990s, heuristic evaluation seeks to evaluate a system based on a number of guidelines/heuristics (Molich & Nielsen, 1990; Nielsen & Molich, 1990, pp. 249–256). This evaluation is completed by three to five usability specialists, with every specialist separately inspecting the system followed by aggregation of the results.

Cognitive Walkthrough. Another inspection-based method also developed in the 1990s (Lewis et al., 1990, pp. 235–242, 1997, pp. 717–732), cognitive walkthrough involves a usability specialist acting out the role of a real user and attempting to accomplish distinct tasks. After each step towards task completion, the specialist asks a number of questions, such as: "Will the user try and achieve the right outcome?"; "Will the user notice that a correct action is available?"; "Will the user associate the correct action with the expected outcome?"; "If the correct action is performed, will the user see that progress is being made towards their intended outcome?". System usability is evaluated based on the answers to these questions.

Feature Inspection. In feature inspection, the inspector identifies the major tasks completed by the system and then identifies the set of features that constitute each task. Each feature is evaluated in terms of its understandability, availability, effectiveness, and simplicity.

Consistency Inspection. This approach focuses on evaluating the system in terms of consistency, asking questions such as: “Do menus look the same across the whole system?”; “Do similar labels produce the same effect?”; “Does the system use the same patterns?”. This inspection is usually completed by all the designers who worked in developing the system.

1.3. Heuristic evaluation

Heuristic evaluation (HE) is one of the most widely used UEM (Barnum, 2011; Rosenbaum et al., 2000, pp. 337–344). It was first developed and proposed by Nielsen and Molich in 1990 (Molich & Nielsen, 1990; Nielsen & Molich, 1990, pp. 249–256), then refined and enhanced by Nielsen in 1994 (Nielsen, 1994, pp. 152–158). HE is a “discount” usability method that seeks to find as many usability problems as possible using minimal resources (Nielsen, 1994). Several usability specialists inspect a particular system based on a number of usability guidelines or heuristics. In HE, each specialist inspects the system separately and, when all specialists have finished their inspections, their results are aggregated and consolidated into a single list of recommendations (Figure 2).

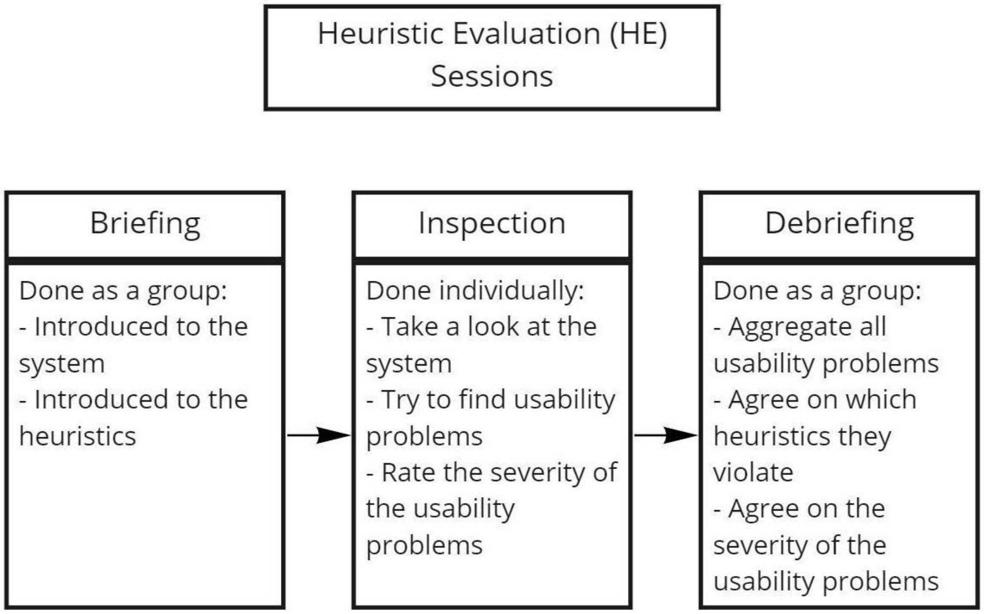

Figure 2. *HE sessions*

Nielsen proposed that the following 10 heuristics should be used during the inspections:

Visibility of system status. “The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.”

Match between the system and the real world. “The system should speak the users’ language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.”

User control and freedom. “Users often choose system functions by mistake and will need a clearly marked ‘emergency exit’ to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.”

Consistency and standards. “Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.”

Error prevention. “Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.”

Recognition rather than recall. “Minimize the user’s memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.”

Flexibility and efficiency of use. “Accelerators—unseen by the novice user—may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.”

Aesthetics and minimalist design. “Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.”

Help users recognize, diagnose, and recover from errors. “Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.”

Help and documentation. “Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user’s task, list concrete steps to be carried out, and not be too large.”

1.3.1. Briefing session

During the briefing session, the evaluators are introduced to the system they are being asked to evaluate and to the heuristics they will use as a reference in identifying usability problems. They are given answers to any questions they have, and existing ambiguities are clarified.

1.3.2. Inspection session

The inspection session is the major stage in HE during which each evaluator separately begins to inspect the system. Usually, an evaluator takes a quick look at the system to develop a general feeling about it, then begins a more detailed inspection of all system aspects, either page by page or by choosing major tasks. The evaluator identifies usability problems based on the given set of heuristics, personal experience, or other usability guidelines. After identifying all usability problems, the evaluator begins assigning a severity level to each usability problem. Nielsen’s (1995) scale is the most commonly used severity scale:

“0 = I don’t agree that this is a usability problem at all.”

“1 = Cosmetic problem only: need not be fixed unless extra time is available on the project.”

“2 = Minor usability problem: fixing this should be given low priority.”

“3 = Major usability problem: important to fix, so should be given high priority.”

“4 = Usability catastrophe: imperative to fix this before the product can be released.”

1.3.3. Debriefing session

During this session, evaluators gather to review identified usability problems in terms of severity and to match usability problems to specific heuristics, agreeing upon the severity of each usability problem and to which heuristic each belongs and producing a final report.

1.4. Problems

Despite its wide usage, HE suffers from multiple problems, with some researchers even questioning its reliability and benefits (Cockton & Woolrych, 2002). One of the known downsides of HE is the so-called expertise effect, that evaluation results depend on the experience of the evaluators (Nielsen, 1992, pp. 373–380); the more experienced an evaluator, the better the results. Some researchers have claimed that the results of HE may be more the product of the experience of the evaluator rather than the HE itself (Cockton & Woolrych, 2001, pp. 171–191). This is possibly because HE is not a structured method (Cockton & Woolrych, 2002), meaning that there is no step-by-step guide on how to perform it. Novice evaluators can find it very difficult to perform.

While ideally HE should be performed by usability experts, studies show that most HE sessions are conducted by novice evaluators (Bruun & Stage, 2014, pp. 1148–1167, 2015, pp. 40–53; Paz & Paz, 2015, pp. 212–223; Renzi et al., 2015, pp. 339–347), suggesting that most of the results produced may be of low quality. Relying on novice evaluators is common for several reasons: hiring experts is expensive, especially for small and startup companies (Koutsabasis et al., 2007); sometimes it is hard to connect with experts; and, if the product schedule is tight, waiting for experts to perform an evaluation might not be feasible.

Still, depending on novices can be extremely harmful. As HCI is a multidisciplinary field, HCI practitioners come from a wide range of different backgrounds, with some lacking formal HCI education (Rajanen et al., 2017, pp. 218–239). Ideally, potential users should be involved in the development life cycle, but HCI practitioners are sometimes expected to represent the user (Iivari, 2006, pp. 185–194), increasing the importance of their roles. Moreover, while HE should ideally be performed by as many as three to five evaluators to increase the likelihood of detecting most usability problems, a survey conducted in Malaysia found that companies sometimes hire only a single HCI practitioner (Hussein et al., 2014, pp. 62–67). Consequently, there is a need to develop a step-by-step guide or protocol to improve novice evaluators' performance.

To produce such a protocol, we must look at the major problems that occur at each stage of the HE. Some known issues for novices regarding HE include: difficulties understanding broad and potentially confusing heuristics during briefing sessions (de Lima Salgado et al., 2016, pp. 387–398); difficulties applying heuristics that require additional explanation for detecting usability problems; a lack of typical ways of approaching the system during inspection sessions, meaning that evaluators are left to determine the approach independently, which may be overwhelming for novice evaluators; and a lack of clarity over assigning severity ratings and usability problems to heuristics, which are generally left to prior experience and understanding that novice evaluators may not possess.

1.5. Significance

The main contribution of this work is to create a step-by-step protocol, with the aim of helping novices improve the results of their evaluations. The central goal is to democratize the use of HE, making it more accessible to a wide range of evaluators, not only in the field of HCI but also from outside the field. The protocol will transfer experts' knowledge and experience to novice evaluators in a simplified manner that is easy to follow. First, the protocol will help evaluators to understand why they should perform HE and to deeply comprehend the usability heuristic they are going to apply. Second, it will explain how to approach the system, how many sessions they should perform, when to conduct the evaluation, how long it should take, and how to accurately detect usability problems. Finally, it will show how to accurately decide which heuristics a certain problem violated, how to accurately rate the severity of the usability problems, and how to produce a convincing and impactful report. As a standalone material, the step-by-step protocol does not require any external help; it will guide evaluators from the moment of deciding to conduct the evaluation until they produce the final report.

1.6. Objectives and research questions

The main objective of this study is to facilitate HE for novice evaluators, rather than to improve HE as a method per se. It aims to help novices to improve their performance and the quality of results they may produce. We want to bridge the gap between the results produced by expert evaluators and by novice evaluators. Two factors tend to distinguish experts from novices in this context: knowledge and techniques. Experts know more about usability and usability problems than novices, and they have previously developed their own strategies for tackling a system and detecting its usability problems. The goal is to transfer both knowledge and techniques from experts to novices by learning from experts and, on this basis, creating a step-by-step protocol for novices. A final step is assessing the effectiveness of the proposed protocol (Figure 3).

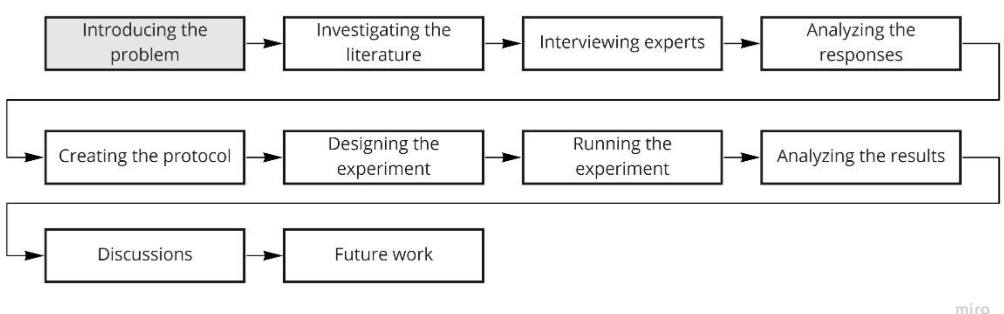

Figure 3. *Thesis organization (Chapter 1)*

1.6.1. Objectives

As mentioned above, the main objective is to create a comprehensive protocol for facilitating HE for novices and to validate this proposed protocol. This task requires accomplishing six sub-objectives:

- Provide novices with the most effective ways to understand heuristics.
- Identify the best ways to approach a system during an inspection.
- Find the best ways to document usability problems.
- Identify the best ways to deliver the results of HE.
- Develop a coherent protocol that could be applied to any set of heuristics.
- Apply parts of the protocol to compare with traditional HE to see how they differ, quantitatively and qualitatively.

1.6.2. Research questions

To achieve the objectives stated in the previous section, there are a number of research questions to be answered within this study:

RQ1: What are the general problems specialists face when conducting HE?

RQ2: What are the potential solutions for the general problems?

RQ3: What are the difficulties specialists face in understanding usability heuristics?

RQ4: What are the ways to overcome those difficulties?

RQ5: What are the best ways to implement HE?

RQ6: What are the best ways to document usability problems?

RQ7: What are the best ways to deliver the results of HE?

RQ8: What are the differences between the parts of coherent heuristic evaluation (CoHE), detailed usability heuristics and operational usability heuristics, and traditional HE in terms of the number of usability problems found, the severity of the usability problems, and validity, thoroughness and effectiveness?

CHAPTER 2. LITERATURE REVIEW

The concept of design and usability guidelines is an established idea; the Shneiderman principles were published in 1987. However, as stated above, Nielsen and Molich's (1990) study was the first to propose the HE method. After the initial emergence of this technique, many researchers began to publish papers concerned with its improvement. HE-related studies have taken various different directions: the first has focused on developing usability guidelines or heuristics, a trend that began even before the creation of the HE method; the second has concentrated on developing guidelines for specific domains, audiences, platforms, or contexts; and the third has focused on improving the technique itself by enhancing some of its elements. However, no one has yet developed a complete protocol, especially aimed at novices (Figure 4–Figure 5).

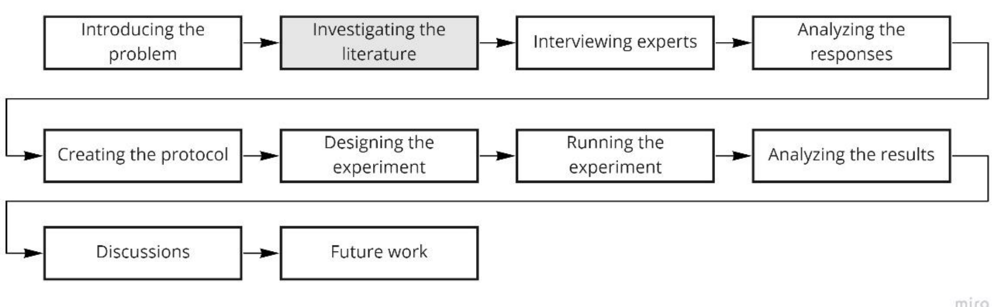

Figure 4. *Thesis organization (Chapter 2)*

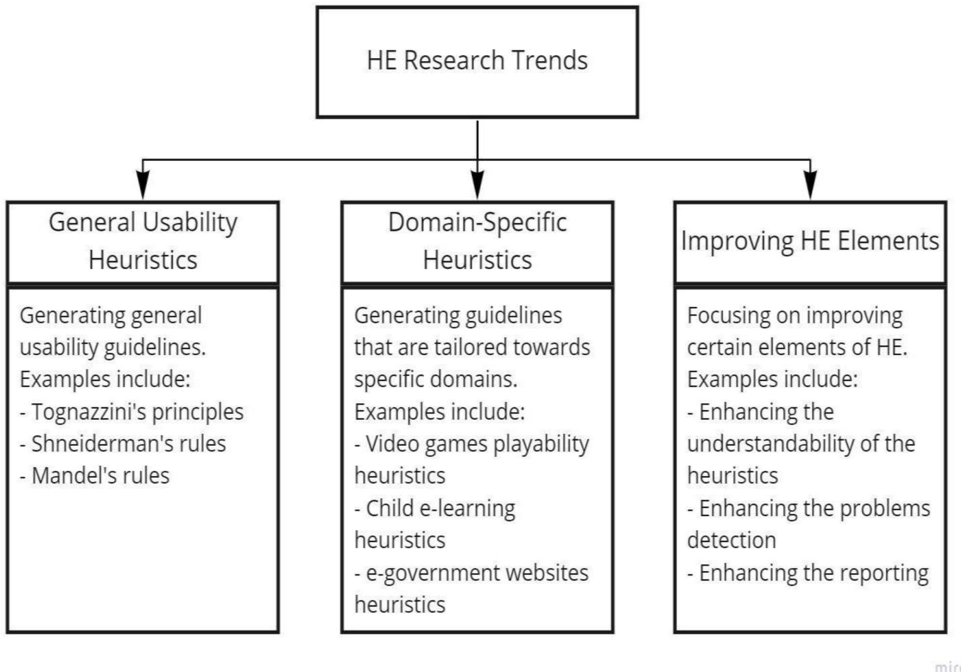

Figure 5. *HE trends*

2.1. Usability guidelines

The concept of developing usability guidelines preceded the creation of HE as a method. Many HCI and UX professionals have proposed variations of usability guidelines or rules of thumb meant to help designers to design systems and assess their usability. The source of such guidelines is usually the experience of each study's author. Shneiderman (1987) proposed eight golden rules and general principles for designing user interfaces and has revised these multiple times (Shneiderman et al., 2016). Based on his experience in the field, Shneiderman's rules are to strive for consistency, seek universal usability, offer informative feedback, design dialogs to yield closure, prevent errors, permit easy reversal of actions, keep users in control, and reduce short-term memory overload.

Tognazzini (2003) devised a more expansive set of design principles and sub-principles. The 19 principles, applicable for Graphical User Interface (GUI) environments ranging from traditional GUI's to wearable devices, are as follows: aesthetics, anticipation, autonomy, color, consistency, defaults, discoverability, the efficiency of the user, explorable interfaces, Fitts's law, human interface objects, latency reduction, learnability, metaphors, protect users' work, readability, simplicity, track state, and visible navigation.

Norman, one of the most prominent HCI experts, laid out seven principles for usable designs in *The Psychology of Everyday Things* (1988), which was later revised and expanded (Norman, 2013). In this influential book, Norman stated that designers should use both field's knowledge and intellectual knowledge, simplify tasks' structure, make things visible, get the mappings right, use the power of constraint, design for errors, and standardize. However, Norman's guidelines, along with Tognazzini's principles, seem better suited as guidelines for designing systems rather than for assessing system usability.

Through her work on cognitive science, Gerhardt-Powals (1996) drew ten principles for usable systems (pp. 189–211), then used these principles to assess a submarine-firing system's usability, with results showing that systems designed following these principles were more functional than other systems. These principles are as follows: automate unwanted workload, reduce uncertainty, fuse data, present new information with meaningful aids to interpretation, use names that are conceptually related to function, group data consistently and in meaningful ways, limit data-driven tasks, include in displays only that information needed by the operator at a given time, provide multiple coding of data when appropriate, and practice judicious redundancy. Hvannberg et al. (2007) and Sohl (2017) compared these principles to Nielsen's (1994) heuristics, revealing that both sets of heuristics were comparable with respect to the number of problems found, albeit with slight differences; the former suggested that both sets of heuristics produced the same results, while the latter found that Gerhardt-Powals' principles produced slightly better results.

Atkinson et al. (2007) created a comprehensive set of guidelines called multiple heuristic evaluation table (MHET) (pp. 563–572), with the underlying idea of integrating multiple sets of heuristics into one comprehensive set by merging similar procedures and adding unique heuristics. MHET, based on Nielsen's heuristics, Shneiderman's rules, Tognazzini's principles, and Edward Tufte's principles (University of Washington Computing & Communications, 2005), consists of 11 principles: software-user interaction, learnability, cognition facilitation, user control, and software flexibility, system-real world match, graphic design, navigation and exiting, consistency, defaults, system-software interaction, help and documentation, and error management.

Granollers (2018) offered another attempt to create a new set of guidelines based on existing approaches (pp. 396–405). Drawing on Nielsen's heuristics and Tognazzini's principles, the authors used a three-step process to derive a new set of heuristics: they first examined each heuristics set separately to deeply understand each individual heuristic; they compared similarities between Nielsen's heuristics and Tognazzini's principles; and they subsequently integrated these similarities. The final result was 15 heuristics: visibility and system state; connection between the system and the real world; metaphor usage and human objects; user control and freedom; consistency and standards; recognition rather than memory; learning and anticipation; flexibility and efficiency of use; help users recognize, diagnose, and recover from errors; error prevention; aesthetic and minimalist design; help, and documentation; save the state and protect the work; color and readability; and autonomy, defaults, and latency reduction.

In addition to these published heuristics and guidelines, there are several other usability guidelines. Mandel (1997) proposed a list of three golden rules: place the user in control, reduce user memory load, and make the interface consistent. Under each of the three rules, there are lists of sub-rules. Moraveji and Soesanto (2012) also identified 10 design heuristics used to reduce stress: reveal the ability to control interruptions, reduce feelings of being overwhelmed, acknowledge human interpretations of time passing, use appropriate tone and emotion, provide positive feedback to user input and events, encourage prosocial interaction, relieve time pressure, choose naturally-calming elements, acknowledge reasonable user actions, and demystify the interface. Their advice is to use these heuristics in combination with a set of other heuristics.

2.2. Domain-oriented heuristics

Given the integration of technology as an essential part of everyday life, specific usability requirements are needed for people in different disciplines, contexts, or age ranges. As the usability guidelines outlined above are quite general, researchers began working on developing a set of heuristics tailored towards specific audiences, platforms, disciplines, and contexts. There are heuristics related to video games, mobile devices, e-learning, and so on.

Desurvire et al. (2004) created a set of heuristics for assessing video game playability. The design of video games is fundamentally different from standard productive systems. Video games are meant to be challenging and not as direct as typical systems. Realizing these profound differences, the researchers created a set of heuristics for assessing game playability, aiming to not only evaluate the usability of the interface but also assess game playability. Since evaluating playability includes factors beyond the interface alone, the researchers divided their heuristics into four major themes: a gameplay in which the player has to overcome obstacles in order to win; a game story concerned with the narrative and the characters of the game; game mechanics, related to how the game is programmed and how the various elements of the game react to player actions; and game usability, pertaining to the interface's usability and the controllers used to interact with the interface. Within each theme, there are a number of heuristics, resulting in 43 heuristics for evaluating playability (HEP). To validate HEP, user testing, alongside HE using HEP was performed, and the results showed that HEP was able to find a greater number of issues than user testing only.

Korhonen and Koivisto (2006) have focused on the playability of mobile games (pp. 9–16). Mobile video games have a unique quality, mobility, which differentiates it from other video games. Their study thus developed a new set of heuristics, which are divided into three categories: usability, which deals with interface and controller issues; mobility, concerned with the effect that mobility has on the game itself; and a gameplay related to story development. Each category has its own heuristics, resulting in 29 heuristics in total. To validate their heuristics set, the authors selected five mobile games for evaluation, and two to four evaluators evaluated each game. The evaluators were able to detect 235 playability problems, proving the effectiveness of the heuristics.

Inostroza et al. (2013) have examined the usability of touchscreen-based mobile devices (pp. 24–29). As they have smaller screens and different settings, due to being operated by users' hands, these devices are differentiated from traditional interfaces and therefore require a specific collection of heuristics to address their unique problems. The authors proposed 12 heuristics, mostly based on Nielsen's heuristics but with redesigned definitions or additional heuristics suitable for mobile devices. To test the validity of the proposed heuristics, they used three techniques. First, they ran inquiry tests through an online survey to test whether these heuristics were understandable by evaluators or not. 27 evaluators participated in the survey, and the results showed that their heuristics were understandable for most participants. Second, they performed Heuristic Evaluation by examining mobile devices using both their proposed heuristics and Nielsen's heuristics, showing that their heuristics outperformed Nielsen's heuristics. Finally, their proposed heuristics were reviewed and evaluated by usability experts.

Educational systems are another area where researchers are developing new heuristics to respond to the critical issues and address shortcomings in a general heuristics approach. Alsumait and Al-Osaimi (2010) created a set of heuristics, called HECE and derived from Nielsen's heuristics and other usability guidelines, to evaluate child e-learning applications (pp. 425–430). The proposed heuristics of HECE are divided into three main groups: the first is Nielsen's heuristics, dealing with general usability issues; the second is child usability heuristics, concerned with issues related to children's capabilities and preferences; the third is e-learning heuristics, dealing with learning issues. The total number of HECE heuristics is 21. To assess HECE's effectiveness, the authors chose two child e-learning applications to be evaluated by usability experts and tested by children. The results showed that HECE is able to detect more issues than usability testing.

Age is an important factor in technology use, as elderly people, in particular, may face difficulties while using technology. Al-Razgan et al. (2014) developed a set of heuristics to evaluate the usability of mobile launchers by older adults (pp. 415–424), drawing on their heuristics on their previous work with older adults (Al-Razgan et al., 2012, pp. 568–574) and related to existing touch-based mobile device heuristics (Zhang & Adipat, 2005). They proposed a total of 13 heuristics divided into three main sections: look and feel, interaction, and functionality. To validate the proposed heuristics, they chose six applications and recruited four HCI students to evaluate them. Their heuristics appeared to be both quantitatively and qualitatively effective in detecting usability problems.

Culture is another factor which influences technology use. Every culture has its preferences when it comes to the design of systems and their applications and such preferences should be considered when designing any application intended for use by a specific culture. Ariffin and Dyson (2015) generated a set of heuristics for mobile learning applications designed to be used by a Malaysian audience. The authors stated that four significant elements must be considered when designing a mobile learning application for Malaysian users: local language, local philosophy, local aesthetic values and colors, and local cultural content.

While this review exhibits a trend toward tailoring heuristics toward specific domains, this is by no means an exhaustive list of all studies that have attempted to generate domain-oriented heuristics. There are many other papers with this aim: Al-Khalifa et al. (2016) created a set of heuristics for e-government websites in Saudi Arabia (pp. 375–378); Mi et al. (2014) provided heuristics for evaluating smartphones' accessibility, specifically by people with visual impairments and upper-extremity disabilities; Baker et al. (2002) developed heuristics for evaluating shared workspace groupware systems (pp. 96–105); Chanco et al. (2019), concerned with automated teller machine (ATM) use, produced a set of 18 heuristics for measuring usability (pp. 3–18); and Saavedra et al. (2019) produced a set of 11 heuristics for evaluating usability and overall user experience (UX) of social networks (pp. 128–139).

2.3. Improving HE

While many researchers and HCI specialists have focused on developing general usability guidelines and tailoring guidelines directed toward specific domains, HE as a method also needs improvement. Many HCI researchers have begun re-evaluating the method's effectiveness by enhancing its various elements to help evaluators better utilize it. To this end, they have worked on improving the understanding of the guidelines, finding better ways to implement HE, and enhancing reporting of usability issues.

Cronholm (2009) realized the importance of making usability guidelines themselves usable, as such guidelines are only helpful if evaluators can fully understand them (pp. 233–240). They addressed two of the most popular usability heuristics, Shneiderman's eight golden rules and Nielsen's 10 heuristics, to examine their understandability and applicability. After analyzing them in isolation using existing theories and by interviewing usability experts, they concluded that these heuristics should be presented and organized differently to be more usable, suggesting that any set of heuristics should be relevant to be functional, should explain why their particular heuristic is important to be applicable, and should present their guidelines in active rather than passive form and embody different abstraction levels to increase understandability, alongside other principles for improving usability.

Performing HE without keeping the target audience in mind could lead to detecting trivial issues that might not affect real users, raising the risk of distracting evaluators from finding real usability issues. Addressing this problem, Friess (2015) examined the inclusion of personas during the process of HE to determine whether or not this affects HE results. They recruited evaluators and assigned them to different groups to perform HE: the first group performed a regular HE; the second group performed HE and were given personas for the target audience; and the third group was asked to develop their personas and use them to perform HE. However, they found that there was no significant difference in the results from the different groups, although each group's types of problems somehow differed.

Another way to improve the effectiveness of HE is to conduct HE based on specific scenarios. Po et al. (2004) and Varsaluoma (2009) both applied this method, called heuristic walkthrough (HW), for the evaluation of mobile usability. In the first study, the authors recruited three groups of evaluators: the first evaluated a mobile device using traditional HE; the second used HW; and the third used a method called contextual walkthrough (CW). They found that HW outperformed both HE and CW in detecting usability issues. In the second study, a mobile device was evaluated by evaluators assigned to two groups, the first using HE and the second using HW. The results suggest that there are no significant differences between the two methods. Given that the results of these studies contradict one another, further research may be needed to examine the usefulness of HW.

Personas and scenarios can be added to improve organization and to keep the user in mind while conducting HE. Muller et al. (1998) proposed adding real users to the team of evaluators while performing HE, a method called participatory heuristic evaluation (PHE). However, the user to which they referred was not necessarily an end-user but rather a user called a work-domain expert, who has experience with the domain they are evaluating. The authors added new heuristics to Nielsen's original list but, since they did not conduct experiments to validate their proposed method empirically, PHE may need to be tested further to examine its effectiveness.

Including real users in HE is not the only innovation that has been proposed by researchers. Sijavi et al. (2013) have suggested integrating usability testing and HE into a method called hybrid usability methodology (HUM) (pp. 375–383). To validate their method, they presented four case studies to which it was applied. The four case studies contained prototypes and systems in the early stages of development in e-government, e-learning, and e-commerce. The results revealed an improvement in the number of usability issues found. However, this method is somewhat problematic because it is sometimes difficult to find real potential users to participate in usability testing sessions and applying both techniques could be costly and time-consuming.

Attempting to apply all the heuristics outlined above in one session could be exhausting and overwhelming. Kurosu et al. (1997) recognized this problem and developed the structured heuristic evaluation method (sHEM), modifying the traditional HE to make it more structured (pp. 2613–2618). In this method, the heuristics list is broken down into smaller categories and, rather than performing the HE in only one session, the evaluator applies each category during a different session. This may also lead to detecting a more significant number of usability problems, as the evaluator can focus on only one aspect of the system in each session. Comparing sHEM to HE revealed that sHEM detected double the number of usability problems than HE (Kurosu et al., 1999, pp. 938–942).

An alternative approach to usability guidelines in HE is to apply different perspectives (Cockton et al., 2004, pp. 145–161). Rather than using heuristics as a means for detecting usability problems, evaluators inspect the system keeping in mind one of three different perspectives: a novice use perspective, an expert use perspective, and an error handling perspective. When applying each perspective, the evaluator is asked to think about Norman's (2013) seven stages of action to help in the evaluation. The main difference between HE and perspective-based usability inspection is that the latter does not rely on the use of heuristics, as evaluators are asked to perform specific tasks using a specific perspective. Perspective-based assessment offers a more structured approach to evaluation, making the process easier for evaluators and finding more usability problems than traditional HE.

The format of the usability issues report is an essential element in HE. Evaluators tend to use a simple report format that includes a problem description and the heuristics it violated. Cockton et al. (2004) attempted to improve the formatting of reports by proposing a comprehensive report (pp. 145–161). This report is divided into four parts: in the first part, the evaluator should state the problem description, likely/actual difficulties, specific contexts, and assumed causes; in the second part, they should describe discovery resources and methods; in the third part, they should discuss why the problem was related to a particular heuristic; and, in the last part, they should explain why the problem should be eliminated.

2.4. Usability experts

What makes anyone an expert? And how does anyone become an expert? These are difficult questions that might seem to have no definitive answers. This is true across all disciplines, but at different levels of precision. Some fields have a higher level of inter-rater reliability, while other fields have lower levels. For example, weather forecasting has a high inter-rater reliability level at around 95%, while the level of inter-rater reliability for clinical psychology is only about 40% (Thomas, 2018). One reason for this discrepancy between different disciplines is that validating outcomes is more challenging in some fields, particularly in domains dealing with human behavior, which is difficult to predict (Thomas, 2018). Since usability evaluation deals with making systems usable for users, it can be challenging to define usability experts.

Given that it is difficult to define who is an expert, knowing how someone might become an expert is equally challenging. Deliberate practice, one of the most well-known methods for becoming an expert (Ericsson et al., 2007), states that there are three main activities required to become an expert: first, practice deliberately by improving what one already knows and practicing new things; second, spend an appropriate number of hours in practice, which some regard as 10,000 hours or about a decade of practice; third, find a coach or a mentor to guide you through the practice. However, this approach has received criticism from a number of researchers. Hambrick et al. (2014) suggested that, while it is essential, deliberate practice alone cannot account for how someone becomes an expert. By analyzing musicians and chess players, they found that some require less practice than others to achieve superior performance and they identified important additional factors to deliberate practice, such as starting age, IQ, and personality.

Defining experts and examining the indicators of an expert is a topic of research across many different fields. For example, Al-Banna et al. (2016) investigated indicators of expertise for software security professionals (pp. 139–148). Still, little work has been done on this topic in the usability field; this review found only one paper by Botella et al. (2014) which addressed this matter, basing their classification on the idea of deliberate practice. They classified usability practitioners into the following five classes:

Novice. No university degree, but at least one course of training and few hours of usability practice.

Beginner. No university degree, but a number of training courses or 2,500 hours of usability practice.

Intermediate. Has a university degree or less than 5,000 hours of usability practice.

Senior. Has a master's degree or less than 7,500 hours of usability practice.

Expert. Has a master's degree or PhD, and more than 10,000 hours of usability practices.

2.5. HE for novices

HE results are highly influenced by evaluator expertise (Nielsen, 1992, pp. 373–380). A few years after the introduction of HE by Nielsen and Molich, Slavkovic and Cross (1999) conducted a study to assess novice evaluators' capability to evaluate complex interfaces (pp. 304–305). They considered a group of 43 graduate and undergraduate students taking an introductory course in HCI to evaluate a personal organizer device. The evaluators were divided into nine groups, each group containing four to six evaluators. On average, each group found only 23% of the usability problems, even though Nielsen had claimed that novice evaluators could find around 51%. This suggests that more work is needed to improve the quality of HE for novice evaluators.

Botella et al. (2018) conducted a study seeking to understand how novice evaluators perceive HE. In their study, 31 students were asked to evaluate two websites using Nielsen's heuristics, then asked to answer questions related to each individual heuristic as well as the whole HE method. The results showed that most students think that while HE is not easy to perform, it is a useful method. When asked about the completeness of the HE, the answers were variable: some students think that Nielsen's heuristics are complete, while others do not agree with this assessment. Alongside results from a previous study by the same researchers (Rusu et al., 2018, pp. 112–120), students' answers to an open question asking them to express their opinions provided some insights into the problems that students face. These included: difficulties linking usability problems to the right heuristic; overestimating the frequency of usability problems; identifying technical problems rather than usability problems; and finding it difficult to write the final report.

Having observed that some students experience difficulties distinguishing between some of the heuristics, de Lima Salgado and de Mattos Fortes (2016) offered another study examining difficulties faced by novice evaluators (pp. 387–398). They conducted three surveys: in the first, they asked 13 usability experts about situations where novice evaluators might have difficulties distinguishing between some of the heuristics; in the second, they asked 15 novice evaluators about the difficulties they have in understanding the heuristics; in the third, they asked seven usability experts to find solutions for issues that the first and second surveys disclosed. This revealed that novice evaluators found it difficult to distinguish between two heuristics, "User control and freedom" and "Flexibility and efficiency of use", and identified the solution to change the heuristic names; "User control and freedom" were changed to "Control to undo and redo actions", while "Flexibility and efficiency of use" were changed to "Accelerators, shortcuts, and efficiency of use."

Wodike et al. (2014) tested whether or not children are able to perform HE, especially for child-related game systems (pp. 353–358). They recruited 20 children aged 12–13 years and divided them into four groups, each containing four evaluators and one facilitator trained for 30 minutes on how to perform HE. Each group was asked to evaluate a game played on the iPad. The results of the evaluation were unsatisfactory; children who acted as facilitators found it difficult to explain the heuristics to the evaluators and the children who had acted as evaluators were unable to detect any problems as they were distracted by playing with other features of the device. Salian and Sim (2014) similarly recruited 12 children between the ages of 10–11 years to play a music game on a laptop, but used a different approach that provided the children with a simplified set of heuristics that contained five specific rules related to the game, and a simplified severity rating based on facial expressions. While the children were able to find a significant number of usability issues, they still struggled with linking the problems to the heuristics.

In an effort to transfer knowledge from expert evaluators to novices, de Lima Salgado et al. (2016) surveyed four usability experts, asking them about the tactics they use in HE that could possibly help novice evaluators (pp. 2931–2946). Their research identified 38 tactics, which were then related to specific heuristics, at around three to six tactics for each heuristic, and also suggested that novice evaluators should experience different usability methods, as this leads to better understanding and better performance.

Lanzilotti et al. (2011) have suggested using pattern-based inspection (PB) to improve the inspection of novice evaluators, relying on patterns rather than general guidelines or heuristics. To test the effectiveness of this method, the authors compared it against traditional HE and thinking aloud (TA), 98 undergraduate students were recruited and assigned to different methods and an e-learning system was used for evaluation. The three methods were compared based on reliability, design impact, effective range, cost, validity, and severity ratings. This revealed that PB has some potential because it outperformed the other methods in terms of reliability and design impact and it performed better than HE and equal to TA in terms of validity, although it fell short of expectations in the other three criteria.

Desurvire and Thomas (1993) also found the programmed amplification of valuable experts (PAVE) method to be useful for novice evaluators (pp. 1132–1136). In this method, evaluators approach the system from ten different suggested perspectives: self, worried mother, human factor expert, cognitive psychologist, behaviorist, social psychologist, anthropologist, Freudian, spoiled child, and health advocate. The evaluator is asked to examine the system ten times, using a different perspective each time. The study showed that PAVE has some potential to improve performance of novice evaluators because it decreased the number of false-positive problems found.

Alqurni et al. (2018) suggested including real end-users within the process of HE to help novice evaluators. They proposed two versions of HE that include real end-users: user exploratory session of heuristic evaluation (UES-HE) and user review session of heuristic evaluation (URS-HE). In the first version, the user explores the system immediately before the evaluator inspects it, and the evaluator asks the user questions regarding their experience with the system. In the second version, the user explores the system after the evaluator has inspected it, and the evaluator attempts to review their evaluation results. Comparing the results of UES-HE and URS-HE with HE showed that both versions were superior to HE: UES-HE found a total of 50 problems, URS-HE found 36 problems, while HE found only 24 problems.

Another way to enhance the quality of HE is evaluating a system as a group, using a method called collaborative heuristic evaluation (CHE) (Petrie & Buykx, 2010), which has been tailored to improve the performance of novice evaluators (de Lima Salgado, 2017). In this study, three groups were formed: novice evaluators, expert evaluators, and a mixed group of novice evaluators and experts. Each group collaborated on evaluating a website, with the goal of seeing whether or not the presence of an expert alongside novices would improve the quality of their work and bring their performance closer to the expert group. The outcomes of the evaluation affirmed that the mixed group of novices combined with experts produced results that were closer in quality to the expert group than the novice group.

Through this examination of the literature, we can see that prior research has not completely addressed the issue of developing a complete protocol for HE aimed at novices. Some papers have addressed some aspects of this problem, such as dealing with the understandability of some sets of heuristics. Others have attempted to improve HE by using personas, but persona creation is a difficult endeavor, especially for novices. Some have suggested including real experts or real users in the process but as discussed above, experts or real users are not always accessible. This evidences the need for a coherent protocol to guide novices from the moment they decide to perform HE until the point at which they deliver the results.

CHAPTER 3. EXPERT INTERVIEWS

One of the central objectives of this study is to develop a step-by-step protocol to facilitate the process of conducting HE for novices, thereby improving the quality of results produced by novice evaluators. To accomplish this aim, we decided to interview a number of usability experts to discover the major issues that make HE hard to implement and sometimes lead to unsatisfactory results.

The interviews we conducted were divided into two parts. The first part was concerned with HE in general. In this part, we wanted to identify the issues of HE as a method, independent of any particular set of heuristics. In the second part, we wanted to specifically examine Nielsen's heuristics. This was chosen because it is considered the most popular set of heuristics, with 3,922 citations on Google Scholar compared with 192 citations for Gerhardt-Powals' principles and 205 citations for Tognazzini principles, and because it is strongly related to the method; for many usability practitioners, they are interchangeable.

As discussed above, there are no clear criteria for assessing what constitutes an expert in HCI, and this lack of definition is a pervasive issue throughout a wide range of disciplines. The literature review found only one paper that addressed this issue for usability experts (Botella et al., 2014), which offered a novice, beginner, intermediate, senior and expert classification. However, these classifications are, to some degree, overly strict as they are based on the idea of deliberate practice. As highlighted previously, deliberate practice been criticized for inaccuracy; working a specified number of hours does not appear to fully explain how someone becomes an expert and other factors, including IQ, starting age, and personality, can play a role in determining expert qualification (Hambrick et al., 2014). On this basis, and since 10 years' experience or 10,000 hours of practice is hard to quantify and measure, we followed alternative approaches to defining experts, in particular drawing on the method used by de Lima Salgado et al. (2016), which defined a usability expert as someone with at least 4 years of experience in the field (pp. 2931–2946).

There are a number of reasons why we chose to interview usability experts rather than simply interviewing novices—the main concern of this work—to understand the issues that novice evaluators face. First, interviewing novices would not have been particularly useful in identifying as many issues as possible, as they may not necessarily be aware of all the problems they may cause while conducting HE. Second, interviewing novices would not have helped to find solutions to these issues, as novices are not sufficiently equipped to provide such solutions. Third, interviewing novices would have been redundant, as all experts have been novices at some point, so can relate to what novices are going through now while also knowing how they overcame similar obstacles in the past. Interviewing experts can thus help identify as many issues as possible, provide solutions to them, and also offer suggestions on how to further improve the method.

Interviews should typically be conducted until data saturation is achieved, meaning no new insights are being added, and Guest et al. (2006) suggest that 12 interviews are usually needed to reach saturation. When considering the number of experts to interview, we planned to conduct 10–15 interviews and add more interviews if we did not find that we had already reached saturation, although the number of interviews would also depend on the availability of usability experts.

Prior to the interviews, we devised three main criteria for screening potential participants: interviewees must have at least 4 years of experience in the field; they must have previously conducted HE at least three times; and they must be aware of Nielsen's 10 heuristics. We interviewed 15 usability experts who matched these criteria. While interviews were semi-structured, meaning that there was a list of major questions to be asked, we allowed participants to dig deeply into areas they wished to talk about in more detail. In line with this, we did not interrupt participants, even when we felt that they were starting to digress, as this additional interaction can yield useful insights.

The major questions asked in the first part of interviews were related to: describing their overall experience with HE, difficulties they had faced, and how they overcame them; detailing their processes of conducting HE; discussing how they documented usability problems, mapped usability problems to heuristics and estimated the severity of usability problems; and suggesting how to improve understandability and applicability of HE for novices. For the second part of interviews, the major questions asked were related to: describing their overall experience with Nielsen's 10 heuristics; outlining difficulties and issues they faced when using them; explaining each heuristic in detail, giving examples for each heuristic; and explaining the significance of particular heuristics and the consequences of ignoring them. For a full list of interview questions, see

Appendix A. During both parts, we asked additional questions based on participant responses (Figure 6).

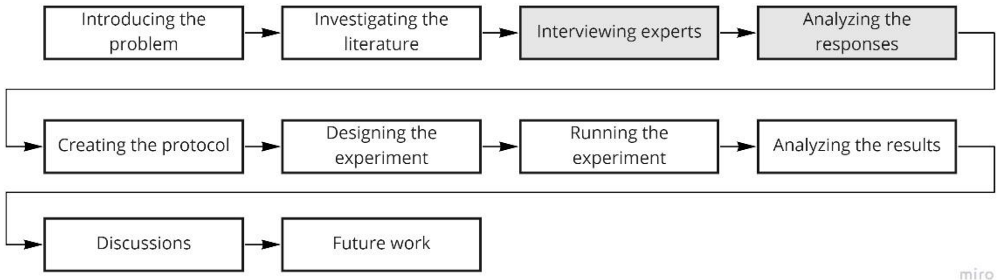

Figure 6. *Thesis organization (Chapter 3)*

3.1. Participants

As outlined above, the main criteria for participants were that they had at least 4 years of experience, had previously conducted HE at least three times, and were familiar with Nielsen's heuristics. In addition, we attempted to recruit a relatively diverse population, aiming to recruit experts of both genders, different years of experience, different educational backgrounds, different work histories, different positions (e.g., researchers, designers, and managers), and with different educational degrees (e.g., BSc, MSc, and PhD), particularly as HCI is a multidisciplinary field with relevance in academic and industry contexts.

Requiring participants to have at least 4 years of experience provided us with some guarantee that the experts being interviewed could provide us with some insights about issues with HE and usability-related issues in general. Ensuring that participants had conducted HE at least three times and were familiar with Nielsen's heuristics provided some assurance that they were fully aware of the method and understood the main set of heuristics in the field. It also meant that any solutions offered in relation to HE would address Nielsen's heuristics.

Interviewing a diverse population was deemed important for helping identify different problems and look at the matter from different angles. Participants from academia could provide insights into not only how they conduct HE but also how they teach it to students, offering deeper understanding of HE's underlying knowledge and the issues with which their students struggle. Participants from industry, on the other hand, may have more practical experience with HE, offering insights into the techniques they use while performing HE. By interviewing experts from both academia and industry, we could expect to cover both knowledge and techniques, which are equally important in improving HE performance. Using experts with different educational and organizational backgrounds is also beneficial for including people who think and act differently, particularly in relation to what requires more attention during evaluation.

We measured experience in terms of the time the participant entered the field of HCI/UX, marked by when they started either studying HCI or working in an HCI/UX position. For example, a person who studied for a master's degree in HCI for 2 years and worked in a UX position for 3 years would be regarded as having a total experience of 5 years. Academic background was defined by the area of college degree. For example, a participant with a college degree in psychology followed by a master's degree in HCI would be considered to be in a psychology-related field. Position was defined by the current job title. Highest degree earned was measured by degrees already completed, rather than any being currently pursued. For example, a PhD student's highest degree earned would be a master's degree, if they have one. College professors, instructors, or PhD students were considered academic, even if they have another job outside academia, while other participants were considered to be from industry.

Out of the total of 15 participants, seven were from academia and eight were from industry; 10 were male and five were female; years of experience ranged from 4 years to 15 years; educational backgrounds were wide-ranging, with eight from computing-related fields, three from engineering-related fields, two from business-related fields, one from an arts-related field, and one from a psychology-related field; three participants held PhD degrees, nine held master's degrees, and three held bachelor's degrees (Table 1; Figure 7-Figure 12).

Table 1. *Participants*

Partic-ipant	*Gender*	*Years of experi-ence*	*Current Position*	*Highest Degree Earned*	*Type of Work*	*Educational Background*
P1	Male	4	PhD student	MSc	Academia	Computing Related Fields
P2	Male	7	PhD student	MSc	Academia	Computing Related Fields
P3	Female	7	PhD student	MSc	Academia	Computing Related Fields

P4	Male	6	PhD student	MSc	Academia	Computing Related Fields
P5	Male	5	UX Researcher	MSc	Industry	Computing Related Fields
P6	Male	5	UX Researcher	MSc	Industry	Engineering Related Fields
P7	Female	10	UX Designer	BSc	Industry	Arts Related Fields
P8	Male	4	UX Manager	MSc	Industry	Engineering Related Fields
P9	Female	4	UX Analyst	BSc	Industry	Computing Related Fields
P10	Female	9	Assistant Professor	PhD	Academia	Computing Related Fields
P11	Male	8	UX Consultant	BSc	Industry	Business Related Fields
P12	Male	4	UX Researcher	MSc	Industry	Psychology Related Fields
P13	Male	5	Assistant Professor	PhD	Academia	Computing Related Fields
P14	Female	15	Assistant Professor	PhD	Academia	Engineering Related Fields

P15	Male	7	UX Designer	MSc	Industry	Business Related Fields

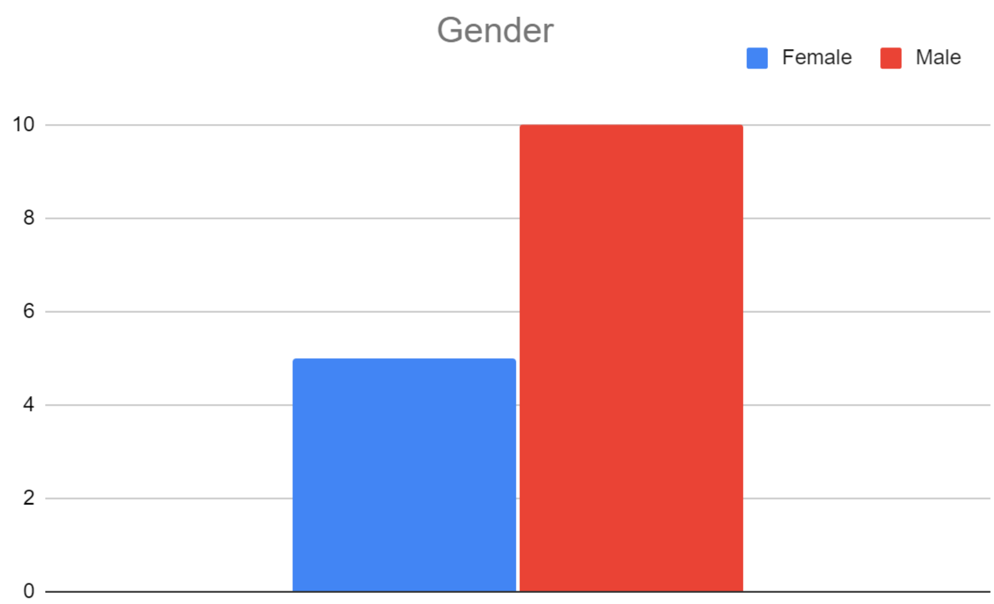

Figure 7. *Gender*

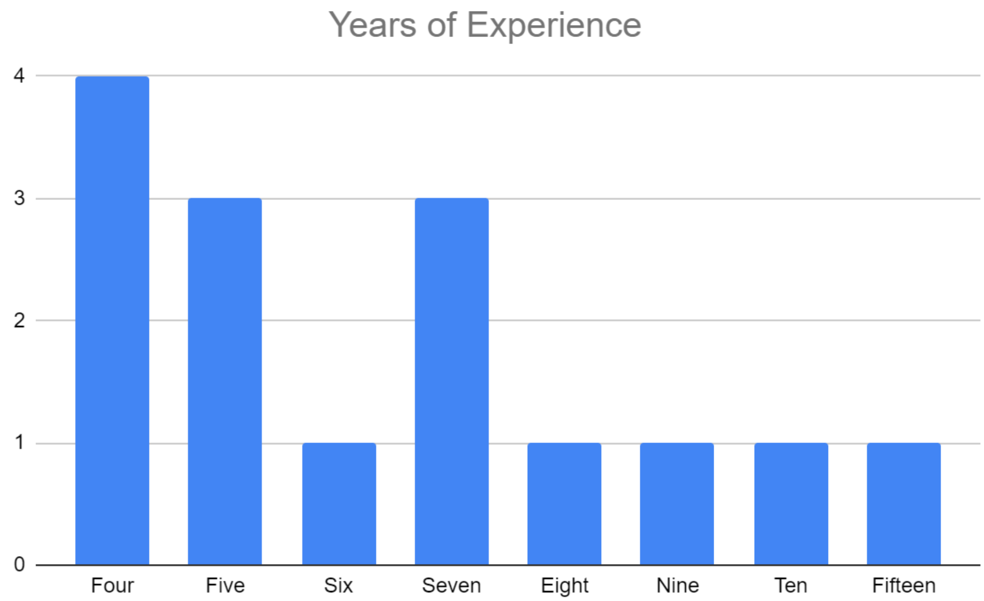

Figure 8. *Years of experience*

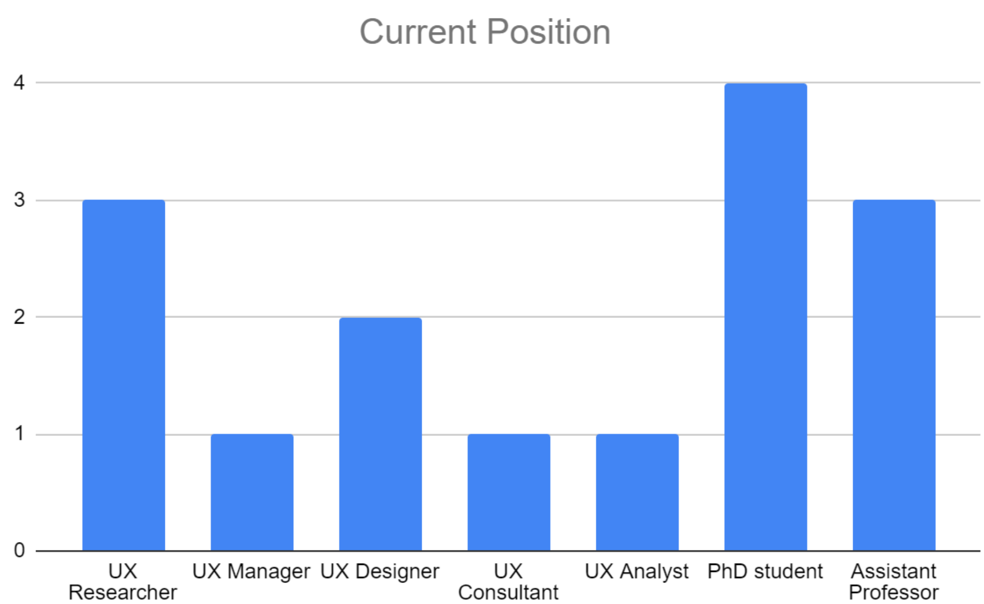

Figure 9. *Current position*

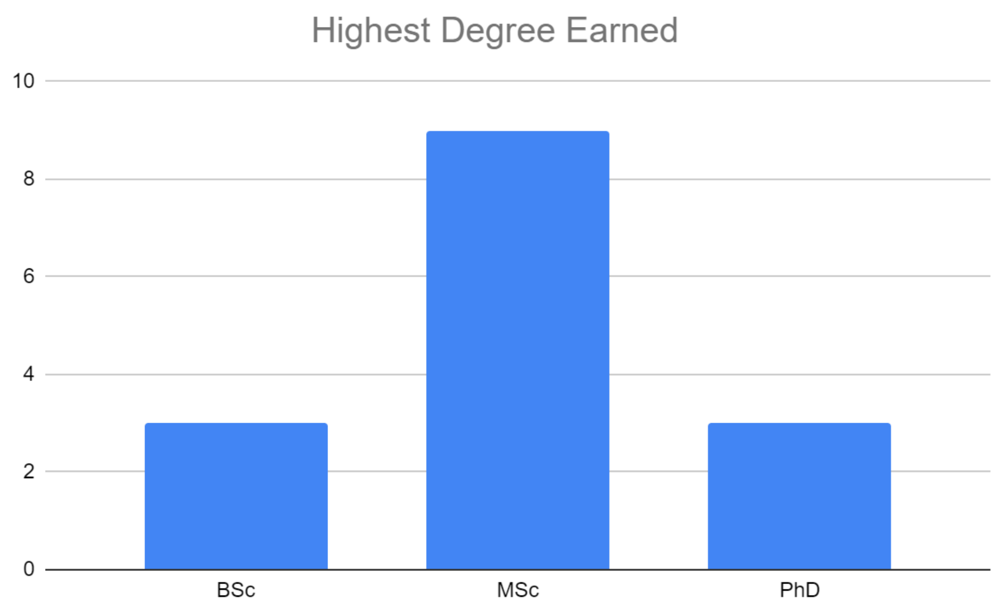

Figure 10. *Highest degree earned*

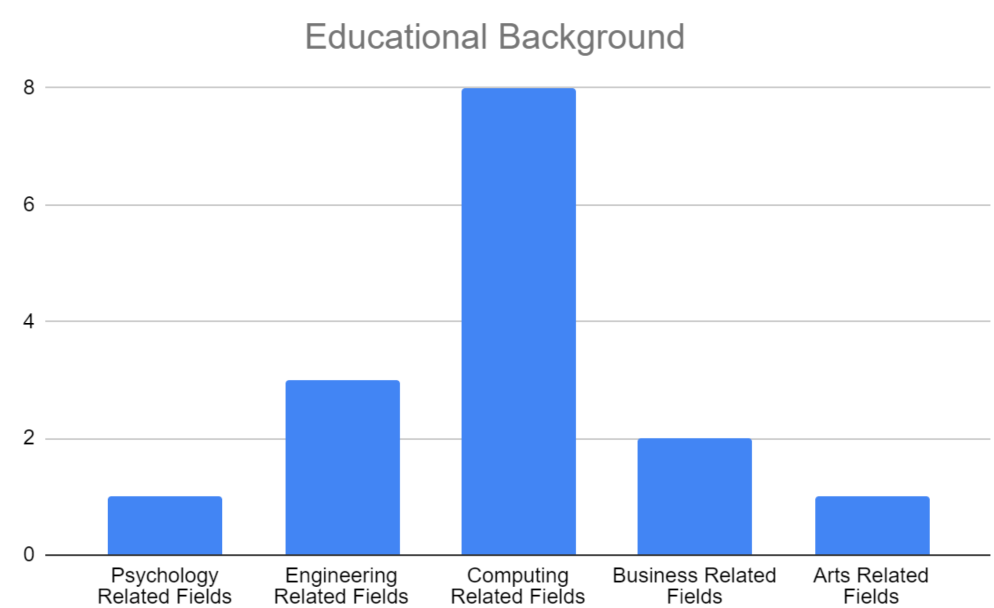

Figure 11. *Educational background*

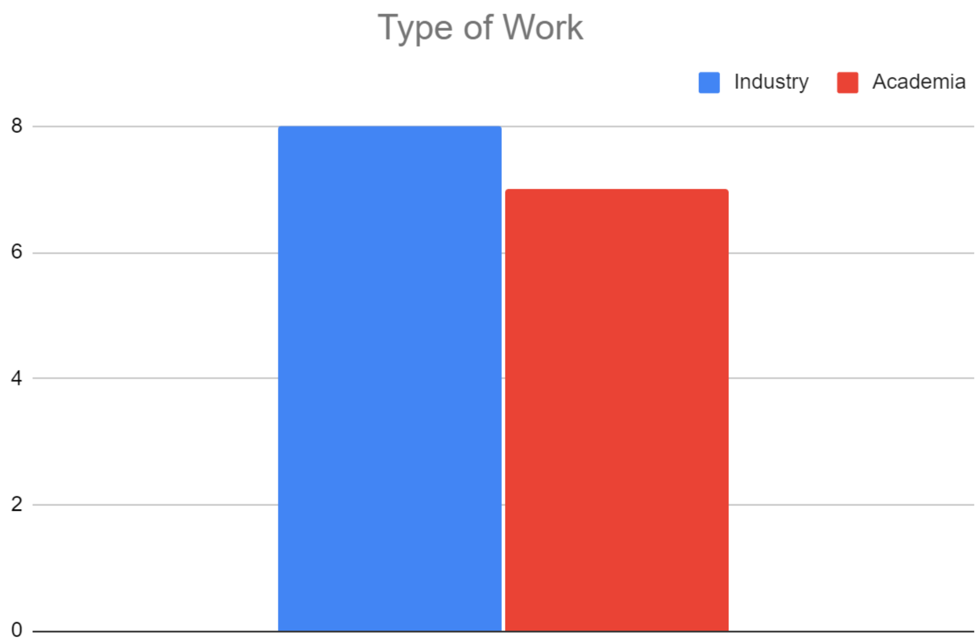

Figure 12. *Type of work*

3.2. Recruiting

After establishing the screening criteria, devising the interview questions (Appendix A), developing recruiting material (Appendix B), and obtaining IRB approval (Appendix C), we began recruiting participants using snowball sampling. We started by contacting people we knew from academia and industry who satisfied our criteria, particularly those who we thought might also know other participants who satisfied the criteria. We contacted them by three means: talking to them in person, sending them emails, or calling them on the phone.

We introduced ourselves, explained the objectives of the research we were undertaking, described the screening criteria, asked if they met those criteria and were willing to participate in the research, explained the settings and the duration of the interviews, and provided contact information for the main researcher and the faculty advisor. We finished by asking if they knew other participants who met the criteria and might be interested in participating. When referred to another participant, we would repeat the process.

3.3. Settings

Following recruitment, when participants had agreed to participate in the study, we set up times and dates for interviews. 24 hours before the meeting, a reminder was sent to the participant to ensure their willingness to participate.

The interviews were held either in-person or via conference call. Ideally, we wanted to meet with all participants in person to facilitate smoother interaction, but transportation time and tight schedules made this impossible for some participants. Only six interviews were held in person, and the other nine were conducted via conference call (Table 2; Figure 13).

Table 2. *Setting*

Participant	*Type of Interview*
P1	Face to Face
P2	Face to Face
P3	Conference Call
P4	Conference Call
P5	Conference Call
P6	Conference Call
P7	Conference Call
P8	Face to Face

P9	Conference Call
P10	Conference Call
P11	Face to Face
P12	Face to Face
P13	Conference Call
P14	Conference Call
P15	Face to Face

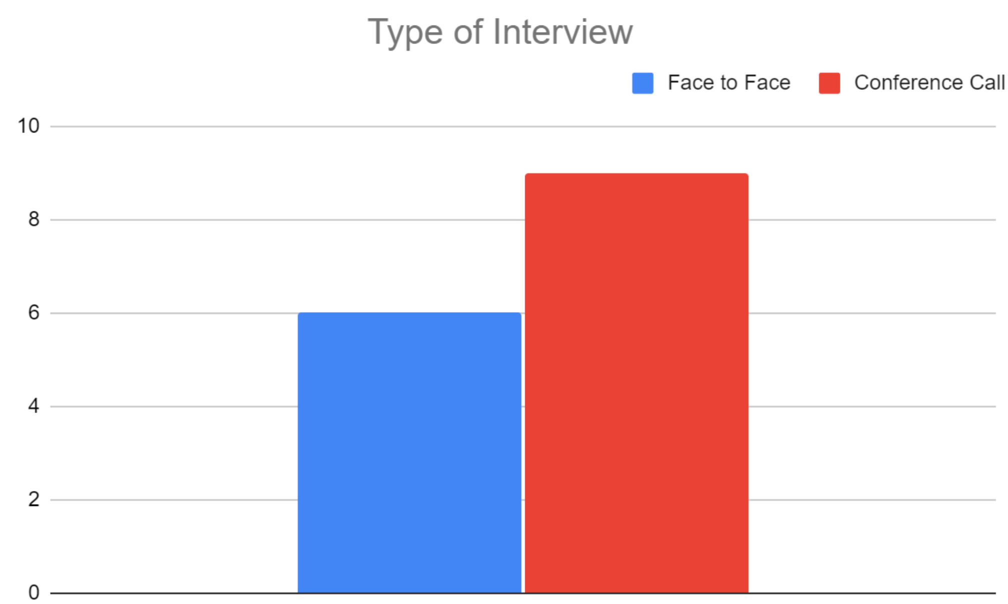

Figure 13. *Type of interview*

Before an interview began, a consent form was read to the participant (Appendix D), who was then asked to give verbal consent, as such information is unidentifiable. Each entire interview was recorded on the researcher's phone using an audio-recording application. Each participant was given the option of keeping a copy of the consent form if interviewed in person or having it emailed to them if they participated via conference call.

Interview duration ranged from 60 to 120 minutes. In some interviews, participants were very enthusiastic and wanted to talk more about the topic, so some took more than 2 hours; three interviews lasted for about 3 hours each. Tight schedules meant that two participants preferred to divide an interview into two interviews completed on separate days. The interviews took place in the interval from May to July 2019, with seven in May, five in June, and three in July.

3.4. Analysis

We used thematic analysis, as described in Blandford et al. (2016), to analyze the interviews because it is more flexible than other qualitative methods and allows for both induction and deduction. We analyzed each part of the interviews separately. We approached the first part in a more exploratory and inductive way, seeking to determine the main ways in which experts perform HE while keeping in mind three abstract themes of understanding, inspecting, and documenting. We approached the second part in a more focused, deductive way informed by the outcomes of the first part.

The six steps of thematic analysis undertaken were as follows:

1. *Familiarization with the data.* During each interview, the interviewer took careful notes. After each interview, we listened to the recording and transcribed the interview. After all the interviews were completed, we

listened to the recordings again and compared them with the data to ensure that nothing was missing.

2. *Generating initial codes.* We separated the two parts of each interview and, starting with the first part related to HE, began to put all related excerpts together in codes.
3. *Searching for themes.* We looked at all the codes and used them to generate themes.
4. *Reviewing themes.* We looked at the different themes to check their accuracy.
5. *Defining and naming themes.* We looked at the different themes and labeled them. We then went through the same procedure with the second part of the analysis, informed by the themes generated during the first part.
6. *Documentation.* We documented the results of both parts.

3.5. First part results

During the first part of the interviews, we focused on HE in general. We wanted to identify its issues and know how to solve them and how to improve the method and make it more accessible for novice evaluators. Thus, this part was independent of any set of heuristics, only exploring HE as a method. We approached this part in a more explorative, inductive way, with a focus on three general areas of interest: how to understand usability heuristics, how to perform an inspection, and how to document results.

3.5.1. Participants' experience with HE

HCI is a multidisciplinary field in which some practitioners have received a formal educational qualification in HCI, while others have learned it either through a self-taught process or as part of a formal degree in another discipline. This also applies to HE, one of the evaluation methods within HCI; some practitioners have intensive training while others do not. Interviews revealed that even some of the participants with formal degrees in HCI did not consider the HE training they received during their formal study adequate in preparing them to use it.

Learning HE

Among the participants we interviewed, seven had no formal education in HCI, while the other eight had at least one degree in HCI. Only one out of the eight participants with formal HCI education thought that the training they received during their study was sufficient in equipping them to conduct HE in the real world. The remaining participants either did not mention the role of the training they received or did not consider it particularly effective. Some participants indicated that they learned how to use it properly when they started to use it frequently in their work, subsequently learning from their mistakes:

> I took a class on evaluation methods, and while we had to learn HE, we didn't do it extensively until I started to work. As I did it more and more, I began to learn how to do it better and be more detailed and nuanced. (P6)

Others reported that, as they read more about HCI and its different methods and concepts, they began to understand the depth of the method and became better at implementing it: “Some books really helped me in the process, books like *Intuitive Design* and *Designed for Use*. Reading Norman’s book, along with other books, improved my understanding of HE” (P15). Specifically, some mentioned that teaching HE was a way for them to improve their use of HE, which served as a learning method: “Teaching was also a learning method for me. The more I teach it and get students to ask me questions, the more I learn” (P14).

Participants with no formal education had to learn HE independently, although their experience was not dramatically different from those who had formal education. The main difference for most of these participants is that they did not know where to start when asked to perform HE. All of them stated that, since there is no single resource that explains HE comprehensively, they had to consult multiple resources such as online classes or books or learn from a more experienced practitioner:

> I took online classes on Coursera and Udemy, and they helped a lot. (P9)
>
> I didn’t find a complete guide for HE. If you type HE into Google, you would find articles that briefly explain the heuristics, so you have to do proper research and read multiple articles and sites to understand it clearly. (P11)
>
> The company hired a new employee who has a master’s in HCI and some experience. She explained it thoroughly to us. (P12)

Despite some agreement that learning HE is difficult, participants expressed a variety of opinions on the reasons for this difficulty. Some participants thought that this it is normal because learning anything can be tough initially and that, with practice and more reading, they would learn how to use it better. Others believed that the difficulty reflects an underlying issue with the available materials on HE, highlighting the need for improvement.

One participant suggested that one of the reasons for the relative ineffectiveness of HE training in formal education is that the focus is usually on other methods, namely usability testing:

> From my observation, usability testing is written about and taught far more than HE or CW. During my master's study, the focus was more on usability testing than other evaluation methods. It is understandable, as, in usability testing, you are getting the feedback directly from the real potential users. (P4)

When to use HE

HE can be applied before launching a system to ensure that the system is usable and has no major issues, but it can also be used as a review after launching a system to check and correct for remaining issues. Participants' responses suggest that, while both purposes are common, there is a difference between the two. Users seeking the former purpose perform a more comprehensive review than for the latter and those seeking the latter purpose usually have an idea on particular items to evaluate rather than evaluating the whole system:

> I usually combine HE with analytics because it helps me know what the pain points are and evaluate those. (P6)

> If we have analytical data, we focus more on pages with which users struggle the most. (P9)

Which heuristics to use

As implied by the name of the method, HE is performed by using a set of guidelines or heuristics to evaluate a certain system. As outlined above, there are plenty of usability heuristics available from which HCI/UX practitioners can choose. While all the participants interviewed stated that they use and teach Nielsen's heuristics, these results could be biased because one of the items in the screening criteria we used before the interviews was familiarity with Nielsen's heuristics. Still, there could be multiple reasons for such a preference. First, Nielsen's heuristics is the standard in the field, with virtually everyone familiar with it, including evaluators, developers, and even clients:

> We use Nielsen's. It is the default that everybody in the industry uses at this point. (P2)

> I use and teach Nielsen's heuristics, the standard in the field to the extent that it has become known to people outside the field. (P3)

Popularity is not the only reason participants use and teach Nielsen's heuristics. They are relatively brief compared to other available heuristics, and their mere briefness is another factor in why some participants prefer them over other heuristics: "I have checked other heuristics before, but the set I prefer is

Nielsen's heuristics, not necessarily because I think it is the best, but it is short and does the job reasonably well" (P10).

Clarity and understandability also play a role. Some participants stated that, compared to other heuristics they had read, Nielsen's heuristics was the easiest set to understand:

> Nielsen's heuristics is the best of the two sets of heuristics I have used; it is easier to understand. (P4)
>
> I think the reason Nielsen's heuristics are so popular is because they are easier to understand and remember than other heuristics. (P9)

The general popularity of Nielsen's heuristics makes it the first choice when HE is taught in schools or in UX/HCI-related courses and programs, adding to its overall popularity and making it the first choice for use by evaluators. One participant highlighted that this gives it a sense of familiarity: "When I first went to the UX design boot camp, that was the one we were introduced to and used the most, so I became much more familiar with using it" (P7).

Nielsen's heuristics, like all sets of available usability heuristics, have limitations. For that reason, some participants stated that they mix and match between heuristics to overcome the limitations of each set. Most pointed out that they started using only one set of usability heuristics but that, as they gained experience, they began to combine different heuristics:

> I used to use Nielsen's heuristics alone, but as I gained more experience, I began to use multiple heuristics at the same time, like Dr. Pete's 25 points, and eight intuitive heuristics. (P5)

> There are Nielsen's heuristics, there are Shneiderman heuristics, there are Tognazzini heuristics, I teach them all to my students. . . . I personally use them all, I mix them. . . . I would recommend anyone to start with one of them, then mix them when they feel the first one is really well-understood. (P13)

3.5.2. Improving HE understandability

Given the importance of understanding from the outset, we focused on activities that our participants had themselves undertaken to understand HE or to explain HE to either their students or their colleagues. While several activities were highlighted by participants, with varying levels of effort demanded, there was no agreement on a single best approach to improving understandability.

Reading

A number of participants recommended reading specific books. One stated that their workplace required employees to read certain books as part of training their new team members: "We then ask them also to read some books. Typically the books are *Practical User Research*, *The Design of Everyday Things*, and *The Elements of UX*" (P8).

Another participant suggested that reading would help novices think beyond the method and deepen their understanding of HCI and usability: "I recommend that every novice read Norman’s book from cover to cover. It will get you thinking about the bigger picture” (P15).

Online courses

Participants recommended taking online courses, specifically for helping those who have little or only minimum training in HCI. These identified several platforms such as Udemy, Coursera, and Lynda, stating that they offer good courses on the subject: “Novices can benefit from MOOCs in trying to understand HE. I have checked Coursera; they have two to three good courses” (P1).

Action model and usability

The first goal of HE is to ensure that users can perform their tasks on the system and can perform them easily, which gives the evaluator some context and helps with focus during the evaluation. Three participants stated that evaluators should read and understand how other evaluators approach HE. Two of them suggested Norman’s model for seven stages of action, while the other participant proposed a three-step model that consists of understanding, executing, and evaluating: “Norman’s seven stages of action represent an incredible model that could be used along with usability heuristics, especially for novices, to give context” (P13).

However, one participant highlighted that understanding the model alone is sufficient. When mapped to the usability heuristics, the model does clarify the heuristics themselves and emphasize their significance and the role they play in helping the user accomplish a goal: “Understanding the model and its relationship to each heuristic is going to help students in visualizing the whole process. It is going to explain why we do a heuristic evaluation and how each heuristic matter" (P13).

Another participant similarly suggested to read and understand usability and its components, as presented by Nielsen. Since usability is the main objective of not only HE but also HCI in general, this suggestion should be kept in mind when conducting the evaluation:

> Nielsen has one of the best definitions for usability in which he explains its components. . . . I first explain usability and its components, then I explain the 10 heuristics. . . . We do a class activity where we discuss how each heuristic affects each of the usability components. (P10)

Examples

As examples can make the concepts of each heuristic clearer and more understandable, most participants emphasized the importance of using examples. Participants raised a number of points to be considered in relation to using examples. First, they suggested that examples should not be limited to the interface alone, as providing examples from everyday events helps deepen an evaluator's understanding: "The best way to explain it to novices is to use lots of examples, most importantly examples from outside the interface, so they understand the power of the heuristics" (P5).

Second, they proposed using both good and bad examples, as good examples can help the evaluator to know what should be done while bad examples can help them to know what to avoid:

> I want to see good examples as well as bad examples. . . . Bad examples are what I always see when I read about heuristics. Bad examples are extremely helpful in making me realize what I should avoid, but just seeing bad examples will not give me a reference as to what I should be aiming at. (P6)

Third, most participants suggested that many examples should be given, as this allows the evaluator to see that each heuristic does not deal only with a certain element or certain situation, but with multiple things. One participant proposed using at least three examples, while another suggested using at least five examples:

> I give at least five different examples to kind of break that mental model of "Oh, it's only about the alert box." (P2)

> There is no fixed number of how many examples I give, but definitely more than three for each heuristic. (P10)

Improving heuristics

Participants offered suggestions for improving the heuristics themselves, mentioning several issues related to how most usability heuristics are presented and described to the user. First, they highlighted that most usability heuristics are abstract and may contain multiple interrelated concepts, or in some cases, consist of more than one heuristic. Some participants suggested clearly explaining the different concepts or ideas related to each heuristic:

> One of the issues that I find in many heuristics is that a single heuristic contains multiple, equally important concepts. Brevity is an understandable goal, but since they are not laid out clearly in the description, students miss some of them. Take visibility of system status, for example, that has about three concepts. The user should know his or her place in the system, that is one aspect; they should also have some feedback about the progress they are making, that is a second aspect; they should know the state or the affordance of things they are interacting with, that is a third. (P10)

> There is a problem specifically with Nielsen's heuristics as well as in other heuristics; putting similar ideas into one single heuristic. This might lead to one point being focused on and others not much. Consistency and standards, help and documentation are examples of this. (P14)

A second approach mentioned by one participant for improving the understandability of usability heuristics is to explain the components of each heuristic and identify what makes the heuristic successful:

> The way I like to explain it to students, and the way I learned it is by looking at what makes each heuristic work. For example, when we say "good systems provide good documentation", what does that mean? It

> means that documentation should cover the material that the user needs, is accessible, easy to understand, and so on. (P3)

A third participant also suggested that to further clarify the heuristics, the elements with which the heuristics deal should be described and explained. If a heuristic is usually found in some position or interface locations, such locations should be listed. However, this approach can introduce risks because the evaluator may look only for those elements or places and ignore other aspects that could potentially be affected by the heuristic:

> I also like to see where the heuristics manifest themselves, like matching the system to the real world; it is related to icons, text, etc. . . . Back to the manifestation point, there is one issue that it could cause. You can't possibly list everything related to a heuristic, and this should be clearly explained, otherwise they will ignore important things thinking they are not related to the heuristic. If done right, it will be very effective. (P3)

Since heuristics labels convey their meaning, it is important to have meaningful and easy-to-understand labels. Two interview participants pointed this out, adding that labels should not only convey meaning but also be easy to remember. They suggested that a good label would make it easy to understand and remember the heuristic:

> The first time you should understand it very well after reading the descriptions and the examples and so on, but the next time you don't want

> to repeat the same process again, you just want to look at the label to recall what the heuristic was about. Clear labeling helps with this. (P1)

> Most labels are not clear and don't reflect the true meaning of the heuristics. They often should be relabeled to make them easier to remember. I don't want to go back every time to read the description to remember what the heuristic was about. (P11)

Sticking to heuristics is not always necessary and, in some cases, it is not even desirable. However, when heuristics are explained, this point is often omitted. Some participants suggested that, to improve the understanding of heuristics, situations in which sticking to heuristics are not necessary or even wanted should be explained:

> At the beginning, I used to be a little bit shallow, writing problems that were not necessarily problems. For example, if a task requires six to seven steps to be done, I would consider it inefficient without thinking whether it could be further simplified or not. Heuristics descriptions don't include such explanations. (P6)

> There are downsides to HE in general when you implement them as a novice. You stick literally to each heuristic. If it says you have to be consistent, then even when violating consistency is necessary, you would report it as an issue. (P8)

Although each heuristic is important and significant for a reason, the underlying reasons why each heuristic is important is often not clearly explained when heuristics are presented. While the importance of a heuristic sometimes could be obvious, this is not always the case. To an expert, it might be clear why each heuristic should be followed, but it might not be as obvious to novices. Some participants affirmed the value of explaining the importance of heuristics when explaining it to novices:

> From my experience with both heuristics I used, none of them actually explained in detail why I should be following these rules. I mean, what is likely to happen? Are there empirical results to support their claims? Are they based on psychological studies? None of that is provided. (P4)

> What I like about the usability.gov list is that it explains the importance of each guideline and gives it a rate. This would be extremely beneficial if this was done to a shorter list like Nielsen's heuristics. What keeps me from using usability.gov list is because it is an extremely long list. (P12)

3.5.3. Improving inspection

Inspection, the act of looking at a system to detect usability issues, is the central activity while performing HE. Similar to the understanding phase, there are many ways in which this phase could be improved to make evaluation easier, more effective and more accurate, as implicit in the suggestions of participants on how to accurately perform evaluation.

The first step in conducting an inspection is to decide exactly how to begin undertaking the evaluation, whether to start an evaluation without having a specific aim, to randomly check the system, to choose certain elements or aspects of the system for evaluation, or to select certain tasks or scenarios to follow and inspect. If other HCI activities have already been performed, the process of system evaluation may be made easier.

Combining HE with other activities

Several participants state that they begin an evaluation by doing other activities, rather than starting the evaluation immediately. One participant indicated that they start by conducting a competitive analysis to identify what the standards are and at what they should be looking, supporting the overall evaluation:

> Most of the time we do competitive analysis before doing HE, and write a comprehensive report about similar products, locally and internationally. . . . Doing the competitive analysis helps us know what to focus on during HE, and we pick one of them to be the benchmark. (P8)

A number of participants mentioned conducting user research before starting an evaluation. While the goal of user research is to understand the target audience of the system, making it easier to keep them in mind during evaluation, the ways by which participants benefit from the results of the user research differ. One participant stated that they transform the user research data into personas to guide them through the whole process, particularly in choosing which aspects to evaluate:

> What I do first is to gather information about target users. From there I create user personas, usually three to five personas. . . . I choose the paths that I am going to evaluate based on the personas. (P1)

Another participant suggested that user research results can be used to create personas leading to a user journey, helping to decide which parts of the system should be evaluated and how the evaluation should be approached:

> We also do user research before HE to understand our users better. The user research informs us about our user goals. We create personas and, based on them, we create a user journey. . . . We almost only evaluate the pages with which the users interact the most. The user journey helps with that, and we look at them from the point of view of each persona. (P8)

Some other participants who mentioned doing user research utilized the results in a more informal way. Rather than creating personas and user journeys, they use the results to provide a general idea about users to support the evaluation:

> We ask the client to provide us with some data about their users and we take it from there. . . . We read through the data and the analytics to give us a sense of what we should evaluate. (P9)

Dividing the system for evaluation

There are multiple ways in which a system can be inspected. These include: a page-by-page approach; going through every single page on the system'; inspecting only some aspects of the system; and inspecting only certain tasks. All participants agreed that the ideal way to perform evaluation is to choose certain user goals and to evaluate either those goals or related flows. However, there are exceptions to this approach, as sometimes clients specifically ask that the whole system be inspected page by page or ask for only a small aspect of the system to be evaluated:

> You have to write out a happy path, write out what you think the user wants to do or is going to do, perhaps the top five actions that they are going to do. Break down those tasks and then break them down into steps then go do them. (P2)

> Heuristic evaluation can be both super complex and super granular. I'm talking about assessing every single button. Or they can be more general. That depends on what the client needs. . . . If we are looking to do a kind of breakdown of the style of the website, we can be a little bit more detailed. Or say we were trying to focus on the flow of this website, what is the flow of the user engaging with it, we may not necessarily focus on every UI detail. (P7)

One participant emphasized that evaluation should be based on user goals and not tasks; the difference is that a goal is what the user wishes to accomplish with the system, while a task is something that a user needs to accomplish a goal:

> By goals, I mean choose specific goals rather than tasks. An example to show the difference would be the simple task "add a new post", whose goal would be "write a post and publish it and share it with friends" so the end goal would be to share it with my friends. (P11)

There are a number of reasons to focus on specific user goals and not to perform a page-by-page evaluation: first, the page-by-page approach is very time and effort consuming and likely to be specifically overwhelming for novices; second, sometimes the approach is unnecessary because users are most likely not going to visit every page; third, determining specific goals provides a clear view of steps to take; and, finally, an evaluator using a page-by-page approach might miss some usability problems, because these problems often occur only in transitions between tasks. Several participants specifically affirmed these points:

> You can't find most errors if you are working page by page. Errors most of the time happen in multiple steps task like filling in a form or making a transaction. (P1)

> I don't do page-by-page examination because the users don't usually visit all the pages. (P5)

> I prefer doing it by user goals because then while you're going through the product, it kind of gives you a logical path to follow. (P7)

> Evaluating every page is extremely difficult and requires a lot of time and effort. I ask my students to pick three to four flows and do them. (P10)

Although all participants agreed that the system should be evaluated based on specific user goals, some participants pointed out that there are certain pages in the system, including the homepage, contact us, about us, and FAQ pages, that should be evaluated individually:

> I would say you should take a look at certain pages. Help pages, like FAQ, are important and if they are not working or have some issues the user might struggle. . . . The homepage, or any landing page, is important. It is the first thing that the user interacts with. (P3)

> There are key pages that should be examined: homepage, contact us, about us. (P5)

Number of sessions

A system can be evaluated with a single session or in multiple sessions. Participants' responses to this aspect varied, with some participants performing evaluation in one session and others preferring to use multiple sessions. However, even participants who preferred the single session approach highlighted that novices should not take this approach, especially if the system is large or complex, as trying to evaluate the system all at once could overwhelm novices and lead to mistakes. The only exceptions to this are if the system is small or if the evaluator is asked to evaluate only a small portion of the system. The deadline for evaluation can be another factor as, if quick submission is required, the evaluator may need to complete the evaluation in one session:

> I recommend that my students give every goal enough time. Like in the morning do one, afternoon do another one, you get the idea. . . . They should only focus on the goal at hand. I don't want them to be thinking about other things on the system. (P3)

> It depends, sometimes I do it in one set, sometimes in multiple sets. . . . How much time do I have to finish it? How big is the product? If the product consists of a few pages, then I do it in one set. If I have to deliver the report in 1 to 2 days, then in one set. (P10)

> I am more comfortable doing it all at one time. Once I am in the mood to evaluate, I will try to finish the evaluation. If I am distracted, it is hard for me to get back in the mood. . . . From my observation, my colleagues take their time, they divide the system and do it in more than one time. (P12)

The duration of the evaluation

How long the evaluation will take is yet another factor that can affect evaluation quality. Again, there seems to be no definitive answer as to how long an evaluation should take, and there are multiple factors at play when it comes to allocating time for the evaluation, including system size, whether an entire system or only a portion is involved, required level of detail, system complexity, time constraints on report submittal, evaluator experience, and whether other activities such as user research or competitive analysis are going to take place alongside the evaluation. One participant stated: "Again, how much time do I have to send the report? How big is the product? If the product is small, 2 to 3 hours is enough. If it has many services, then it is hard to tell" (P10).

Due to the complexity of such factors, most participants were not able to give an approximate time for evaluation completion. However, some gave an estimate for how long on average their own evaluations take, ranging from 5 hours to 3 months:

> On average the evaluation itself takes me about 5 to 7 hours. (P12)
>
> It depends on the site or the app, usually it takes about 1 or 2 days to prepare, 1 to 2 days to evaluate, and 2 to 3 days to prepare the report. (P1)
>
> The whole process of doing HE plus the user research and the competitive analysis takes up to 3 months. (P8)

This shows that the variety of factors involved makes it very difficult to determine how long an evaluation should take. Still, two participants recommended that, no matter how long an evaluation takes in total, evaluators should take breaks every 30–45 minutes to regain focus and perform a proper evaluation:

> There is a technique I use when I study, called the pomodoro technique; it basically says, study for about half an hour, take a break for 10 minutes, then go back to study for about half an hour and so on. . . . I use the pomodoro technique when I do HE as well. (P1)
>
> Let’s say 30 or 45 minutes of an evaluation is a reasonable time that allows you to be very focused. . . . After more than 45 minutes, an hour maximum, you would lose focus. Since we are talking about beginners, that’s really important for them. (P9)

Suggestions for improving an evaluation

Participants offered suggestions for a number of ways of accurately detecting as many usability issues as possible. One participant suggested that if the evaluation is being undertaken by a novice, it is better to conduct it in combination with other novices. Performing the evaluation as a group may help in detecting more issues because, when one evaluator misses an issue, others may find it and a group approach enables real-time discussion of issues, making it possible to eliminate any issues that have been misidentified. One participant explained: “There is no easy answer on how to make it easier for novices because it takes practice, but I would recommend working first in groups; they get together with their friends and do the evaluation” (P3).

Another participant recommended that, at the beginning of an evaluation, a novice evaluator should perform the evaluation with a more experienced evaluator to help them to gain first-hand insights into how the evaluation should be conducted: “Try to work with someone experienced in doing it, your skills will improve by doing so” (P10).

Five participants emphasized that, before doing an evaluation, the evaluator should not be overly familiar with the system, because detailed knowledge about a system could introduce bias and prevent them from approaching it like an actual user. They suggested that the evaluator should have only a general idea about the system, such as its main services and users:

> Part of the evaluation is to experience the site as the user does. If I am too familiar with the site, most likely I will overlook some of its issues. (P1)

> Before doing the evaluation, the evaluator shouldn't be overly familiar with the app, because if that is the case, they will be biased. (P4)

> I usually don't try to learn a lot about the system so I can examine it like a newcomer. . . . Newcomers are the hardest users to attract, so you should keep that in mind before performing the evaluation. (P11)

Three participants suggested that novices have a tendency to treat the usability heuristics like a checklist, which can cause problems as the evaluator may invent usability issues or exaggerate a problem simply to be able to cross off a specific heuristic from a list and they might miss a usability issue because they have already found an issue that belongs to the same heuristic:

> When students are going through the process, they think if they run across one that is like a checkbox kind of, so they are saying "Oh, there's been a violation of system status, check." A lot of students will often never go back or never think to write violations of system status multiple times, thinking "Well, I already listed one." (P2)

> It is not a must to find a usability problem for each heuristic. If there is no issue with help or documentation, don't make up one. . . . Yeah, you see this done more often by less experienced evaluators. (P7)

Two solutions were proposed to eliminate or reduce this problem. The first solution is to clearly point out to evaluators that there may be multiple issues that are related to the same heuristic and that sometimes there might be no issue related to a certain heuristic:

> Just tell them that the purpose of this list is to help you find usability issues, not to control you. The purpose is to fix the system, not to find as many problems as possible; it is not a contest. (P11)

The second solution is to conduct the evaluation twice, with the first based on user goals without using heuristics, simply trying to accomplish certain goals and noting down any usability issues found along the way, and the second based on heuristics. This minimizes the control of the heuristics on the evaluator:

> When I do the evaluation I do it twice. First, I do it naturally and write down usability issues I encounter. Second, I go through it while intentionally looking for issues. The only caveat to keep in mind is "Don't make up problems." In my opinion this is the best way to reduce biases. (P7)

Another aspect that can lead to false problems being detected or real problems being missed is the evaluator's mood while doing the evaluation. Evaluation is usually done when the evaluator is fully focused, which might cause them to be too critical and detect issues that would go unnoticed when actually using the system and may mean that they do not find issues that could occur when they are tired. One participant discussed this issue and suggested that the evaluation should be done twice, once when the evaluator is tired or about to sleep, to resemble tired users' situations, and once when they are fully focused, to identify the issues they did not notice when they were tired and to simulate alert users' situations. The participant underlined that the procedure should be done in this order, to avoid a learning effect:

> I don't do the evaluation all at once. I do it when I am very tired and about to sleep. Then I do it again first thing in the morning when I am very focused and critical. . . . Based on the mood, I usually find different issues, and see it from different angles. . . . Well, I prefer doing it first when I am tired. If I do it first when I am focused, then when I am tired I will probably be able to perform better because I've already done it before. (P11)

Having enough time to do a full evaluation is not always possible. In some cases, the evaluator must submit the results in a short period of time, making a detailed evaluation impossible. However, some participants suggested that, in such cases, evaluators can still take action to improve the quality of their evaluation. One participant proposed that a study should be done on the usability heuristics to determine which heuristics find more usability issues than others. In this way, usability heuristics could be ranked from those that tend to find the most usability issues down to those that usually find the least usability issues. In this way the evaluator could begin with those likely to find the most issues and then, if there is enough time, move on to those that find fewer usability issues:

> There is something called the pareto principle, or the 80/20 rule, where 20% is responsible for 80%. . . . If we can discover which heuristics find most of the issues, we can create a shorter list to be used when the

> evaluator has little time or a very tight deadline. . . . The evaluator starts with the shorter list, and if there is time can check the rest of the heuristics. (P12)

Two participants stated that common issues that occur recurrently on systems should be presented to evaluators, so they can start with these issues to make sure they are not present in the system under evaluation. In essence, these issues are the low hanging fruit that evaluators can find quickly, after which they can continue the evaluation until allotted time has elapsed. This approach could also be used as a review strategy after doing a full evaluation, if the evaluator has sufficient time:

> If there is something like a list of the most common problems, it would be amazing. You know, you could start by making sure those problems don't exist. It would save me some time that I could spend finding other problems. (P6)

> We store problems we see all the time in a file. We use them after the evaluation just to cover the basis and make certain that these problems aren't there. (P9)

Three participants emphasized that, during evaluation, the evaluator should not be overly concerned with the particular heuristics violated by a found usability issue; this concern should be left until the documentation stage when they begin to look at usability issues discovered, map them to the heuristics and rate their severity. During evaluation, the evaluator should only write down the usability issues found, leaving such details for the documentation stage:

> You should focus on finding as many problems as you can. The severity of the issues should come later on. (P4)

> Some students rate the severity of an issue right when they detect it. The goal of the evaluation is to find the problems, and after the evaluation is done and all problems are detected, they should then rate them and compare them to the heuristics. (P14)

3.5.4. Improving documentation

While the inspection phase is the central activity of HE, the documentation phase is arguably the most important phase, since it delivers the results of the evaluation; delivering results in a way that is not clear and understandable can undermine the whole purpose of the evaluation.

Mapping usability issues to the heuristics

Mapping usability issues found during the evaluation to the violated usability heuristics should be included in the evaluator's report. While eight participants stated that they include heuristics violated by the usability issue in the report, seven participants suggested that it is not necessary to include such heuristics, although some of these participants clarified that they would include this in certain situations. One participant stated that such information should only be included for novices, because novices may have little credibility in the eyes of the developers, so they need to support their claims by including heuristics. They suggest that, if an evaluator has already built-up trust with the receivers, including heuristics is unnecessary:

> Internally, I just give them a small report that has screenshots of the problems, some description of it, and my recommendations. But I didn't start that way, I first built some trust with developers. . . . From my experience, if you only do that at the beginning, they will say it is just your opinion. . . . At the beginning, you should talk about the heuristics, and you have to include some numbers and things like that to show them that what you are doing is legitimate. (P9)

Two participants highlighted they would only include of the report if this would help convince the receiver of the validity of the report. Including heuristics can let the receiver know that the evaluator has followed a systematic approach to producing results:

> If my client is aware of the heuristics, I will include them in the report. (P1)

> I don't think it is important to link usability issues to any specific heuristic. If you think that is going to convince the client, then do it, otherwise it is not important. (P8)

One participant stated that, while they do not explicitly include heuristics information in the report, they do include it as part of the description of the issue. Rather than explicitly indicating that the problem violated certain heuristics, they instead describe the issue and explain why it is an issue and, in that description, they mention the heuristics that the issue violated:

> In my description, I touch on the heuristics. I don't list them, what is the point of listing them? The client who is reading the report will not understand it that way. If you have found a consistency problem, in the description explain it and why not being consistent is a problem. (P15)

Three participants pointed out that they do not include heuristics at all in the report. Instead, they only use the heuristics as a guide to deeply understanding the issues and to help them make the argument and better explain the issue:

> I link to a specific heuristic merely because it helps me understand why it is a problem and strengthens my argument when I discuss the issues with the developers, but I don't care about adding it to the report. (P6)

The process of mapping usability issues to usability heuristics is not as straightforward as it may seem; this can be confusing, especially for novices. First, a usability issue can simultaneously violate multiple heuristics, which can cause confusion since it is widely thought that a usability issue should be mapped to only one heuristic:

> I think that there are severe issues based on one heuristic, but I think of the strength of the method or the way I sell it for real world uses. If I'm looking at a site and I see something, I'm going to say that's a violation of heuristic one, three, seven, and nine, or something like that. (P2)

> Some issues violate multiple heuristics; in fact, most issues violate multiple heuristics. . . . It is confusing, but not just that, it also requires some deep thinking, and for a novice it is easier to just say, "Well, the issue violated standards, even though it also violated matching between system and the real world." (P7)

Second, it is not always clear exactly which heuristic a usability issue violated. The usability issue might result from an unconventional issue, making it hard to link to a certain heuristic, or a complex issue, which must be dismantled to be clearly understood:

> And some issues are just legitimately confusing. You run into a problem and you just can't decide which heuristic it violates. I would say this is rare, but in such a case you need to break down the problem so you can know which heuristics it violates. (P7)

Most participants state that there are no easy answers for mapping usability issues to usability heuristics, implying that the process should be thought of carefully and critically. However, four participants proposed ways to facilitate the process. Three participants suggested that usability issues should be thought of in terms of a human model, considering how the usability issue affects the ability of the user to accomplish a goal. After this, the evaluator can then look at the usability heuristics and have a greater chance of discovering which of them deal with the same stage in the human model. This process may help an evaluator to accurately link usability issues to the usability heuristics they violate:

> Is the issue one of understanding? Is it an issue of executing? Is it an issue of evaluation? If we assume it is an issue of understanding, it is probably related to a match between the system and the real world or documentation. If it is an issue of evaluation it is probably related to visibility of system status. (P9)

> I don't think it is tough to link the issue to any heuristic; it might just require some reflection, especially if the evaluator doesn't understand the heuristics really well. Seven stages of action would be helpful here; if you understand which stage is affected, you could narrow it down to a few heuristics. (P13)

One participant suggested looking at the usability issue in terms of usability components, considering which of the usability components the usability issue at hand falls under and then finding the heuristics that deal with the same usability component. They stated that this can facilitate the process of linking a usability issue to usability heuristics: "The same for writing the heuristic, I ask them to think of the problem in terms of what we discussed about usability; it will make the process easier" (P10).

Rating the severity of usability issues

Not all usability issues found have the same effect or the same urgency in terms of needing fixing. The act of rating the severity of the usability issues is the act of prioritizing the fixing of usability issues. 13 participants stated that they always rate the severity of usability issues found during the evaluation, while two indicated that they do not rate the severity of the usability issues. The latter suggest that, by clearly explaining the issues, the receivers of the report can assess the severity of the issues for themselves:

> Rating the severity of the problem is not as important; the most important thing is that you convey how severe you think the problem is by your description of the problem. If you thoroughly explain the problem, they will be able to take the decision. I think our job is to explain the problem to them and how likely their customer is going to struggle, but giving numbers and rates isn't very meaningful. (P8)

> I don't really rate severity, I leave it to them, because the severity is related to their business goals, so they can choose what to fix and what not. (P11)

The other participants, who stated that they always rate the severity of the issues, use a variety of different scales. Five participants indicated that they use a three-level scale of low/medium/high:

> My scale is always, how much is this going to hinder the user from accomplishing their goal? Then, I rate it from 1 to 3, with 1 being the least severe and 3 being the most severe, when the user can't accomplish their goal. (P4)

> I use a three-level scale . . . mild, medium, and severe. Mild means I can overcome it easily, medium means I can overcome it but it will take some effort, and severe means I can't overcome it. (P6)

Five participants claimed that scale choice is not important, suggesting that multiple scales are available and that any one of them can work satisfactorily as long as it is clearly explained:

> We don’t actually use a scale but instead might think, is it a deal breaker? We use idiomatic language. The issue might be a showstopper. Can this prevent a launch? Is it a critical issue? (P2)

> There are several scales, I don’t stick to a specific one. The important part of the severity rating is not the numbers but explaining what those numbers represent. (P7)

Three participants identified the most popular scale, Nielsen’s scale, as the scale that they use in teaching their students. Nielsen’s scale has five levels: not a problem, cosmetic problem, minor problem, major problem, catastrophe:

> I like Nielsen’s scale for its severity ratings, from 0 to 4. . . . 0 no problems, 1 cosmetic problem, 2 minor problem, 3 major problem, 4 catastrophe. (P1)

> There are different scales out there, but Nielsen’s scale is a good one. I personally use it and teach it to my students, but I don’t force them to use it. (P10)

Rating the severity of usability issues is to some extent a subjective process in which an evaluator examines the issue at hand and attempts to rate it on their chosen scale based on their perception of its severity. For novices with little experience, underestimation, or overestimation of a usability issue can occur. All participants agree that this process is subjective and is likely to be subject to the evaluator's judgment. One participant stated: "Rating the severity level is subjective. While I might feel that this issue would stop the user from accomplishing their goal, you might think something different" (P13).

However, there are some steps that can be taken to enhance the accuracy of the rating judgment. First, all participants who habitually rated the severity of the usability issues carefully and critically examined the consequences of the usability issue, considering how business goals will be affected, how different users will be affected, or how frequently this issue is going to occur. By carefully thinking about these factors, the evaluator can give a more accurate estimation of the severity of the problem:

> In reality, severity, down to its core, is an assumption, but it is thought through. . . . So, severity is not "I think", it is "an assumption because it is critically thought of". It is an assumption because we thought of how could it impact your conversion, how could it impact your business, how could it impact checking out, how could it impact added products to the cart. (P15)

One participant discussed including personas in the process of rating the severity of the usability issues because this can help the evaluator to visualize the target audience: "My process again is to look at the personas because the issue might be minor for one user but catastrophe for another. Keeping all personas in mind clears up any ambiguity" (P1).

Three participants suggested that to enhance the accuracy of the rating, the evaluator should think about the issue while considering the human model. By understanding which step towards accomplishing the goal was affected by the issue, an evaluator can better understand the consequences and severity of the issue:

> So if you detected an issue, think of it this way: does it affect understanding? Or execution? Or evaluation? When you do this, you are going to come up with a better judgment. (P9)

> Thinking of it in terms of seven stages of action might help visualize the process. . . . If the user is not able to form the goal, what would happen? If the user is not able to perceive the output, what will happen? And so on. (P13)

One participant suggested that to improve the accuracy of the rating, the evaluator should think of the issue while considering usability and its components. By looking at which component the usability issue affects, the evaluator could arrive at a better judgment:

> Rating severity accurately requires experience; you have to be able to envision how users are going to deal with this issue, this is a difficult task for novices. My solution for this is also to think of it in terms of usability. I myself do that, and I think it helps. (P10)

Two participants suggested showing the issue to other evaluators. This assumes that the evaluator is working with other evaluators, following Nielsen's suggestion that evaluation should be conducted separately by three to five evaluators, after which their results should be compared. However, this arrangement is not always possible, as in many cases the evaluator might be asked to work alone:

> If they are doing the evaluation with others, they can compare results. When I do it with others, we just look at the mode. If two raters rated the problem as low and one rater rated as medium, then it is low. (P6)

> Each of us rate the issues then we discuss them as a group. If we disagree on something we just vote. (P11)

The final report

Given that it summarizes the whole evaluation and is the method by which the results are delivered, the final report should be of high quality. While there appears to be no overall model on the best way to write the report, there are certain common elements that participants agree should be in it. All participants stated that the report should include screenshots, as pictures can clearly illustrate the problem and show exactly where it occurs:

> For documentation, I give each problem a number, a screenshot or screenshots of the problem, the heuristics violated, a description of the problem, the severity of the problem, the possible consequences of the problem, suggestions on how to fix it, and studies and articles that explain the research done in the same area. (P5)

Participants indicated that they always include a detailed description of the problem along with the screenshots in the report:

> My report is very rich usually, I describe the usability problem in detail, I cite studies, I add pictures, I talk about what is likely to happen if the usability problem isn't fixed; basically, I make sure they understand why this is a problem and why it needs to be fixed. (P1)

While all participants agreed on screenshots and a problem description, many also had their own individual elements they add to the report; severity ratings and heuristics violated by a usability issue were the two most common. As discussed above, 13 participants stated they always include the rating, and eight indicated they include the heuristics violated by usability issues. Five participants pointed out that they also include the consequences and the possible costs of ignoring the usability issues. One participant stated: "For external reports, we add screenshots, description, the cost of ignoring these issues (in terms of money and reputation), severity to prioritize the issues, and suggestions on how to solve the issues" (P9).

Seven participants stated that adding articles or citing studies and principles to support evaluator claims is another element that can be added to the report, although two of them suggested that this may be challenging for novices, since it requires additional effort and knowledge about the literature of the field:

> I try to find articles and studies that talk about the problem. If the problem is that they have too lengthy documentation, I will cite a study from

> psychology that shows the number of people who have dyslexia or an article that talks about how people don't usually read much. . . . For sure, it is not easy for novices to go and do good research, and this is also not expected, but it is helpful for them in the first place; they will be more convinced of their results this way. (P4)

Thirteen participants also added recommendations on how to solve the issue in their report, but they again acknowledged that, even though this is an important element, it might be difficult for novices to provide useful recommendations, as they probably have little experience with respect to what should be done:

> I think for novices it's very, very hard to give good recommendations. I still make them give recommendations. (P2)

> I take screenshots for all the problems I find and write the description under each. I rate severity as either mild, medium, or severe. I give my recommendation for how the problem could be solved; most of the time I have a specific idea of how it could be solved. Other times I just give generic solution. (P6)

> I would say that adding recommendations to the report would be tough for novices, but if they can, that would be a plus. . . . In my class, I ask my student to add recommendations, then ask them to think on how it is going to affect usability, then add it. (P10)

Three participants suggested that an evaluator can become creative in delivering the results by adding additional material that they think may be appropriate, such as adding video clips, writing a story-like report, or providing a description of the struggles a user may face with the issue:

> Mainly, we describe each problem; sometimes we do this by including videos, sometimes only in writing. In the report, it is important to focus on the emotional aspect, to show the client how users are going to suffer if this problem is not fixed. (P8)

> That doesn't mean this is the only way to do the report. I encourage my students to write a scenario for each problem, to give it a context, something like "If the user didn't get the task done, X is going to happen and because X happens Y is also going to happen." . . . They can do a small video for themselves, acting like a user, to show how the user is going to feel. (P14)

However, one participant suggested that elements chosen should be consistent in the way they present all the usability issues:

> I hate it when the report is not consistent; using one severity scale for one problem and another scale for another problem or adding pictures for one problem and not for others. If you decide to add something, add it throughout the whole report; it just makes it easier to digest. (P10)

One participant emphasized that usability issues on the report should be listed in an ordered way, ideally starting with the most urgent issues, and moving down to less urgent issues:

> The report should start with the most urgent problems followed by less urgent ones, or vice versa. Listing the problems in a logical order is the most important thing, and I think it would be more logical to start with the most urgent ones. (P3)

One participant suggested that, after finishing the report, the evaluator should send it to an experienced practitioner to check it for validity and plausibility: “If they can send the report to someone with more experience, they should. Ask about the clarity of the report and see if they have suggestions to improve it” (P9).

3.6. Second part results

As outlined previously, during the second part of the interviews, we focused specifically on Nielsen’s heuristics. When analyzing the responses, we were informed by the results of the first part’s analysis, focusing on five particular items: first, for each heuristic, asking how many underlying concepts are present and what they are; second, examining which labels are difficult and how could they be improved by relabeling; third, considering what makes each heuristic work; fourth, asking what is the significance of each heuristic and when are they not applicable; and, finally, mapping usability heuristics to seven stages of action and usability components.

3.6.1. Dividing heuristics

Nielsen's heuristics are sometimes either too abstract, contain more than one idea that refers to the same concept, or have different but interrelated concepts. Using responses gathered from the participants, we sought to further explain the heuristics and reveal their underlying concepts.

Visibility of system status

From the responses, we concluded that there are four different ideas underlying this heuristic. First, the user should know the state of the system, meaning that the user should know what they are capable of doing with the system at any given time:

> The second concept is what Norman calls the affordance. It lets the user know what a certain element or elements do. The example he uses is the door handle; its shape gives the user an idea about how to use it. The same thing could be applied to interface elements. When you see a checkbox you know you can choose more than one option, when you see text that has a different color and is underscored, you know it is a link, and so on. (P10)

Second, users must have a sense of location, of knowing where they are in the system and understanding their location in relation to other parts of the system:

> How I understand visibility of system status is that the user should always be informed about everything: about their location, about what they can

> do, about how long something is going to take. The user shouldn't question anything about the system. (P12)

> The last thing is the relation of the user to the system. At all times, the user should know where he is on the system; if I am on the homepage, it should clearly indicate that I am on the homepage. (P14)

Third, users should be both actively and passively informed about progress they are making. They should be actively informed when completing a task and passively informed when they are waiting for downloading or when the system is busy processing:

> Visibility of system status is showing the user the progress of their action, how close they are from getting to their goal. This is very significant because information about progress lets the user know what to do with their time. (P4)

Finally, users should be informed when a task at hand is completed and should be provided with a sense of closure:

> A popular example is when you make a transaction and don't know if it went through or not; that is extremely confusing and frustrating. It is even worse if they made the transaction again and it turns out that the first one had already gone through. (P8)

> It is important to let me know that I have completed my action. Some people think it is only important to let the user know when something wrong happened, but if everything is OK, they think it is fine to just

> redirect them to the homepage, and that’s so wrong. Users need closure so they can know they are good to go. (P13)

Match between system and the real world

When we discussed matching the system to the real world with participants, three major concepts emerged. The first is that the content presented on the system should be understandable by the target audience, so the user should not need to question the meaning of content presented on the system:

> Match between system and the real world is like windows. They did a great job using their metaphors. Folder, recycling bin, eraser, etc., all these are real-world objects that the user is familiar with and knows their function. (P9)

> Everything you show to the user on the interface should make sense to them, not only to you. I recently visited a medical site supposedly created to increase the awareness of normal people, but the system is full of medical jargon impossible to understand by normal people. (P13)

The second is that the content presented on the system and the actions performed in the system should follow a logical and natural order:

> Make the flow as natural as possible; for any action in your system see how people do it in real life and imitate that. Don’t do otherwise unless you can come up with something easier and intuitive. (P8)

> Actions also make them natural, like when I buy something from the market, so when I buy from an online market, it should be a similar experience in terms of the order. (P15)

The third is that the content presented on the system should be appropriate to the target audience and match the purpose of the system. Even if the content is understandable, it still must be acceptable:

> It's not only about using clear content on which many people solely focus. It is to use appropriate content. The content should be polite and always match the identity of the application or the website. (P6)

> If the website is for children, you shouldn't use explicit language that is not appropriate for their age. This is going to be offensive. (P12)

User control and freedom

Based on the interviews, we identified three concepts related to user control and freedom. The first is the idea of undo and redo, which provides the user with the ability to undo anything previously done and redo anything deleted or lost:

> When I use the site, I should have control over it. I can delete whatever I want then get it back, I can go to a certain page and get back to where I was. Things like that. (P6)

> This is like enabling a back button because if a customer presses the wrong button and has to go through an entire process just to get back to

> the page that they were first at, no one's going to use the product because it's way too much work. (P7)

The second is the idea of an emergency exit to give the user the ability to escape any undesirable situation. Some participants suggested that providing undo and redo can also serve as an emergency exit, while others thought that undo and redo capability is slightly different from an emergency exit. They pointed out that, while undo and redo enable the user to get something back, an emergency exit allows escape from an undesirable situation:

> User control and freedom means making redo and undo available at all times. Nielsen mentioned an emergency exit, and I feel that undo and redo could be an emergency exit as well. (P1)

> User control and freedom means to make all the actions reversible to give the user a way to exit from any situation they don't like. (P3)

> Allow the user to get out of any unpleasant situation. Don't force me to see all the popups; give me the option to eliminate them completely. (P14)

The third is giving the user sufficient information about the task at hand to support making an informed decision. The user should be able to know why the system is asking them to do something so they can decide whether or not to do it:

> Also in regard to user control and freedom, I should know why you are asking me to enter certain information. Like when a bank asks me "Do you have another passport?", I should know why they want to know that. Only when I know why I can decide whether to answer or not. (P6)

Consistency and standards

Most participants highlighted that consistency and standards, while closely related, are not interchangeable. Standards basically ask that common practice, which are practices followed by most similar systems, be followed, whereas consistency relates to being consistent within the system, meaning that, once an action is applied in one place in the system, it should be similarly applied throughout the system:

> Consistency and standards relate to following what is known to be typical practice in a certain situation, and to be predictable in the different parts of the system. If you looked at a certain action and said "Why are they doing it this way? This is so different from most apps", standards have been violated. Or if you said "Why is this different from the previous page, or even from the last time I used it?", consistency is missing. (P3)

Nevertheless, two participants suggested that following standards usually enforces consistency:

> It is better to stick to the standards. This will also help with consistency. If the standard is to place the search bar in the top of page, and you follow standards and place it in the top of each page, you have both followed the standards and ensured consistency. (P1)

> Consistency and standards: standards usually enforce consistency. . . . The design system that most people are using now is a good example of

> standards; if you follow standards you increase the chance of being consistent. (P8)

Error prevention

Error prevention is the heuristic with the greatest number of underlying concepts. We concluded that there are seven distinct ways to prevent errors or reduce their likelihood. First, constraints can be used to prevent errors: “An error prevention example is when entering letters on a phone number field, the system shouldn’t allow me to enter letters but should allow only numbers” (P11).

Second, clear instructions can help users avoid mistakes:

> You can use placeholder text to guide the user as to what to enter. In a form, when you are asked to enter your email, the placeholder text can give an example of the right way to enter it. (P8)

Third, asking users to confirm each action before completing it can also help prevent errors: “Asking users to confirm before submitting their input is one of the ways in which you can ensure that the inputs are correct” (P14).

Fourth, notifying users when important or serious errors occur would help them to take actions to prevent future errors: “In any phone, when the battery reaches 20%, it lets you know so you don’t forget to charge it before it shuts down; the notification helps you prevent the error” (P1).

Fifth, automatically saving the input can reduce the effort of users by avoiding the need to re-enter information when something goes wrong:

> The first one would be auto-saving. I don't see people talk about this a lot, even though it is really important. I am sure you have experienced this in some form or application; when you are halfway through and by mistake close the tap or the computer shuts down, you lose everything. Auto-saving would help avoid these types of errors, sometimes basically saving your life. (P13)

Sixth, while using defaults is important in reducing errors, bad defaults can themselves lead to errors: "The second one is default; imagine if the system default is not to notify you if there is an important update or something like that; how many errors could happen as a result of that?" (P13).

Finally, providing users with the capability to enter flexible inputs can reduce errors:

> You can make the input flexible so that can take many forms. When you enter your credit card number it should allow you to either enter it with space between every four digits or you can just enter it without spaces. The same thing for date; you can either write the name of the month or just enter the number of the month. (P8)

Recognition rather than recall

Two ideas derived from the responses were related to using recognition rather than recall. The first, related to information availability, is the idea that the information a user needs should be presented to them so that they do not need to remember it:

> Voice-conversational interfaces. They are terrible for music because you need to say the band that you like and then you're like, "Alexa, play," so you're having to recall what it is. You don't get to talk about album art or get to see anything. I don't know my band's favorite song. (P2)

> Recognition rather than recall, simply don't make me remember anything as much as possible, make everything available on the interface. (P5)

The second idea, related to giving suggestions, is that, while it is not always possible to present everything a user needs and sometimes it is not clear what a user has in mind, the system should try to provide appropriate suggestions:

> The best example for recognition rather than recall is Google. When you start to type anything, it gives you a list of suggestions. This is literally life-saving. In so many cases, I roughly remember the name and Google helped me to reach what I want. I think this should be applied in any app. (P12)

> The example I like is when I create an account on Twitter, they give me suggestions on who to follow. It is great because I don't need to recall who the popular users are, or they suggest friends based on the email I entered. (P13)

Flexibility and efficiency of use

While most participants thought that flexibility and efficiency of use is related to the ability of the system to meet the needs of all the different users who may use the system, some participants differentiated flexibility from efficiency.

Flexibility means that the system should provide different alternatives to the user, while efficiency means making sure that any given system task is presented in its most simplified form and does not require unnecessary effort:

> Flexibility is to provide more than one option, like accelerators, shortcuts, and the like. . . . Efficiency is related to measuring the effort needed by an action. There is a rule in UX called the three-clicks rule; any goal in the system should or would be preferred to be reachable in three clicks or less. Although I don't think this always means efficiency, it is a good way to think of it. (P1)

> My interpretation of flexibility is to make the same thing accessible from different places and done in different ways. Now, this in many cases could really lead the product to be efficient, but not always. Now, if all options for doing the same thing are terrible, the product is not going to be efficient, so we need also to examine each option for efficiency to see if it is in the simplest form. (P10)

> Flexibility and efficiency of use. If I am a novice, can I use the system? If I am disabled, can I use the system? To me this heuristic means you should accommodate everyone you expect to use your system. It is like having different views of the system; if I like the dark mode or the bright mode, both should be offered. (P11)

Aesthetic and minimalist design

Participants expressed different opinions on aesthetic and minimalist design. Some participants did not like this heuristic, with some suggesting that it assumes that aesthetic and minimalism are necessarily related and others pointing out that minimalism is very subjective and varies from culture to culture, so some indicated that they usually simply ignore it:

> We delete it sometimes; we don't want to use it because you can't tell me that it has to be minimal. I get it. I understand why. I think it should be minimal. I enjoy it personally, but in certain environments you need a maximalist design or you need something that's showing way more. . . . Something that would be minimalist in one culture might not be minimalist in another. Some cultures might want one thing to be present while others might think five's okay, and others might want a hundred. (P2)

> There is a problem with the heuristic of minimalist design; they assume that minimalist designs are always good and beautiful and that's not always the case. . . . What is minimalism anyway? It is hard to define that. . . . In some cultures one thing could be considered minimalist, while the same thing could be considered too much in another culture. (P5)

From this, we can conclude that there are three ideas related to this heuristic. First, the content of each page should be organized and clear:

> Organization means they follow gestalt principles, so everything is grouped correctly and there is a space to separate them. (P11)

Second, only necessary content should be presented: “When you are deciding and defining content for a page, you need to be focusing on what’s relevant and important to the user. Provide only what’s necessary so you don’t distract the user” (P7).

Third, although some participants viewed aesthetics as subjective, they highlighted the importance of making the interface attractive, feeling that this might attract some users even though there might be some usability issues:

> Aesthetic is simply making the system appear in a beautiful and appealing way. This one is a bit hard to measure. It might differ from person to person, age group to age group, culture to culture, and so on, but it is known that beauty has an effect on how users perceive the system. If it is beautiful, they will perceive it as more usable. It is the aesthetic-usability effect. (P1)

> Focusing on the beauty of the interface is a must, I personally use some apps even though I know I have better alternatives just because I like their design. (P11)

Help users recognize, diagnose, and recover from errors

This heuristic encompasses three different underlying ideas: first, the user should recognize any error, meaning that they should know that an error has occurred; second, the user should know what the error is, which means that they should understand the error they made; and, third, the user should be able to recover from the error. Some participants focused on one of these ideas, some referenced two, and some discussed all three:

> Help users recognize, diagnose, and recover from errors. It is quite simple, let the user know how to correctly and efficiently solve the mistakes they made. (P1)

> Helping users recognize, diagnose, and recover from errors. Making sure that people understand what problems they've caused and then how to possibly prevent them in the future. (P2)

> Helping users recognize, diagnose, and recover from errors is to make the inevitable errors easy to recover from. When I make an error I want a solution, a clear one, I don't want to only know what I did wrong. If I don't know how to solve the issue, then I am not going to get my goal done. (P4)

> Help users recognize, diagnose, and recover from errors. Nielsen here gave enough details in the label. One, recognize, which is noticing the error. Two, diagnose, which is knowing the error. Three, recover, which is knowing how to overcome the error. (P10)

Help and documentation

Some participants gave generic answers regarding this heuristic on offering help and documentation for the users, suggesting that it is clear and does not require further explanation:

> Help and documentation I think is pretty straightforward. It's just the help system or the documentation. (P2)

> Help and documentation, not sure if this needs any explanation, but briefly it is providing written or visual material that emphasizes the important things in the system. (P4)

However, other participants differentiated between help and documentation. They stated that help is usually available externally from someone with whom the user makes contact:

> Help is different. I mostly associate it with solving immediate problems. I am stuck on something and don't know how to use it and I want the answer now, so I'll call customer service to ask them. That's help, basically their phone number or a chat with their customer service representative. (P11)

Documentation, by contrast, is internal; it is something that the user can read, or observe on the system, to get help without interaction. It might be described as self-help:

> I will start with documentation; you can think of it as the catalog of the system. Everything that is related to the system should be written down, including things like how to use every function, what the available options are, what everything means. The main reason for this is to show me how to learn to use the system and prevent errors. (P11)

3.6.2. Labeling heuristics

While most participants agreed that Nielsen's heuristics have unclear or ambiguous labels, with only two disagreeing with this assessment, not all believed that these labels should be renamed. Two participants stated that giving more examples and better descriptions reduces the need for better labels, as the description and examples can sufficiently convey the meaning of the heuristics. Others also suggested that renaming a label should not be done individually, meaning that they should not be asked to rename the labels, and that naming labels should be done through card sorting or survey, where a set of proposed labels for each heuristic is presented to a large number of usability practitioners and they are asked to choose the easiest to understand:

> I don't think l can label it on the spot. I should give it a lot of thought. (P3)

> I would say yes the labels are clear, but what is the importance of the labels anyway? As long as the concept is clear by the examples, I think any label would be OK. To me, after I read the labels for the first time, I was able to tell what the general idea is. (P4)

> Doing the naming myself would be tough. I think we should do it as a card sorting or something like that, where we can test different labels and see which ones would make more sense to most usability practitioners. That's the best we can do, but I will try to give you my opinion. (P12)

Others agreed to contribute by offering suggestions for new labels, which are discussed in the following sections.

Visibility of system status

Eight participants stated that the “visibility of system status” label is not the best label for describing this heuristic. Seven of them proposed other labels, while one suggested that they did not like the current label but could not think of a better one. The suggested labels were: keep the user informed, visibility of location, make the status of the system visible, visual feedback, feedback mechanism, feedback, and visibility of feedback.

> I would change visibility of system status to keep the user informed. The whole idea of this heuristic is to keep the user informed at all times so that would be a more appropriate label. (P1)

> Visibility of system status could be a feedback mechanism because it won’t just focus on the status but make it general to include the user location. (P11)

Match between system and the real world

Seven participants stated that the “match between system and the real world” label is unclear and does not effectively convey the meaning, and six participants made the following alternative suggestions: use familiar and relatable content and structure, meet users expectations, matching users expectations, the flow should feel natural, use appropriate content, and match users mental model.

> Match between system and the real world could be changed to appropriate content. (P12)

> If I would rename it, I will make it match user's mental model, because it is all about relating to what people know and experience in the real world. (P15)

User control and freedom

Nine participants stated that "user control and freedom" is not a clear label, and six of them suggested alternatives as follows: increase user control and freedom, self-service, let the user be in control, user control, undo, and redo, and give the user the freedom to make mistakes.

> User control and freedom gives the impression that the user should have absolute control and that is not realistic. What this heuristic wants to say is increase user control and freedom, and that would be more accurate. (P1)

> I am not sure about the freedom part, so let the user be in control, is a better way of writing it. (P6)

Consistency and standards

Only one participant had a problem with the "consistency and standards" label for this heuristic. They suggested either separating the two concepts or dividing the heuristic into two heuristics, with one to be consistent throughout the system and the other to follow standards:

> Consistency and standards should be separated, so one heuristic should be consistency, or we can call it be consistent throughout the system.

> Standards should also be separate; it could be just standards or we can call it follow the standards. (P1)

Error prevention

While none of the participants found the “error prevention” label unclear, two participants thought it could still be improved. Their alternative suggestions were: prevent errors and prevent them as much as possible.

> Error prevention is a good label, but I also think it is not totally accurate. It gives the impression the errors could be completely prevented, and we all know that is kind of impossible. . . . So we can change error prevention to prevent errors as much as possible. (P1)

> Error prevention conveys the meaning, but prevent errors is a better way of putting it, because that’s the purpose, to try to prevent errors. (P6)

Recognition rather than recall

Eight participants stated that “recognition rather than recall” does not clearly explain the meaning of the heuristic, and six of them provided suggestions for improvement: do not make users rely on their memory, make options visible, users should not have to remember, wayfinding, reduce memory load, cognitive load, minimization of cognitive load, and offer suggestions.

> Recognition rather than recall, there’s a lot baked into that. Not necessarily sure how to change it. Make options visible or users shouldn’t have to remember. Also wayfinding popped into my head. (P2)

> Recognition rather than recall. When I first learned this, this term did not make sense until you really dug deep into it. Minimization of cognitive load is another way to name it. (P7)

Flexibility and efficiency of use

Seven participants thought that "flexibility and efficiency of use" is an ambiguous label, and one stated that, since flexibility and efficiency are not the same, they should also be separated. Four participants suggested alternative labels: accommodate different types of users, reduce the effort as much as possible, give the user multiple ways to do the same task, and cater to multiple user experiences.

> Flexibility and efficiency of use are two different things. One heuristic could be flexibility, or we can call it accommodate different types of users. The other heuristic could be efficiency or we can call it reduce the effort as much as possible. (P1)

> Flexibility and efficiency of use is to give me multiple ways to do the same task. . . . I will call it give the user multiple ways to do the same task. (P5)

Aesthetic and minimalist design

Four participants stated that "aesthetic and minimalist design" should be relabeled. Two of them suggested new labels, aesthetic and organization and focus on efficient content design, while one proposed that aesthetic and minimalist should be separated into strive for beauty and make the design as simple as possible:

> Aesthetic and minimalist are not good terms at all for the label and not in terms of simplification. It could be simplified to two heuristics. Aesthetics could be called strive for beauty and minimalist could be make it as simple as possible. (P1)

> No need to change aesthetic but I would change minimalist to organization, so it would be aesthetic and organization. (P5)

> The best way to name it is perhaps to say the product is not excessive or focusing on efficient content design. (P7)

Help users recognize, diagnose, and recover from errors

While all participants agreed that this heuristic label is clear and conveys the meaning effectively, two suggested that it is very long and could be simplified. Their alternative suggestions were: help users recover from errors easily, and error recovery. One participant explained: “I love to help users recognize, diagnose, and recover from errors. The only thing I don’t like about it is that it is a bit lengthy. We can change it to ‘help users recover from errors easily’” (P1).

Help and documentation

While all participants found the “help and documentation” label clear, three provided suggestions for improving it further. One suggested that help and documentation should be separated and labeled as provide help when needed and provide sufficient documentation, while the other two proposed the labels provide accessible help and documentation and providing guidance:

> Help would be a heuristic and could simply be called “help” or we can call it “provide help for users when needed.” Documentation is another heuristic and could be just named that or it could be named “provide sufficient documentation.” (P1)

> “Provide accessible help and documentation” is better than just “help and documenting” because you don’t want to just provide help in documentation, instead you want them to be accessible. (P6)

3.6.3. Heuristics components

One part of explaining heuristics is to identify what makes a heuristic work well, to show what are the things that need to be examined most carefully for each heuristic. To extract this information, we looked at the descriptions of the heuristics and the examples given by the participants.

Visibility of system status

Visibility of system status has four underlying ideas: state, location, progress, and closure. State allows the user to know the condition of the system through its three components: first, there should be some indicator to inform the user of the current system state; second, the state of the system should be visible and easy to interpret; and, third, if any changes have happened to the system, they should be reflected immediately on the interface.

> Some UXers would say, “Well there is an indication that the item is sold,” but they forget that the user also has to see it and understand it. (P3)

> In a mobile app we designed, we had an issue. When the user pays the bill, it takes hours to change the status to paid. . . . This confuses users, and they usually call the support team to complain. (P6)

Location refers to the current location of the user inside the system. There are four components related to this idea: first, there should be an indicator to tell users their location in the system; second, the indicator should inform them of the relationship between their current location and the rest of the system; third, the indicator should be clear and visible; and, finally, it should be presented throughout the whole system, not just in some places while ignored in others.

> The user wants to know where they are exactly. . . . When you go to any shopping mall, you look to find where you are and how to get to where you want to go; they have the panels that help you with that. . . . When we do eye tracking, we notice that one of the first things that users look at is where they are located, so the navigation bar should be really clear. (P5)

> In certain sites, some pages just don’t have a navigation bar, and your only option is to press the back button or press the home button. . . . You suddenly feel you are in a different site. (P11)

Progress means that the user should know how far they are from accomplishing a certain goal. There are three components related to this idea: first, the appropriate type of indicator, which depending on the case can be presented in terms of time, steps, capacity and so on and, in some cases, should be presented as a combination of two or three different measures; second, the progress should be accurately described as, for example, indicating that a task will take 10 minutes which then takes 30 minutes defeats the purpose of showing progress; and, third, progress should be visible in both active and passive situations, meaning that it should be visible both when the user is working on a task such as filling a multi-page form, showing how many pages are left, and when the user has taken an action such as downloading or uploading, showing how long they have to wait for the outcome.

> If I'm clicking on something and there's going to be a loading, it needs to tell me it's loading. (P2)

> If I am filling out an application and this application consists of multiple pages, I should know which page in the application I am in. I should know how many pages are left to finish the application. (P12)

> But be careful with the progress. You know that when you download a file and it says it is going to take an hour, it takes a whole day. Obviously in this case "progress" is meaningless, and not having it is better. (P13)

The idea of closure is to clearly mark to the user that their action has been completed. This has four underlying components: first, closure should be offered immediately after the completion of the action; second, it should be offered no matter whether the task was successful or not; third, it should be clear and understandable; and, fourth, if the task was not successful, some explanation for this should be provided.

> When you make a transaction, and you don't know if it went through or not, that is extremely confusing and frustrating . . . so I should know the result immediately after the transaction to avoid any confusion. (P8)

> Some people think it is only important to let the user know when something wrong happened, but when everything is OK they think it is fine to just redirect them to the homepage. . . . When something wrong happens, it should explain why that wrong happened. (P13)

Match between system and the real world

Match between system and the real world has three underlying ideas: the understandability of the content, the logical and natural order of the content and actions, and the appropriateness of the content. Understandability of the content refers to using content that is understandable for users, meaning that anything used in the system, including a word, phrase or icon, should be understandable for the target audience. It is therefore judged on its understandability by potential users of the system: "Don't use technical words or jargon. Unless the target audience knows them" (P8).

The logical and natural order of the content and actions refers to the way that content is presented and the sequence taken for each action. There are two components to this idea: first, if the content or the action offered on the system is also followed in the real world, it should follow the same sequence as that of the real world; and, second, if the content or the action does not have a counterpart in the real world, the order should be logical, meaning that it needs to be intuitive and should not requiring too much thinking.

> Real world is not always easy to understand. What they mean by the real world is logical or intuitive, something that doesn't require a lot of thinking, something that comes naturally. (P1)

> Make the flow as natural as possible; for any action in your system, see how people do it in real life and imitate that. Don't do otherwise, unless you can come up with something easier and intuitive. (P8)

> Actions also, make them natural, like when I buy something from the market, so when I buy from an online market, it should be a similar experience in terms of the order. (P15)

The appropriateness of the content refers to the acceptability of the content for the target audience. There are two aspects which should be taken into account regarding this idea. First, the content should match the purpose of the system. For example, if the system is official, then the content should be formal, so that the content is consistent with the identity of the system. Second, the content should be appropriate for the target audience. For example, if the system is for children, then explicit language or imagery should not be used, or if the system is for a certain culture, then content considered inappropriate within that culture should not be used.

> Is not only about using clear content on which many people solely focus. It is to use appropriate content. The content should be polite and always match the identity of the application or the website. . . . Imagine visiting a government website or university website and they are using internet memes. That's not appropriate. (P6)

> If the website is for children you should not use explicit language that is not appropriate for their age. This is going to be offensive. (P12)

User control and freedom

User control and freedom has three underlying ideas. The first is that the user should be able to reverse any action they took in the system, which itself has three components: first, the user should be able to reverse any action, at any time and at any place in the system; second, the way to reverse the actions should be clear and easy to accomplish; and, third, the user should preferably be able to go back as many steps as desired while undoing and redoing actions.

> To reduce the effort, I should always be able to get back to where I was. (P7)

> The worst thing is when you change something multiple times and it allows you to only reverse the last change. . . . I should be able to reverse changes as much as I need. (P9)

The second idea is that an emergency exit should provide the user with the ability to escape from any undesirable situation, which has two components: first, the user should be exit any undesirable situation from any place at any time; and, second, the user should know easily how they can make this exit. One participant stated: “Sometimes they intentionally make hiding ads or popups hard; this violates the emergency exit concept. It should be easy to block popups or skip the ads” (P14).

The third idea is to give the user enough information about any certain task to enable them to make an informed decision, which has two components: first, explain to the user why you are asking them to do certain things; and second, explain to the user how their input is going to be handled.

> When a bank asks me do you have another passport? I should know why they want to know that. Only when I know why I can decide whether to answer or not. . . . But it’s also important to know with whom the information is going to be shared. (P6)

Consistency and standards

Consistency and standards have two underlying ideas. The first is consistency, which refers to using system elements consistently throughout the whole system. We concluded that there are five types of consistency. First, consistency in meaning; if an element has specific meaning in one place, it should have the same meaning throughout the system. Second, consistency in functionality; if an element does something in one place, it should do the same thing in each part of the system. Third, consistency in organization; if one part is organized in a certain way, then the remainder of the system should follow the same general organization. Fourth, consistency in feeling; there should be a consistent feeling throughout the system, meaning that the system should function as one unit. Finally, consistency in effort; in a multi-step operation, every step should require a similar effort.

> We have a design system to make sure that, if there's a call to action button on one page, that it's the same on another page. It doesn't just have to be in the same place, but there's still a conceptual model. We use headers, we use the same header, like hierarchy there's a visual hierarchy. So any consistency that we can have across sites. (P2)

> Balancing the effort is mostly overlooked. If I am filling a multi-step form, let's say four steps, and the first step took me 2 minutes, then I would expect that every other step would also take me 2 minutes. The effort should be equally divided. (P6)

> There are three main ways for ensuring consistency. Being consistent in the meaning: if a certain phrase means something in one place, then the same phrase should mean the same thing in other places. Being consistent in function: if a certain element does a thing in one place, it should do the same thing in other places. Consistent in feeling: the feeling the user has on the system should be consistent in all the system. (P10)

The second idea is standards, which refers to following conventions and best practices. We concluded that there are three things that standards follow: first, they should follow established meanings in names, phrases, etc.; second, they should follow known ways in which certain actions are done; and, third, they should follow standard schemes with which similar systems are organized.

> If you looked at a certain action and said "Why are they doing it this way? This is so different from most apps," standards have been violated. (P3)

> If most apps and sites call the search button "search" it is pointless and confusing to change to "query" or "look up." (P10)

> The best example is where to place the logo of your system; in most systems, the logo is placed at top left. In your system, it is better to follow the same thing because this is the first place the user will look, top on the left, to know what the system is. (P11)

Error prevention

Error prevention is a rich heuristic with seven different underlying ideas. The first idea is to provide instructions for the user on how to do certain tasks, which has three components: first, instructions should be prominent and clearly visible; second, they should be understandable; and third, they should be as concise as possible.

> In many cases, the placeholder text isn't clear, either because it is very light to an extent that I can't read it, or because they're just writing things that are meaningless. . . . I am not a big fan of providing long paragraphs, as most users aren't going to read anyway. Be straight to the point and explain what they have to do in two to three sentences. (P10)

The second idea is to use constraints to prevent users from entering invalid input. A main point here is to give users hints on why they are being limited, otherwise they will think there is something wrong and may continue trying: "When they hover over a frozen element, give them a hint, like "You don't have enough credit" or something like that just to let them know they can't use it at the moment" (P10).

The third idea is to ask users to confirm their actions to ensure they are intended. The main point here is letting the user know that confirmation is requested; sometimes the user might unintentionally press an "OK" tab, so it is important to ensure that it is clear that confirmation is being requested.

> Don't use confirmations repeatedly; that's going to make users click "OK" every time it appears. Use it wisely with important actions only and make it clear that this is a confirmation message, either by making the font really big or by using a catchy color. (P14)

The fourth idea is to notify users when changes have happened to the system that might affect them, which has three components; first, it should clearly tell the user what the notification is about; second, it should explain to the user the consequences if they don't take action; and, third, it should notify users only about important actions.

> In any phone, when the battery reaches 20%, it lets you know so you don't forget to charge it before it shuts down; the notification helps you prevent the error. . . . Having too many notifications would be annoying. It should

> be used only when something is about to happen that is going to stop the user from getting their work done. Like the example I gave, if the user didn't charge their phone, it will shut down. (P1)

The fifth idea is to autosave user input, which has two aspects: first, users should know that their input is being autosaved; and second, retrieval of the saved input should be automatic, or at least easy.

> It would be nice to inform the user that their work is being saved so they don't freak out if anything happens. Just like Google Docs, it informs you that your work is saved just after you type. . . . It would defeat the purpose if the user can't easily retrieve their work after the shutdown. . . . They should just get back to where they were, or should be presented with the option of retrieving their work. (P13)

The sixth idea is to use appropriate defaults so that users don't make errors, which raises two points to consider: first, that the default should be the most expected thing; and, second, that there should be some indicator describing the default.

> In most smartphones, I wouldn't expect that the default state to not to notify me of important updates. . . . It should start with what I expect the most, then tell me for example "The default is to receive notifications, do you want to change that?" (P13)

The final idea is to make the input flexible so the user can enter the input in a desired form. Flexible input means that the conversion from user form to system form should be visible to the user, to permit assessment as to whether the conversion is accurate and as intended. One participant explained: “But it should present it back to the user before submission to make sure it is the right input” (P8).

Recognition rather than recall

There are two ideas underpinning recognition rather than recall. The first is to make available everything that a user needs to accomplish the desired goal, which has two underlying components: first, all needed information should be clearly visible to the user; and, second, if task completion requires multiple steps, at each step all information needed from previous steps should be transferred to the next step.

> Before filing a lengthy form, it is better to tell the user what they are going to need in order to fill that form correctly and easily. Don’t let them enter the form and try to remember everything by themselves. If you are filling a form that requires your ID number, your credit card, and so on, you should know before filling the form that you would need these things, so you can prepare yourself, and don’t put so much effort into remembering them. (P9)

> The thing that I always use as an example is when the system gives me an order number and later asks me to enter it somewhere, it should be written somewhere so I can remember it; it is not logical to assume that I will recall it. (P11)

> Everything needed is readily available; make all options available. It might be tough to present everything, but at least the most important things. When I look at the menu, I shouldn't spend time to recall what these things are, they should be clear. (P13)

The second idea is to provide users with suggestions, which has four underlying aspects to consider: first, when users are beginning from an empty state, provide them with the most suitable suggestions based on their personal information; second, when users are searching for something, give them relevant suggestions; third, when users are browsing for something, present them with similar things at the side; and, finally, all suggestions should be as accurate as possible.

> The only example I have at this moment is, when you shop on Amazon or Souq.com and buy a phone, it gives you some recommendations to buy covers or headsets. Many times you only want the phone, but in other cases you really want some accessories as well, so this just makes it easy for you . . . but it is extremely annoying and distracting if the suggestions have nothing to do with what you're browsing. Like if you're looking for a phone and it swamps you with all sorts of different electronics on the side. (P10)

> The example I like is when I create an account on Twitter, they give me suggestions on who to follow. It is great because I don't need to recall who the popular users are, or they suggest friends based on the email I entered. (P13)

> I love the suggestions of Google, it really saves my time, plus I don’t always remember what I am searching for exactly, so it saves me the effort of trying to recall the name. (P15)

Flexibility and efficiency of use

Flexibility and efficiency have key differences. Flexibility is providing users with different ways to accomplish the same goal, which has two main aspects to consider: first, when providing different methods, different users should be kept in mind, meaning that the system should accommodate all types of potential users; and, second, situations in which the system is going to be used should be kept in mind because users may use the system in different situations and settings.

> Flexibility and efficiency of use is to tailor the system for the needs of each individual user; this is going to make the system very relatable to the user. (P4)

> Flexibility of the product to allow the user to do it the way they want to achieve their goal. (P7)

> If I have a headache, I don’t want the bright view, I want the dark one, so that’s a good example of accommodating different situations. (P11)

Efficiency means that the task should be structured in its simplest form with no extraneous steps, which requires three checks: Are there steps that could be removed? Can we minimize the effort required? Can the task be accomplished in less time?

> There is a rule in UX called the three clicks rule. It is that any goal in the system should or is preferred to be reachable in three clicks or less. Although I don't think this always means efficiency, it is a good way to think of it. (P1)

> Efficiency is to reduce the effort of the task as much as possible. (P13)

> We should examine each task to see if we can do anything to reduce the time it takes to be completed. (P10)

Aesthetic and minimalist design

Aesthetic and minimalist design encompasses three underlying ideas. The first is to have well-organized content, which has two components: first, all related elements should be presented and organized in a way that shows their relationships; and second, different groups of information should be clearly distinct from one another.

> I should be able to clearly tell that this is a menu, a title, or a search bar. Each element should be clearly and easily recognized. (P3)

> Organization means they follow gestalt principles, so everything is grouped correctly and there is a space to separate them. (P11)

The second idea is to have simple and efficient content, which has two components: first, all information presented in the system should be required or necessary, and extraneous information or features that serve no purpose should be removed; and second, anything that might distract the user from focusing on accomplishing a goal should be removed.

> When you are deciding and defining content for a page, you need to be focusing on what's relevant and important to the user. Provide only what's necessary so you don't distract the user. (P7)

> Many sites and applications tend to include animations and fancy visuals. While they are nice, they distract the users from focusing on accomplishing their goals, so it is best to remove them. (P15)

The third idea is to have an attractive design for the system, which has three components: first, the visual elements on the system should be used carefully and appropriately chosen: second, all elements of the system should fit together and appear in harmony; and third, videos and audio clips should be attractively presented.

> The beauty standards differ from culture to culture; I would be careful on how to choose the visual elements. Things like colors, symbols, and so on . . . I should design for my users not me, so I should take their preferences into account. (P1)

> The way that multimedia is used on the system is important. I don't only mean how they are placed on the interface, but also the content itself. . . . You don't want a nice system and poor video content. (P11)

Help users recognize, diagnose, and recover from errors

As indicated by the label, this heuristic has three main ideas. The first is that the user should know if an error has occurred. The main point is that the error indication should be presented in a manner expected by the user: "To recognize an error, I should be notified in a way I would expect and be familiar with. Like most error messages appear in red. There is no need to be creative and change it" (P13).

The second idea is that the user should be able to diagnose the error, which has five components: first, error information should be presented in a way that is readable for the user; second, error information should be presented in a way that is interpretable for the user; third, error information should be presented in a manner appropriate to the purpose of the system; fourth, error information should not be presented in a way that assigns blame to the user; and, fifth, error information should not be presented in a way that is intimidating to the user.

> Error codes aren't going to help the user know what the error is unless they google it. So, instead of error codes, just use plain language, something that the user would understand immediately. (P1)

> In finance, I think it'd be like, even funerals, like different situations where you don't need flowery language. (P2)

> Don't use words like "illegal action". Words like this scare the user; use softer words. . . . Don't make it personal, don't make the user feel it is

their fault. Don't use "You made a mistake," just generic sentences like "An error occurred." (P10)

The third idea is that the user should be able to recover from errors. The main point here is that the solution should be actionable, meaning that it is step by step and therefore easy to follow: "I don't want a message like 'Your mic isn't enabled,' I want to know how to enable my mic, because in many cases this isn't obvious" (P4).

Help and documentation

There are two underlying ideas for help and documentation. The first, documentation, has the basic mission to provide enough material for the user to understand the system, which has many components: first, documentation content should be complete and explain all the different elements of the system; second, documentation content should be clear and easy to understand; third, documentation content should be accessible, meaning that the user should know how to access it; fourth, documentation content should be searchable, which means that the user should be able to search through the content; fifth, documentation content should be relevant, so no extraneous information should be presented; and sixth, documentation content should be prioritized, so it should be prioritized by frequency and importance of actions; seventh, documentation content should be clearly categorized; eighth, if the documentation is presented in an audiovisual form, there should either be multiple audiovisual materials to explain the different elements or, if they are contained within a single audiovisual material, each element should be clearly tagged so the user can easily get to the desired part; and, ninth, the contextual documentation should be provided for important and frequent tasks.

> The most important qualities in documentation for me is that it has to be easily accessible, it should be categorized, and I can search on it. . . . The worst example is if you can't find any material on the site when you are faced with an issue or when you get the material but it is not understandable and very long and hard to read. (P6)

> Providing the question mark button in the major tasks to help users know how to overcome any issues they might face. (P7)

> You can think of it as the catalog of the system. Everything that is related to the system should be written down, including things like how to use every function, what the available options are, what everything means. The main reason for this is to show me how to learn to use the system and prevent errors. (P11)

The second idea is providing external help to users, which has two components: first, there should be multiple methods of contact from which users can choose; and second, the expectations of each method should be clearly stated.

> Some users prefer to send an email; others prefer to call by phone. You should offer all these options for your users. . . . If you provide an email, tell users when you are going to respond. In 2 days? 3 days? Or when they call, you should tell them how long they are going to wait. (P11)

3.6.4. Heuristics significance and applicability

Nielsen's heuristics explain basic concepts without giving details of their significance and relevance and it does not explain when they should not be used. Using the participants' responses, we tried to gather further explanations of the significance of these heuristics and why they should be applied, while also attempting to explain when they should not be used.

Visibility of system status

There are four reasons for knowing the system state: first, such knowledge is essential for learning, and clearly showing the system state will let users know what to expect and what to do; second, if a user does not know the system state, they will need to spend a lot of time to figure this out, meaning that the user may not use the system and might search for alternatives; third, users may be given false hope, because, if they think they are capable of doing certain things and this turns out to be untrue, they may feel frustrated; and, fourth, if the system state is not clearly stated, errors may occur, because a user who assumes the system to be in a certain state and acts accordingly might produce erroneous results.

> If you don't know the state of the system, chances are that you are going to make mistakes. (P3)

> If the user doesn't know the status of the system, one of two things would happen: they would either wait till the results, appear but they will be feeling lost meanwhile, or they would quit, because they will be thinking that the system isn't working and will go somewhere else. (P9)

> Affordance is also important for learnability, I am going to know quickly what to do, not just for learnability but also to prevent errors. (P10)

> Once I chose an item, and in the checkout they said they don’t have it, they basically wasted my time. I was disappointed because I was excited to purchase it. (P15)

There are also four reasons for informing a user of their location in the system: first, not knowing this location may confuse users; second, if users don’t know where they are located, they might not know how to accomplish their goal, because the steps required to accomplish any goal are often dependent on where they are in the system; third, when a user leaves the system for a while and returns, they might not remember where they were before leaving the system, which can cause problems because there the user’s task may have been uncompleted and they might unintentionally close the system and lose ongoing work; and, fourth, the logo and system brand can tell the user where they are located. Moreover, if the logo mostly conveys the purpose of the system, it may help the user know what to expect, but, if it is absent, a user might leave the system without exploring it. By informing the user who is running the system, the logo can also generate credibility and motivate the user to investigate the brand to know whether or not to use it; without knowledge of brand identity, however, the user may become suspicious and avoid using the system.

> No one wants to feel lost; if they feel lost, they wouldn’t know where they are or where they can go so they would close the system altogether. (P5)

> If you see a familiar logo, you would be more comfortable to buy from that site. For example, when you see the Amazon logo, you would be more certain that you will get your products vs if you saw a different logo, or no logo at all. . . . If I am on the homepage, the route I will take to get certain tasks done will be different than if I am on a different page; simply knowing the page they are in will eliminate the confusion. (P10)

> It might not be a big issue to know your location as long as you are still on the site, but imagine if you left the site for any reason and you got back; in this case, it is important to know where you are so you can know what you were doing. (P12)

While presenting the logo and the brand of the system is always advisable, there are some cases where it is not essential to show users their system location, most often when the system consists of a very small number of pages:

> I don't think you need to show me where I am in the system if the whole system is just two to three pages, although, while it is good to always do that, it won't be a big deal if it doesn't in that case. (P11)

Showing the user progress toward their goal is important for two main reasons: first, showing progress gives the user a feeling of control and users prefer to feel they are in control rather than feeling lost or uncertain; and, second, displaying user progress plays an important role in helping with decision making. If a process takes a long time, a user might decide not to perform it and, if they are doing a series of tasks and the current step is quite time-consuming, they might do something else and return once it is finished. In this way, showing progress allows the user to make efficient use of time:

> Showing the user the progress of their action, how close they are to getting to their goal . . . is very significant because the information about progress lets the user know what to do with their time. . . . It will make them seem more in control. They will have the choice as to whether to wait and do something else while waiting or stop and get back when they have time. (P4)

> If I know how long things are going to take, I can know what I am going to do with my time. (P13)

The only exception in the need to show progress is when an action takes very little time to accomplish; if it only takes a few seconds, for example less than 10 seconds, to complete, then it is not essential to display progress. One participant stated: "I think if the outcome appears very quickly, the need to show progress wouldn't be as necessary. I read that you need to show progress if the outcome is going to take more than 5–10 seconds" (P13).

Knowing when a goal has been completed is essential for two reasons: first, users typically use a system to accomplish specific goals, and if a user does not know whether or not they have accomplished their goal, the whole purpose of using the system is defeated; second, a user may think that an action has not been completed and might repeat it, which can have serious consequences especially in financial transactions.

> When you submit a form, and it just directs you to the homepage and doesn't tell if your response was received or not. This drives me crazy because I get confused. Do I submit it again? Or not? (P6)

> Letting them know what has gone right and what has gone wrong. . . . The user is going to question their action if we don't tell them if things went right. (P7)

> A popular example is when you make a transaction and don't know if it went through or not; that is extremely confusing and frustrating. It is even worse if they made the transaction again and it turns out that the first one had already gone through. (P8)

Match between system and the real world

Providing understandable content is a key factor for usable systems, because offering content that the user understands facilitates the learning process. If the user is unable to interpret what the system is attempting to communicate, then they either cannot use the system at all or will at least face difficulty in using it. If they misinterpret what the system is telling them, this can also lead to mistakes.

> If the elements in the system aren't shown to you in a way you understand, how would you know how to act? You are probably going to make mistakes because you are going to try different things till you get the outcome you want. (P3)

> I visited a site that was supposed to sell electronic parts. It refers to everything by very long serial numbers; it doesn't use normal and understandable explanations for people who are not very technical. I consider myself pretty much technical and, even for me, it was hard to know which type I wanted because there were no pictures to show how different parts look. (P6)

> If the user doesn't understand what you are communicating, then they will either quit right away, or try to explore the system to figure out what everything means, causing a lot of errors in the process. (P9)

> Match between system and the real world has a direct effect on learnability, since understanding the interface will lead to learning it faster. . . . I guess it is clear what the consequences of not using an understandable language are, simply not being able to do anything on the system. (P10)

The specific language used depends heavily on the target audience; what is understandable to one group of users might not be understandable to another group of users.

> Don't use technical words or jargon. Unless the target audience knows them (P8)

> If the system is meant to be used by older people, don't use slang used by teenagers. (P11)

Following a natural order for content and actions is important for two reasons: first, a user can understand and learn how to use the system more quickly, as a natural order takes advantage of the user's mental model and thereby facilitates the process of learning the system; and, second, avoiding a natural order might lead a user to make mistakes, as they may assume that the system follows a natural order.

> If everything was put in a natural manner then the user will not need to learn or recall anything. (P12)

> Making the action as natural as possible extremely reduces the possibility of any errors to occur since users would know what to do exactly. (P15)

However, following a natural order can be challenging if the system provides something unconventional or completely new. While system designers usually try to make the order as logical and intuitive as possible, in some cases the system may adopt a new approach to certain tasks in an order that, while not necessarily natural, is logical and easy to use, saving the user effort. One participant explained: "Make the flow as natural as possible; for any action in your system see how people do it in real life and imitate that. Don't do otherwise unless you can come up with something easier and intuitive" (P8).

Not using acceptable and appropriate content might make users feel offended, uncomfortable, or confused. Content appropriateness may also depend on the target audience and the nature of the system:

> Imagine visiting a government website or university website and they are using internet memes. That’s not appropriate. (P6)

> If the website is for children you shouldn’t use explicit language that is not appropriate for their age. This is going to be offensive. This is not about language only, but also pictures, metaphors, and so on. (P12)

User control and freedom

Providing a user with capability to reverse actions is important for four reasons. The first is to support learning; if a user can’t easily go back and forth and undo what they have done, the learning process may be severely compromised, because many people depend heavily on trial and error as a learning method. The second reason is to help in effective dealing with errors. Users sometimes slip and make unintentional mistakes and, if undo and redo are supported, they can usually easily recover from them. However, without such support, they might continue with undesired outcomes. The third reason is to help reduce stress and anxiety, keeping users from being afraid each time they interact with the system; if they cannot use undo and redo, they might be hesitant and uncomfortable every time they use the system. The fourth reason is to avoid embarrassing or risky situations in some specific cases.

> Without the ability to reverse the action, learning would be extremely hard, and no one wants to get stuck in a situation they don’t want; it is frustrating. (P3)

User control and freedom is to give the user the ability to make mistakes freely and without fear that these mistakes are going to affect them. The value of that is it makes performance less stressful and allows the user to get back and try different things other than their original plan. (P4)

When you start a wizard, you only have one path, you can't go back, or if they allow you to go back, you would lose the information you entered. In this case the user would feel that you took their control over the system. . . . By placing the user in control, you will make them less afraid to try new things on the system and they will be at ease, and it will minimize the cognitive load because they will not be thinking about the consequences. (P8)

If you don't provide the user with a way to undo their incorrect or accidental actions, they will get stuck with them, which obviously will lead them not to use the system again. (P9)

At the same time, user control and freedom also has a strong effect on learnability since it encourages the exploration of the interface. (P10)

User control and freedom is about being the owner of the system. If I am in a page and I want to leave it, I should be allowed. If I deleted something and want to get it back, I should be able to do that; the system shouldn't control me I should control it. . . . User control and freedom would reduce the stress, I won't be stressed every time I use the site, because I know I

> can undo everything I did on the site. How users feel when they use our site is essential for the success of it. (P11)

> I love that WhatsApp now allows you to delete texts you have sent. This basically what it means to support undone. It saves you embarrassment. . . . I can interact with everything freely because I know I can recover from it with no issue; this would encourage me to explore more. It would also make the errors less of a big deal because I can recover from it by just undoing them. (P12)

> This is where technology differs from the real world. In the real world most actions can't be undone, once it is done it is done. But let's compare it with when you teach your little brother or sister how to write some words; you make them try once and then they will get it wrong and erase it and try again. Undo and redo is the same thing in technology, it serves the same purpose. . . . User control and freedom is the cornerstone for the learning process. Not only that, it is important to reduce the stress and the anxiety of the user. (P13)

> User control freedom is to give the user the freedom to make decisions inside the website without the fear of making mistakes. (P15)

In some specific cases, and in some parts of the system, users might not be allowed to undo and redo their actions or there may be only a limited time during which a user can reverse an action. This can be a significant inconvenience in situations such as a survey where the questions must be answered in a specific order.

> Most of the food apps now allow you to cancel the order after it is placed, if that happens in a reasonable time. After that, you just can’t cancel the order, especially if the order was picked up by the driver. (P1)

Allowing the user to escape from any undesirable situation is important for two main reasons. First, it is possible that the user has completed an action by mistake or wanted to do something but has changed their mind. In such situations, a user should always be able to exit the system or a specific part of it; the user should always be in control of the system. Without such control, a user might feel frustrated and helpless. Second, if a user is using the system and wishes to maintain their privacy, but is unable to quit whatever they want, this might raise privacy issues.

> There are situations where you are writing something or watching something, and you don’t want others to look at what you are doing. Now if you can’t get out of that page or chat quickly when someone enters, it could lead to an undesired situation. (P13)

> The idea is that the user should be in control, so intentionally making hiding popups hard places the system in control instead of the user. . . . As a recent example, I subscribed to a newsletter and then after a while I got annoyed by their constant emails so I decided to unsubscribe. Usually there is a button in the email itself to allow you to unsubscribe. In this

> particular newsletter, it wasn't there. I spent almost an hour just to figure out how to unsubscribe. (P14)

Giving users sufficient information before asking them to take any action is important for two reasons: first, if users do not know why they are being asked for certain information, they are likely to be hesitant to respond and take action, perhaps eventually leading them to not use the system; second, it can also raise ethical concerns, as users should always know what you are doing with their data and why you are doing this. Since such concerns are limited to personal or sensitive information, it is not necessarily essential to provide this information in ordinary cases where a user is asked to enter normal information or in cases where the reason for the request is obvious.

> When a bank asks me "Do you have another passport?", I should know why they want to know that. Only when I know why I can decide whether to answer or not. . . . I am simply not going to use their service and I am going to look for alternatives. It is their ethical obligation to tell me how they are going to handle my personal information so I can decide whether to give it to them or not. . . . Now if they asked me to enter a username and password, I wouldn't care much, but as I said when they ask me something like if I have another passport or any other personal question, I would want to know how they handle this information. (P6)

Consistency and standards

There are three main reasons for system consistency. First, consistency makes learning how to use the system much easier. If a user has learned how to use one part of the system and the system is consistent in all its parts, there will tend to be few issues in using the other parts. An inconsistent system, on the other hand, might require a user to spend a significant amount of time learning each part. Second, consistency increases system memorability. If a user has stopped using the system for some time, they should have few issues in trying to return to it if the system is still consistent with its earlier structure. Conversely, if the system changes constantly, a user may have to learn the system all over again each time they return to it. Third, inconsistency might result in user mistakes. If a certain function button is located in one place on one page and located in a different place on another page, a user who is using the system rapidly might click the wrong button and cause unnecessary mistakes.

> Lack of consistency will result in so many errors. Users will automatically think that since two buttons aren't the same, then they must serve different purposes. This no doubt is going to cause confusion and mistakes. (P1)

> Consistency helps memorability. If you keep your system consistent even when you update it then it is going to help users know how to use it when they get back to it. (P3)

> If you ignore consistency you will end up with really confused users. They will constantly be asking themselves if this means the same as this. They

> will be a bit afraid to use the product because they don't know what to expect, but if it is consistent, then the time will take them to learn the system will be significantly reduced, and they will be more confident. (P7)

However, in some cases, consistency can be violated. If certain aspects of a system need to stand out, they might be presented in a way that is not consistent with the rest of the system: "Sometimes it's counterproductive to make everything standardized and consistent. . . . For example, if you want a certain button to stand out, you might violate consistency" (P8).

There are two reasons why standards should be followed. First, following standards facilitates the learning process, as a user will tend to know what to expect and how to easily interact with the system; if the system does not follow standard conventions, the user will take longer to understand how to use it. Second, following standards might help reduce the number of errors; if certain actions are performed in a conventional way, or certain elements have previously known meanings which the system does not follow, the user tends to make mistakes more often.

> Standards is to see how actions are done in most systems and copy that to your system to reduce learning time and increase familiarity with your system. (P1)

> I read a saying for Nielsen that says something like this: users spend more time on other systems than yours so take advantage of that. One of the things that would allow you to understand the system is its standards; if

> the system follows them, almost no effort will be needed to know how to act on it. (P4)

> Not following standards isn't always bad as I told you but following them in most cases will make learning the system much easier than using novel ways. (P9)

> Standards are about learning, you can see for yourself that any application that follows standards would be easier for you to learn than other applications that don't follow standards; while standards are not only about that, it will reduce the errors as well. (P12)

Still, in some cases, the system may not follow standards, perhaps because there are no agreed upon standards for certain aspects in the system or if they have created an easier and more intuitive approach:

> You shouldn't follow standards in everything, specifically when it comes to your brand. . . . We have internal procedures for our system that make our system unique and match our brand, so we don't follow the standards just in cases where we already have our way to do it. (P5)

> You shouldn't follow standards blindly. If you can get something better, and you have tested it on users and found out that it works better than the common way, then by all means do that. (P9)

Error prevention

The seven underlying ideas for this heuristic, as outlined above, all have the same purpose: error prevention. Preventing errors is important for two main reasons: first, errors extend the time it takes to complete a task; and second, errors can break the whole experience of a system and affect a user's mood, resulting in a negative impression of the system.

> Errors affect the user more than most people imagine. I remember reading in an article that errors have an emotional effect that could persist throughout the day; some errors have serious consequences. (P11)

> Errors reduce efficiency. Just look at the time lost when an error occurs, including the time it takes to recognize it, the time it takes to fix it, and to fix any input that was affected by the error. It is unnecessary wasted time. (P14)

Recognition rather than recall

Presenting all the information that users require directly and clearly is important for three reasons: first, it enables the user to accomplish their goal and reduces the time required to do so; second, by making the system useful, it means that users are less likely to seek alternatives than if they must remember all the information independently; and, third, it prevents errors, as users are more likely to commit mistakes if they are not presented with the information they require directly.

> It seems like an efficiency kind of thing if they recognize something. . . . Voice-conversational interfaces . . . are terrible for music because you need to say the band that you like and then you're like, "Alexa, play," so you're having to recall what it is. You don't get to talk about album art or get to see anything. I don't know my band's favorite song. (P2)

> Recognition rather than recall means to not force the user to recall information; this will extremely increase the efficiency of the system, and it is going to reduce the errors too. (P4)

> Going back to the example of letting the user knows what they are going to need for the form, if you don't let them know, they would either spend some time trying to remember or just put what they have in mind, and then they will make mistakes, try again, and so on. (P9)

> We know for a fact that human memory is limited, and we should design on that basis. If a user needs to recall all the time, then what is the value the site brings? (P11)

Offering user suggestions is crucial for two reasons. First, sometimes the system may include such complex aspects that it is impossible to present everything to the user in one display. In such cases, presenting the user with suggestions based on their history on the system or the input they are entering can facilitate the process of achieving their goals. This feature can decrease the extra time required by the user, improving efficiency. Second, without such suggestions, a user may commit mistakes, entering incorrect inputs because they are unable to correctly recall what they want.

> When you are on YouTube, and watching a video that has multiple parts, it facilitates the process by suggesting the second part of the video in case you forgot that. (P6)

> When you start to type anything, it gives you a list of suggestions. This is literally life-saving. So many times I just roughly remember the name and Google helped me to reach what I want. I think this should be applied in any app . . . suggestions would make you extremely efficient. As I told you, Google search is extremely efficient; it takes me less than a second to get what I want. That also means less errors; imagine if these suggestions aren't there, I would try five to six times or more to get what I want. That's a lot of errors. (P12)

However, it is important to note that, if suggestions offered are not accurate, they may distract the user from accomplishing their goal; it is key that suggestions be as accurate and appropriate as possible. One participant stated: "But it is extremely annoying and distracting if the suggestions have nothing to do with what you're browsing. Like if you're looking for a phone and it swamps you with all sorts of different electronics on the side" (P10).

Flexibility and efficiency of use

Flexibility is important for two main reasons. First, most systems are not designed for only one group of users, but for multiple types of users, and providing different methods to perform the same task can accommodate all users. If the system only provides one approach to completing a task, and it is aimed at a particular group of users, other groups might feel excluded and quit using the system. Second, since most systems are not designed to be used only in one situation, system should be usable in different situations, and providing users with alternatives enables them to choose what works the best for them under specific circumstances; if a system fails to do this, it will most likely be replaced by other systems that better meet user needs.

> The user should be able to quickly accomplish their goal regardless of their level of experience. . . . So, the system should accommodate multiple types of users, otherwise they will limit the use of the system only to users who can or know how to use it. (P7)

> Having multiple options available to the user so they can interact with the system in the way that suits them. . . . For example, they can search or browse by categories. . . . If I am not busy, I will browse; if I am busy and I want to get what I want quickly, I'll search. (P8)

> If I am a novice, can I use the system? If I am disabled can I use the system? To me this heuristic means you should accommodate everyone you expect to use your system. It is like having different views of the system; if I like the dark mode or the bright mode, both should be offered. . . . As I said before, you don't always approach the system in the same mood, and not one group only is going to use the system, so if you don't

> provide alternatives, you are going to limit the user of the system to a specific group only. (P11)

> It means you respect your users and you respect their time. Why do you force me to sign up if I can do it as a guest? Users' time is valuable, and they would switch to competitors. I literally didn't purchase from apps that require me to sign up. (P12)

However, there are two exceptions to flexibility: first, if the task is very simple and requires very little effort in terms of thinking and action, it might be acceptable to provide only one way of completing it; and, second, if a system is used infrequently for a very specific application for a short time, it might be understandable or even necessary to limit the ways in which the system can be used.

> I wouldn't expect the system to offer many options if the task at hand is very simple and easily learned. In this case, it might be even better to just keep it simple. (P11)

> It makes sense that in things like ATMs, for example, which you don't use frequently, to not offer so many options and to standardize the way you use it so that you don't confuse users and make it easy to remember how to do it. (P14)

System efficiency is important for two reasons. First, given that time and effort are very important factors in using technology and users are always looking for ways to accomplish their goals as quickly as possible and with minimum effort, it is always important to ensure that all tasks are performed in the easiest and most efficient way possible; and, second, as the technology sphere is highly competitive, it is vital to make systems as effortless to use as possible, otherwise users will immediately switch to other, more efficient systems.

> The main point of technology is to make things more efficient. Allow us to do things with less time and effort than without it. (P1)

> A product that takes 17 steps to reach checkout is obviously not efficient, do you can imagine what will happen; they will go to any competitor. (P7)

However, an exception to this is that in some specific systems, the actions taken are intentionally made challenging, for example in video games. These systems require different measures to assess the efficiency of the actions:

> Really the concept of efficiency is system dependent. As a general rule, you want the system to be easy and fast, but in specific cases that might not be desirable. . . . When you have a sensitive system where you don't want anyone to easily use it, or for example video games where you make things challenging on purpose, in these cases, you might have other measures for efficiency. (P14)

Aesthetic and minimalist design

Making a system attractive is important for two main reasons: first, an easy-to-use and beautiful system can add positivity to the user experience; and, second, the "aesthetic-usability effect" suggests that the more aesthetically pleasing, the more usable the perception. Thus, a more beautiful system might lead users to be more forgiving with respect to usability issues or other difficulties, making them more likely to use the system:

> Aesthetic is simply making the system appear in a beautiful and appealing way. This one is a bit hard to measure. It might differ from person to person, age group to age group, culture to culture, and so on, but it is known that beauty has an effect on how users perceive the system. If it is beautiful, they will perceive it as more usable. It is the aesthetic-usability effect. (P1)

> I personally use some apps even though I know I have better alternatives just because I like the design. . . . Imagine you put so much effort in the site but you didn’t put the same effort to make it beautiful, and if a competitor has similar or slightly less usability but has a very beautiful look, then this could be the tie breaker and users will use their site instead of yours. (P11)

However, while it can help if a system is aesthetically pleasing and an aesthetically-pleasing system might lead users to forgive minor issues, appearance is not going to compensate for major usability issues. Additionally, given that aesthetics is more subjective than any of the other heuristics, it is not always possible to accurately judge the system on this basis:

> But making it beautiful shouldn’t be the main focus. Plus, users would be forgiving to some extent; if the system isn’t functional or has major issues, they are not going to use it even if it is beautiful. (P1)

> It is a bit subjective to measure the attractiveness of a system, but as I said, it has an impact; I prefer some apps based on the design. (P11)

Presenting content in an organized and simple manner is important for three main reasons. First, it can help the user better understand the system; if a system is disorganized, users may have to spend a significant amount of time attempting to decipher the information that is being presented to them, which can make the system more difficult to learn and potentially result in the user switching to other systems that they can learn to use more quickly. Second, even if a user knows a system and is prepared to use it, overcrowded content can distract the user from accomplishing their goal, which may slow down the user and affect their efficiency. Third, a crowded, cluttered system can induce user mistakes, as users may click incorrect buttons unintentionally; reducing clutter as much as possible can therefore reduce errors, enabling users to interact easily with the system without making mistakes.

> Good design makes the interaction quick as long as we can see or understand how to utilize it. (P2)

> You need to be focusing on what’s relevant and important to the user. Provide what’s necessary only so you don’t distract the user. . . . Example, an app that started very simple then suffered from scope creep and has all these cool features but wasn’t efficient anymore because it takes so much effort to do simple tasks. (P7)

> Simplicity will ease interaction with the system. Too many features will confuse the user. (P9)

> No one would use a site that is not organized, basically because I will not be able to understand it correctly. Also, the busyness of the page will make me make mistakes. (P11)

> What is the point of putting in so much content or features if your users are going to use only some of it? When you do that, you just distract them and confuse them. The best sites are the ones where you can quickly get what you want. . . . Simple design makes everything clear. I can know what is being offered, or what is the main purpose. I remember visiting many sites where I spent 10 minutes just trying to know what they offer or what they are all about and I wasn't able. (P12)

Help users recognize, diagnose, and recover from errors

Recognition is important because a user who does not know that they have made a mistake will often continue to repeat what they are doing, which can potentially lead to more significant problems; an action taken on an incorrect basis will always produce an incorrect result. It is therefore vital to let users know when something has gone wrong, so they can stop and examine the problem: "The problem of not knowing that they made a mistake so they are going to continue as if nothing happened, meaning that anything they will do after that is going to be wrong. It makes the problem worse" (P9).

There are two reasons to enable users to identify errors. First, knowing that an incorrect action has been taken can allow the user to rectify the action; by avoiding the need to review every action that has been completed to identify the problem, this can save the user time. Second, a user might change a correct input, thinking that the problem lies with it, which can lead to multiple problems instead of only one problem.

> If the user doesn't know what they did wrong, they will go and change something else thinking it is the problem, and end up with two problems. (P9)

> Without knowing how to recover from errors, I might get stuck, and I might also make the mistake again simply because I don't know what it was in the first place. (P11)

> To solve a problem, you should first know what it is. If you don't know that, it would be almost impossible to solve it. (P15)

There are two reasons for providing users with steps to recover from errors: first, not all problems have obvious solutions, so users need actionable, step-by-step methods to guide them in solving an issue; and, second, a user who does not know how to solve a problem is likely to waste time testing out different ways to achieve a solution, which can lead to manipulating aspects that were correct in the first place, introducing further problems.

> The same thing here: if I don't know how to solve the problem, I might go back and change things thinking I am solving the problem while I am changing something that was correct and I end up with two problems. (P9)

> There should be some guidance on how to solve the problem if the solution isn't intuitive. (P12)

Still, while it is always important to provide users with actionable solutions, this might not be needed if the required solution will be obvious for the user. One participant explained: "But if the solution is intuitive then no need for the guidance because users are going to figure it out" (P12).

Help and documentation

Providing help for users is important for four reasons. First, given that documentation cannot possibly cover all problems, users should be offered the option to contact an appropriate entity to explain their unique situation and receive advice on how to solve it; without this option, users might experience serious issues. Second, as users do not always read written documentation, there is a need to provide other means of help; without this support, users might become confused or even decide to never use the service again. Third, some users are still uncomfortable in dealing with technology independently and feel more comfortable in dealing with humans, so providing real time help can make users feel more at ease and safer when using the system. Fourth, even if users trust the system, conversations with advisors can still feel more natural to many users, helping to make the experience as smooth as possible.

> We need to emphasize that people don't read. You need to give the option to call you or email you. Unfortunately, it is still the most convenient method for users till now. . . . I stopped using a food delivery app because there is no customer service I could contact. Most of the time that's OK because I usually receive my food just fine, but when the food is late or when there is a problem in the order, I have to go read all the documentation to see how to solve the issue, and you know when people are hungry, they can't function properly. (P11)

> In most cases it is OK not to have support, but if the system is complicated and users don't have enough support, they either might make a lot of mistakes or will quit the system altogether. (P9)

> The first thing users will do when having a problem or when having a question is to try and contact the support; it just feels more natural than going through all the material, but the material still should be there. (P6)

> People feel more comfortable and will trust your app or system more if they know they can contact a human being who can address their concerns. I think it is extremely important to offer that. (P12)

There are three reasons for providing proper system documentation. First, given that it is not always easy to know how to use the system, there should be documents that explain how to use it; without such documents, users might be hesitant to try interacting with the system. Second, a lack of documentation can lead to errors; if a user does not know how actions should be performed, which can be difficult to learn, they will often try learning through a process of trial and error, which might cause them to make mistakes. Third, it is not always easy to know all features the system offers, and documentation can serve as a way to explain them to the user; without documentation, users might stick only to what they already know and never gain the opportunity to experience some valuable features offered by the system.

> When I want to learn how to use something on the site, or when I am stuck at something, I should find the material to get me through the issue. I should find the material easily. (P6)

> You can think of it as the catalog of the system. Everything that is related to the system should be written down, including things like how to use every function, what the available options are, what everything means. The main reason for this is to show me how to learn to use the system and prevent errors. . . . There should be a separate page I can go to, to learn more about the system itself because I, as a user, don’t always know the full capacity of the system. This could introduce me to features and functions I don’t know about. . . . Documentation is advantageous mostly to show your users what it is that you offer. Basically, like a presentation. It happened to me before where I actually discovered things about the system about the app that I didn’t initially know. (P11)

However, documentation is not always needed. If the system is small and relatively easy to use, then it may be sufficient to only provide help for unique cases, meaning that documentation is unnecessary. One participant stated: “Again, as the system gets more complicated, the more you need to provide help and support" (P9).

3.6.5. Mapping heuristics to seven stages of action and usability components

There are two main reasons to perform heuristic evaluation: the first is to ensure that users are able to accomplish their goal using the system being evaluated, and the second is to ensure that users can accomplish their goals on the system with relative ease. Heuristics should therefore be mapped to seven stages of action and to usability components, to inform the evaluator about how these heuristics can help users accomplish their goals more easily. We performed this mapping based on results from the second part of the interviews on heuristics components and significance (Figure 14 - Figure 15).

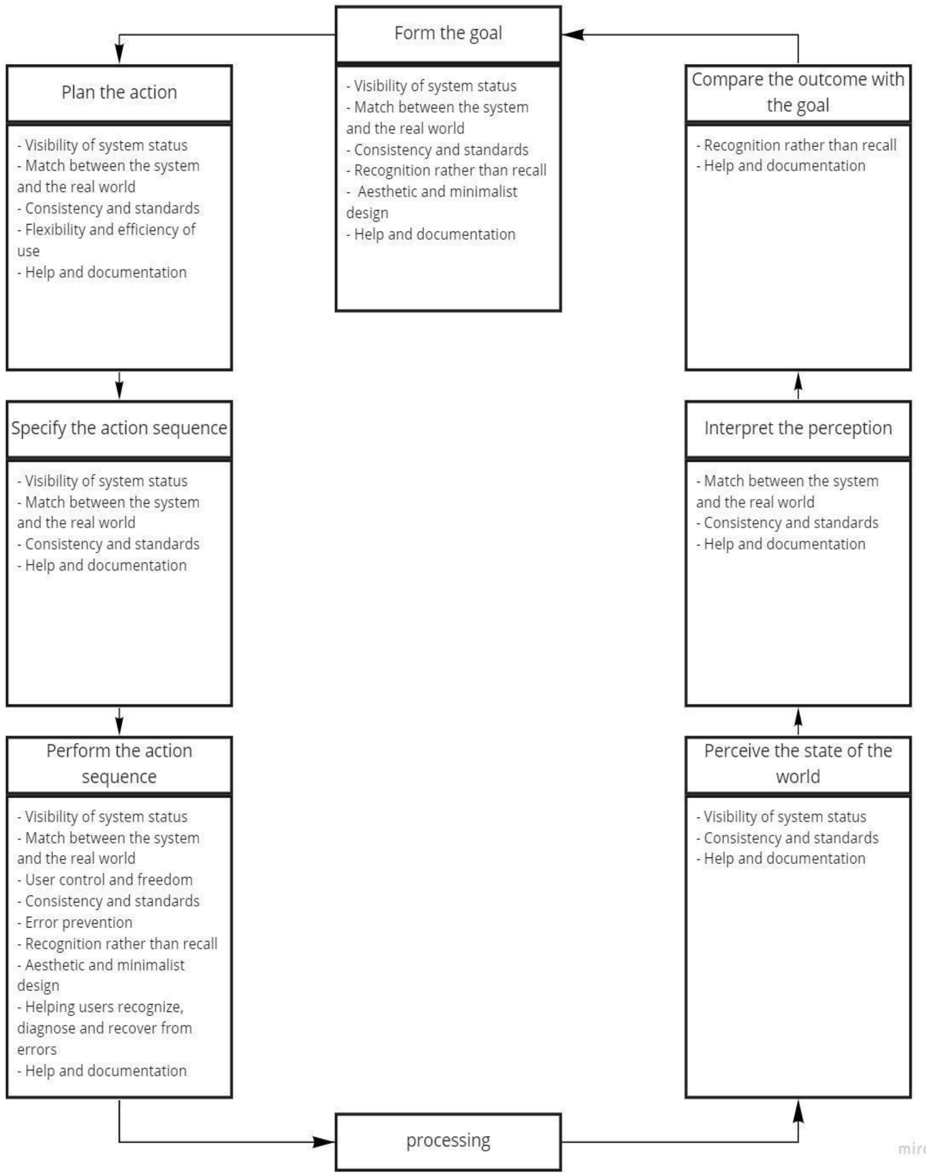

Figure 14. *Mapped to seven stages of action*

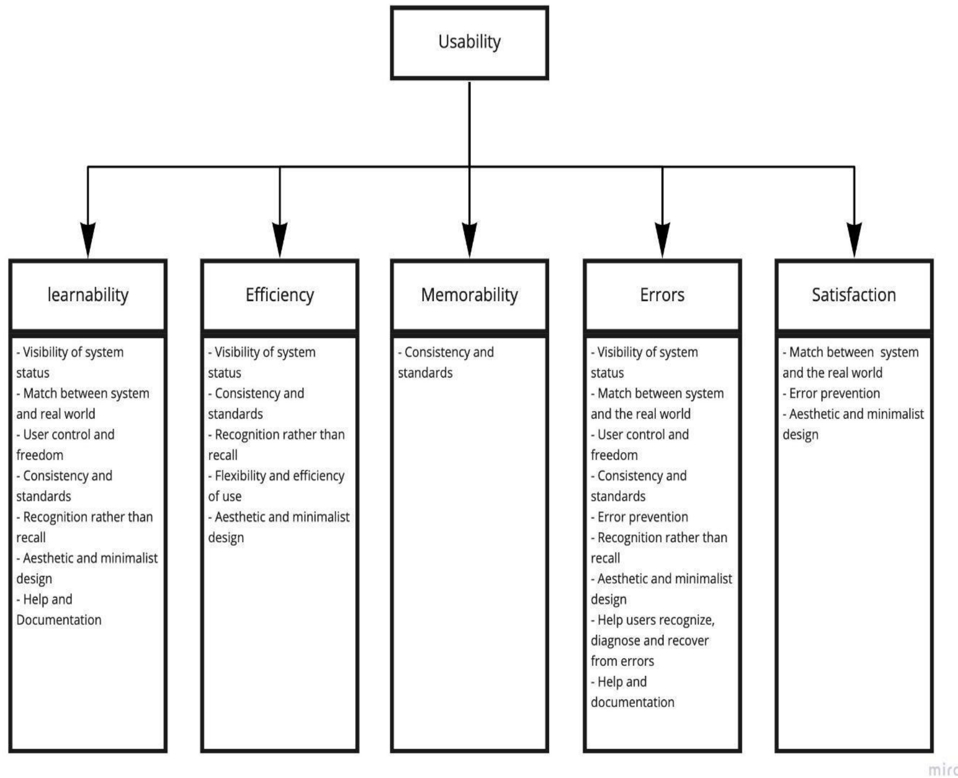

Figure 15. *Mapped to usability components*

Visibility of system status

With respect to seven stages of action, visibility of system status plays an important role in five out of seven stages. In the first stage, forming the goal, if a user is unable to determine the system state, they are not going to be able to decide whether or not they can pursue that goal. In the second stage, planning the action, if users do not know their location, they will not be able to decide which path will be better in achieving their goals. In the third stage, specifying the action sequence, if users know their location, they will know something about how many additional steps are required to achieve their goal, because the number of steps will vary depending on the location. In the fourth stage, performing the action, displaying progress, and giving user feedback about changes in system state is crucial for successful implementation. In the fifth stage, perceiving the state of the world, if the user does not achieve a sense of closure, showing whether or not their action was successful, the user will not know what they should do next.

For usability components, visibility of system status plays a role in three out of five components. For learnability, knowing the system state is essential to ensure that the user is aware of what the system can and cannot perform. For efficiency, progress increases use of users' time; when a user knows how long a certain action should take, they will be able to decide how to spend their time. Knowledge of location also increases efficiency because users can choose the best path for accomplishing their goal based on their present location. Finally, by informing users, knowledge of system state also helps reduce errors.

Match between system and the real world

Match between system and the real world is involved in five out of seven stages. First, in forming a goal, system information should be presented in an understandable way to the user. If jargon or complex technical language is used, the typical user will not be able to understand what is being offered by the system. Second, in planning the action, matching the natural and logical order of the actions facilitates the process of choosing which option will be adopted by the user. Since a user will assume specific flow for the option, it is important to follow the natural order. Third, in specifying the action, the user, after picking an approach to performing the action, will specify the action sequence, so it is best if the action follows a logical or natural order; otherwise, users will face some difficulties in this stage because they lack knowledge about how actions can be performed. Fourth, in performing the action, information should be presented in an understandable and clear manner; information clarity will facilitate the process of getting from one step to another without too many issues. Fifth, in interpreting perception, changes in system state should be presented in a way that the user can easily interpret; if users notice a change in system state but are unable to interpret its meaning, they will feel lost.

Match between system and the real world is involved in three out of five components. For learnability, if system content is presented to the user in a manner they can understand, and if the actions a user will perform in the system follow a natural order, users will not require as much effort to learn how to use the system, because it will come to them more intuitively. For errors, using content that users understand and presenting action in a logical or natural way reduces the chances of users making mistakes. For satisfaction, using content appropriate for the target audience reduces the risk of offending users and increases the chances that users will be satisfied.

User control and freedom

User control and freedom is involved in mainly one out of the seven stages. It is involved in performing actions where people learn by trial and error; when performing an action, the user should be allowed to undo and redo their actions and they also should be able to quit the whole process at any time. Moreover, while performing an action, if they are asked to enter any personal or sensitive information, they should know why this needs to be entered and how this information is going to be handled.

User control and freedom is involved in two out of five usability components. In relation to learnability, given that one of the most widely used methods for learning is trial and error, the system should allow the user to experiment freely and make mistakes in order to learn. Support of undo and redo will enable the user to try new things on the system without fear, facilitating the learning process. In relation to handling errors, allowing the user to reverse any action means that recovery from unintentional mistakes can be accomplished.

Consistency and standards

Consistency and standards play an important role in most stages. First, in forming the goal, following standards with regards to the way information is presented on the system eases the process of recognizing what is being offered. Second, in planning the action, if the system follows well-known standards, the user will have more information on what each option entails and how it works. Third, in specifying the action, if the system follows standards, a user will be better informed on how many steps the action requires. Fourth, in performing the action, following the standards can make the process of executing any system action easier for users, because they will know how to behave and interact with the various elements of the system properly. In addition, by using the various elements of the systems consistently, users will avoid constantly questioning their meaning and will be able to interact with it more smoothly. Fifth, in perceiving the state of the world, following standards can facilitate the process of knowing that something has happened. Finally, in interpreting perception, consistency and standards play an important role in interpreting feedback; if the system follows standards, users will know what is being presented to them. Moreover, if users have used the system or parts of it before, and the system is consistent in the way it presents information, they will have no problem in understanding the feedback.

Consistency and standards also play an important role in most of the usability components. For learnability, by following the standards, the user will know what to expect when using the system. Since they have probably used similar systems before, the process of learning how to use a system becomes easier when standards are followed. For efficiency, by following standards and maintaining consistency throughout the system, users will be able to accomplish their goal quickly. For memorability, if the system remains consistent, a user returning to the system should know how to act on it without issues. Furthermore, if issues arise, it will be easier for the user to get back on track with minimal effort if the system follows standards. For errors, by following standards, users should not face any issues in navigating the system, leading to reduction in their number of errors.

Error prevention

There is only one stage in which error prevention plays a role: performing the action. When performing a sequence of actions, a user might be prone to making mistakes, which may frustrate the user and lead to the action taking more time than it should to be completed. As such, the system should, as far as possible, prevent the user from making mistakes.

Error prevention plays a role in two out of five usability components. In errors, if the system does not allow users to enter invalid inputs, guidance performing the actions should be provided, including autosave, flexibility for user input, and asking for user confirmation when executing important actions. This guidance will help or prevent users from making a significant number of errors. In satisfaction, by preventing errors from occurring, users will have more positive experiences with the system as errors frustrate users.

Recognition rather than recall

Recognition rather than recall is important in three out of seven stages. First, in forming the goal, all information should be available at the user interface. Sometimes users forget their purpose for coming to the system and, by presenting all needed information on the interface, the system can facilitate the process of recognizing the goal, rather than forcing the user to recall it. Second, in performing the action, during each step toward completing the action, any necessary information should be presented to the user in the interface, because forcing users to recall information required during the process might hinder their ability to complete the task; they may fail to recall this information or take additional time to complete the task. Third, in comparing the outcome with the goal, to make this comparison easier, it is not sufficient to use generic language, such as the task was completed; the initial goal should be stated explicitly so that a user can know exactly which goal has been successfully completed.

Recognition rather than recall plays a role in three out of five usability components. For learnability, if everything the user needs is available to them, then they will be able to learn the system more quickly, as they will not need to spend as much time trying to resolve their uncertainties. For efficiency, if everything the user needs is available at the interface, the user will not need to spend time or effort attempting to remember anything and will be able to accomplish their goal as fast as possible. For errors, providing users with real-time suggestions when interacting with the system helps them to accurately enter and choose the right decision, helping to minimize mistakes.

Flexibility and efficiency of use

Flexibility and efficiency of use is involved in mainly one out of seven stages: planning the action. Once the goal has been formed, users will want to know the possible means for accomplishing their goals. By providing more than one option, the system will provide more freedom for the user. Efficiency also plays a major role in determining the option a user is going to choose based on their time, level of expertise, and situation.

Flexibility and efficiency of use are also involved in one out of five usability components: efficiency. By giving users multiple options for performing the same action, the system can accommodate different types of users and situations. Moreover, requiring the user to spend only the minimum possible effort to perform the tasks will improve system efficiency.

Aesthetic and minimalist design

Aesthetic and minimalist design plays a role in two out of the seven stages. First, in forming the goal, the interface should present content in a clear and organized manner; if the user is not able to clearly see what the system offers, they will not know if their needs can be met. Moreover, a busy page can distract the user from their goal. Second, in performing the action, system content should be organized, well-ordered, and include no extraneous information, to facilitate the process of moving from one step to another.

Aesthetic and minimalist design plays a role in four out of five usability components. For learnability, if the content presented in the system is well-organized and well-ordered, the user can easily navigate and interact with the system with minimum effort, enhancing learnability. For efficiency, by having a clear and simple interface, a user can navigate easily and perform tasks quickly and efficiently. For errors, a cluttered interface might cause the user to click on unintended buttons, while a clearly organized and simple interface will help a user avoid such unintended mistakes. For satisfaction, as users perceive aesthetically pleasing systems as more usable, as specified by the aesthetic-usability effect, it is important to focus on making a system aesthetically pleasing to increase user satisfaction.

Help users recognize, diagnose, and recover from errors

Help users recognize, diagnose, and recover from errors is involved in one out of seven stages: performing the action. If the user makes mistakes during the process of executing a goal, the user should be able to easily recover from it and return to executing the steps.

Help users recognize, diagnose, and recover from errors is involved in one out of five usability components: errors. When errors happen, it is vital to make the user aware of the error, understand its nature, and know how to recover from it. In this way, the system can help the user recover and avoid repeating the same mistake again.

Help and documentation

Help and documentation plays an important role in each of the seven stages of action. First, in forming the goal, while it might sometimes be difficult to list all information clearly on a homepage or landing page, clear and comprehensive documentation should be provided to help the user know relevant information about the system, informing the user whether or not the system provides the desired function. The system should also provide some method of contact so that the user can reach out for help, to gain answers to specific questions about what is being offered. Second, in planning the action, the user might not know all the options for accomplishing the goal or might not have sufficient information to determine the best approach, so there should be high-quality material to explain how to achieve the goal. If users still have questions, they should be able to contact someone who can provide further explanations. Third, in specifying the action, if for some reason an action is complicated and needs to be performed in a novel and unconventional way, documentation should explain exactly how each action can be performed in a step-by-step manner. Moreover, the user should be able to make contact with advisors to learn how to perform an action. Fourth, in performing the action, if a user has faced difficulties or obstacles during execution and does not know how to handle them, they should be able to find relevant information on how to overcome such challenges. A user should also be able receive real time help, in case they are unable to find the information needed in the documentation. Fifth, in perceiving the state of the world, if a user is not able to determine whether or not something has changed on the system, they should be able to contact someone to consult about the system state. Sixth, in interpreting the perception, if users are unable to grasp feedback provided to them, they should be able to find the desired meaning in the documentation or contact an advisor who can explain the result. Seventh, in comparing the outcome with the goal, if the user is still unsure as to whether the goal has been accomplished, they should be able to obtain the information from the documentation or contact an advisor for further help.

Help and documentation plays a role in two out of five usability components. For learnability, by providing sufficient and clear documentation, FAQs, and tutorials, the user should be able to learn to use the system in a short amount of time. For errors, if provided with understandable and actionable documentation, a user will know how to adequately use the system to minimize the number of errors.

CHAPTER 4. COHERENT HEURISTIC EVALUATION (COHE)

Informed by the analysis of experts' interviews, as presented in the previous chapter, we developed a step-by-step protocol called coherent heuristic evaluation (CoHE) (Figure 16).

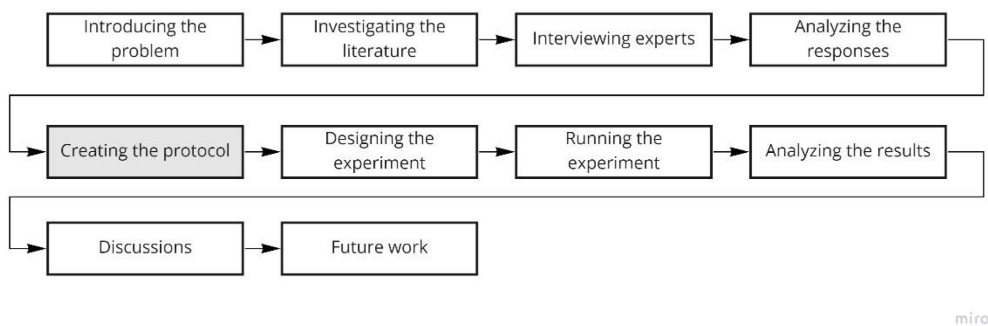

Figure 16. *Thesis organization (Chapter 4)*

The goal of this effort is to produce a standalone protocol, meaning that it can be read and understood by novices without external help; in line with the purpose of discount usability methods, it requires relatively small amounts of money, time, and effort. To this end, we tried to exclude suggestions of asking for external help in the protocol, such as adding experts to the process to help novices or adding other evaluators to help novice evaluators with finding usability issues. We also excluded any suggestions of taking special courses or reading complete books to seek explanation of the underlying principles of HCI/UX to help in conducting HE. Finally, we also excluded suggestions of undertaking other HCI/UX activities, even if they might help in the process, since those activities are difficult and require extensive study and practice to be performed correctly so will most likely overwhelm novices.

Moreover, some of the valuable suggestions we received during the interviews do not feature because they require further research beyond the scope of this project, such as providing a list of the most common problems for each heuristic, and prioritizing or reordering the heuristics based on the number of issues they found; both of these suggestions require further examination and statistical analysis of a number of heuristic evaluation reports to yield accurate findings. We also excluded presenting evaluators with specific aspects for each heuristic, as the aspects we received were few. Instead, we narrowed inspection to very few aspects that engender missing potential usability problems; further investigation is required to develop a more substantial list of aspects. Finally, given that it was difficult to identify an agreed-upon label for each heuristic, we decided to maintain Nielsen's original labels.

CoHE involves three stages. The first stage is the understanding stage, during which the evaluator begins to understand heuristics and why they should be applied by receiving an explanation of their seven stages of action and usability components. The second stage is the inspection stage, during which the evaluator receives step-by-step guidance on how to start inspecting the system and how to use heuristics to detect usability issues. The third stage is the documentation stage, during which the evaluator is given tips on how to map usability issues to usability heuristics and how to accurately rate the severity of usability issues discovered. In addition, the evaluator will be given advice on how to prepare the final report during this stage. Figure 17 – Figure 20 show a high-level overview of CoHE stages.

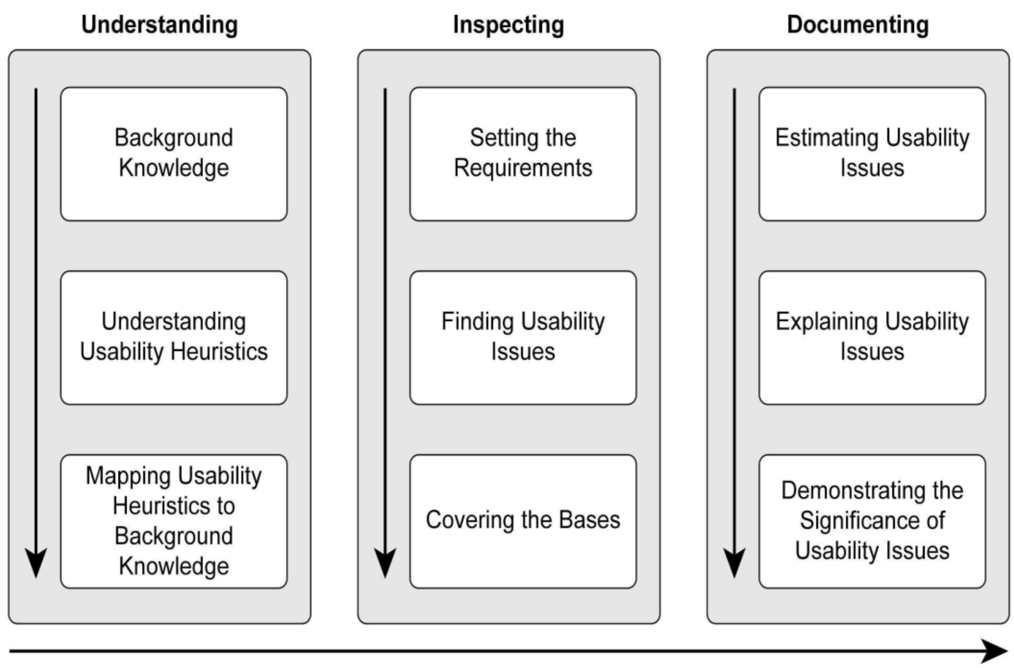

Figure 17. *CoHE*

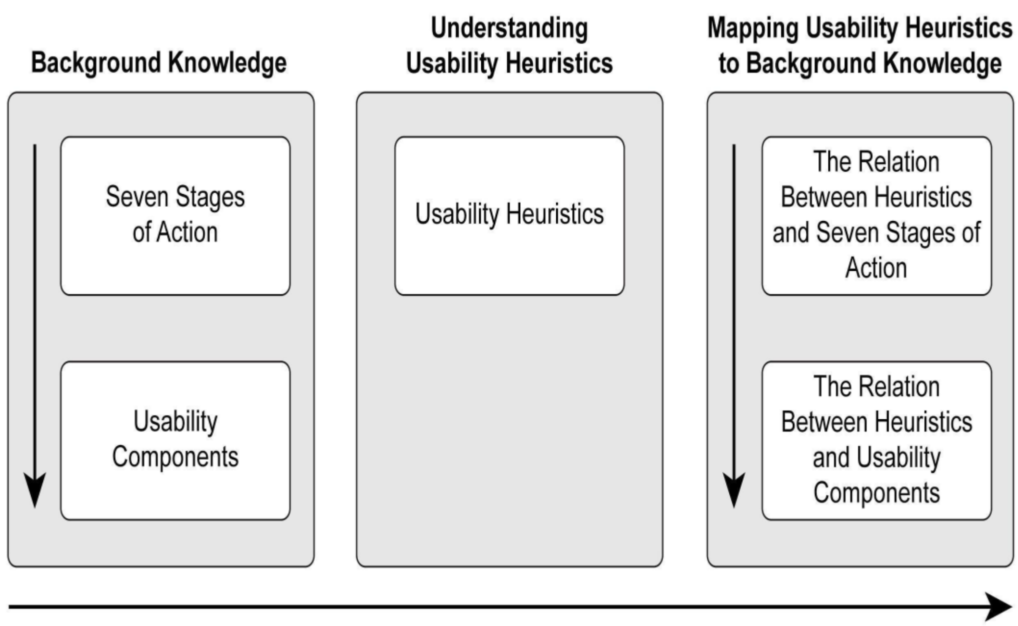

Figure 18. *Understanding*

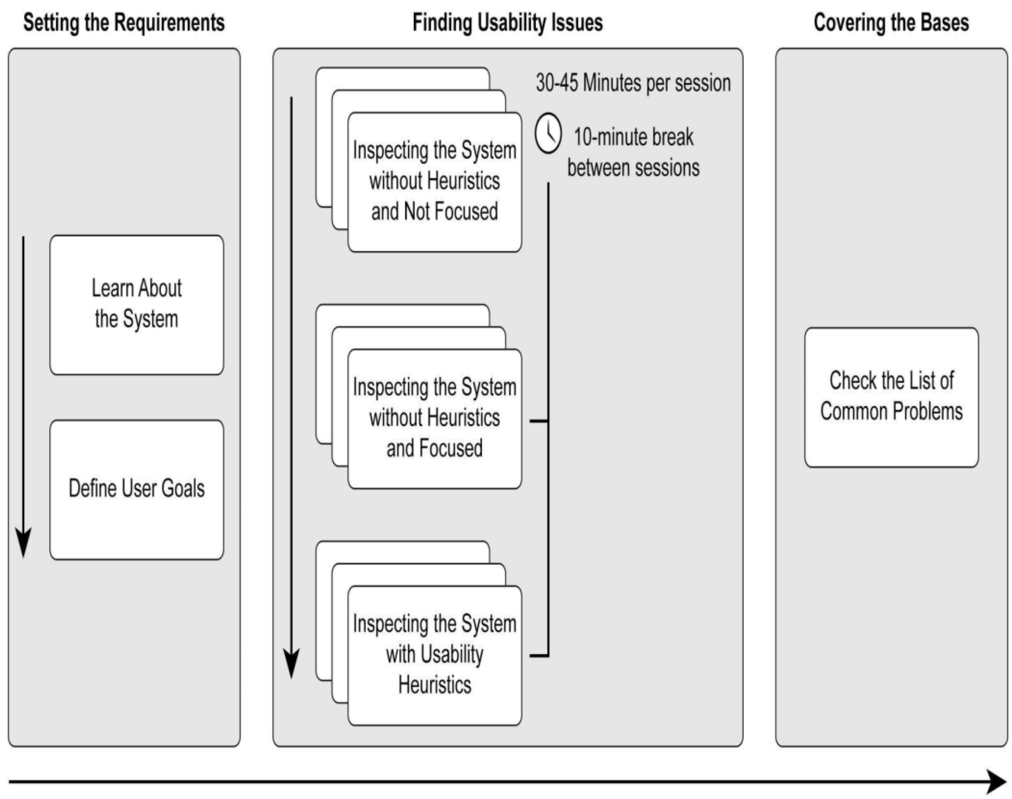

Figure 19. *Inspecting*

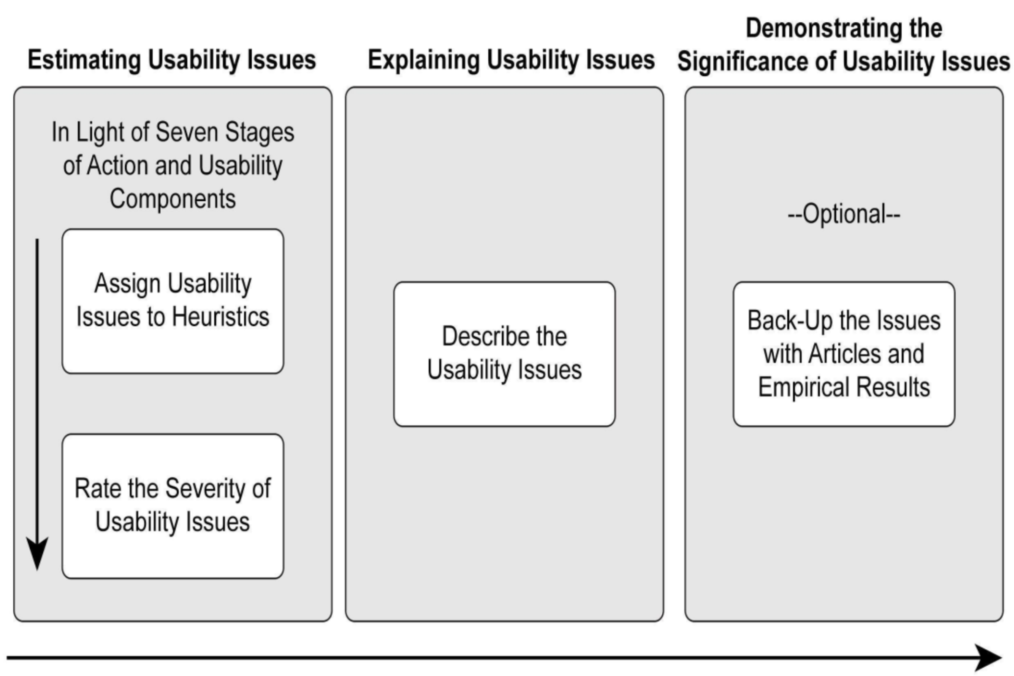

Figure 20. *Documenting*

4.1. Required material

There are multiple steps in each stage of CoHE and specific activities required to successfully complete each step. During the understanding stage, the evaluator should have a list of usability heuristics so that they are aware of what is going to be used for the evaluation. They should also have lists of the seven stages of action and the usability components as well as their relations to the heuristics. During the inspection stage, evaluators should have a list of the heuristics and related questions, so that they can ask themselves the questions which need to be answered by the system. Finally, during the documenting stage, an evaluator should have a checklist, to remind them what needs to be included in the final report.

4.1.1. Detailed usability heuristics

While Nielsen's usability heuristics are currently the most popular set of heuristics, they are fairly abstract and require additional detail and explanation. Table 3 provides a detailed list that expands on these usability heuristics, including the heuristics, the underlying concepts of the heuristics, the components of each concept, the significance of these components, and the applicability of each component. This list will provide the evaluator with a better and deeper understanding of the usability heuristics.

Table 3. *Detailed usability heuristics*

Usability Heuristic	*Concepts*	*What makes it work?*	*Why does it matter?*	*Exceptions*
Visibility of system status	State: the user should know the state of the system and what they are capable of doing in the system at any given time.	- There should be a visible and easy to interpret indicator to inform the user about the state of the system. - Any changes in the state of the system should be reflected immediately.	- Lets the user know what to expect and what to do. - Reduces time since the user does not need to spend time to figure out the system state. - A lack of this quality will give user false hope and possibly lead to errors.	
	Location: the user should know which part of the system they are in, i.e., their location in relation to other parts of the system.	- There should be an indicator to tell the user where they are in relation to other parts of the system. - The indicator should be clear and visible. - The indicator should be present throughout the whole system.	- Knowing where they are helps the user know where to go. - The steps a user takes to accomplish any goal is mostly dependent upon where they are in the system. - When the user leaves the system for a while and returns, they might not	- There are some cases where it is relatively unimportant for a user to know where they are located in the system, most often when the system consists of a very small number of pages.

			remember where they were before leaving the system. - The logo of the system tells a user where they are located and also conveys the system purpose and quality.	
	Progress: a user should know how far they are from accomplishing their goal.	- The indicator should be appropriate (time, capacity, steps). - The indicator should be accurate. - The indicator should be present both in active and passive situations.	- Makes the user feel in control. - Helps the user make decisions about how to best use their time.	- The only exception to showing progress is when the action takes little time to accomplish. If it takes only a few seconds, generally less than 10, it is relatively unimportant to show progress.
	Closure: the user should know that the task at hand has been completed and whether it has been completed with desirable outcomes.	- This should be offered immediately after completion. - This should be offered no matter what the results are. - This should be clear and understandable. - If the action was not successful, an explanation should be offered.	- Not knowing whether the goal is completed defeats the purpose of the system. - If the user does not know whether a goal is completed, they might repeat the action.	

Match between system and the real world	Understandability: the content presented on the system should be understandable for the target audience.	- The content should be understandable for the target audience, with the target audience being the key word.	- Not understanding the content makes learning how to use it difficult. - Mistakes can occur if content is not understood.	- It should be emphasized that understandability depends heavily on the target audience. Something considered understandable to one group of users might not be considered understandable to another.
	Natural and logical order: the content and actions presented on the system should follow a logical or natural order.	- The action sequence should follow the same order as similar tasks in the real world. - If this is not possible, the action sequence should be logical and intuitive.	- Following natural or logical order makes learning faster. - Not following natural or logical order cans lead to errors.	- Following natural order can be challenging if the system operates unconventionally or in a completely novel way. In such cases, while it is possible that the natural order should not be followed, effort should be made to make the order as logical and intuitive as possible. - In some cases, the system could suggest a new way to do certain tasks in an order that may not seem natural,

				but it should still be logical and easy to use.
	Appropriateness: the content presented on the system should be acceptable and appropriate to the target audience.	- The content should match the system identity and purpose. - The content should be appropriate and avoid being offensive.	- Using inappropriate content could offend the users.	- The appropriateness of the content also depends on the target audience and the nature of the system.
User control and freedom	Reversibility: the user should be able to undo and redo any action performed on the system.	- The user should be able to reverse any action at any time and any place. - Reversing actions should be easy. - Users should be able to go back as far as they want in reversing the action.	- Reversibility capability facilitates the learning process. - It helps in effectively handling errors. - It helps reduce the user's stress and anxiety. - Inability to reverse actions could be risky and embarrassing.	- Sometimes the system intentionally does not allow the user to reverse some actions, for example in a survey in which the questions should be answered on a specific order. - Sometimes, a user should be given a limited time to reverse an action.
	Emergency exit: the user should be able to escape from any undesirable situation in the system.	- The user should be able escape any situation, regardless of time and place.	- A user might have performed an action and changed their mind or made a mistake, so escape from	

		- It should be easy for the user to exit.	such a situation should be possible. - The user should feel that they are in control of the system. - Lack of an emergency exit can cause privacy issues.	
	Informing users: before asking the user to enter any input or take any action, the user should be presented with enough information to enable them to make an informed decision.	- Why information is being asked for should be explained. - How information is going to be stored and handled should be explained.	- If the user does not know why they are being asked to provide certain information, they might be hesitant to use the system. - There is also an ethical obligation to provide this context, particularly if the information being requested is sensitive.	- A reason only needs to be provided for sensitive and personal information. - If the reasons for asking for personal or sensitive information is obvious, it might not be necessary to provide an explanation.
Consistency and standards	Consistency: once a certain element of the system is used in one place of the system, it should be presented in the same way throughout the whole system.	- Meanings should be consistent. - Functions should be consistent. - Organization and layout should be consistent. - The effort should be consistent.	- This makes learning the system easier. - It also increases the memorability of the system. - Lack of consistency can lead to errors.	- If certain aspects of the system should stand out, this should be presented in a way inconsistent with other aspects of the system.

		- The feeling should be consistent.		
	Standards: the design of the system should follow common practices and conventions of similar systems.	- Follow standards in meanings. - Follow standards in action sequences. - Follow standards in organization.	- This facilitates the learning process. - It helps in reducing errors.	- If there are no agreed-upon standards for certain things on the system. - If there is an easier and more intuitive approach than the current standards.
Error prevention	Instructions: give users sufficient guidance before taking any action to avoid making errors.	- This should be visible. - This should be understandable - This should be concise.	- Errors makes task completion longer. - Errors will negatively affect the user experience and mood.	
	Constraints: placing constraints on some types of inputs that are clearly invalid save user time and effort.	- Give users hints on why they are being constrained.	- It makes task completion longer. - It will negatively affect the user experience and mood.	
	Confirmation: asking users to confirm an action before completing it to ensure that they want the action to take place.	- Make sure the user knows that you are asking for confirmation.	- It makes task completion longer. - It will negatively affect the user experience and mood.	

Notification: the user should be notified about any changes in the system, especially those with serious consequences.	- Tell the user clearly what the notification is about. - Explain the consequences of ignoring the notification. - Only notify the user about important changes.	- It makes task completion longer. - It will negatively affect the user experience and mood.	
Auto-saving: user inputs should be saved in case something has gone wrong. In such a case, the user will avoid losing the effort they had expended.	- Let the user know that their input is being auto-saved. - Saved input should be retrieved easily or automatically.	- It makes task completion longer. - It will negatively affect the user experience and mood.	
Flexible inputs: the user should not be forced to enter input in a certain form; they should be able to enter the input in any form they wish.	- Conversion from form to form should be clearly visible to the user so they can assess its accuracy.	- It makes task completion longer. - It will negatively affect the user experience and mood.	
Defaults: the default state of the system should be used carefully to prevent users from making mistakes.	- The defaults should be the most expected things. - There should be an indication as to the scope of the default state.	- It makes task completion longer. - It will negatively affect the user experience and mood.	

Recognition rather than recall	Availability: anything the user will need to accomplish a certain goal on the system should be presented to them; the user should have to rely on their memory as little as possible.	- All information should be clear to the user. - If there is a multi-step task, the information needed should be presented at each step.	- This increases system efficiency. - Users should not be expected to recall everything. - It helps in reducing errors.	
	Suggestions: it is not possible to know what every user wants to accomplish on the system; to facilitate the process, the user should be provided with suggestions.	- Provide the user with suggestions when they start from an empty state. - Give the user suggestions when they are searching. - Provide the user with suggestions of similar or related things when they are browsing. - Suggestions should be as accurate as possible.	- This increases the efficiency of the system. - It reduces errors.	- It is key that suggestions be as accurate as possible; if they are not, or cannot be, then they should not be offered.
Flexibility and efficiency of use	Flexibility: every major goal or task on the system should be accessible and capable of implementation in more than one way.	- It should accommodate all types of users who are expected to use the system. - It should accommodate all expected situations of the system use.	- Most systems are designed to be used by multiple users, and if a system only accommodates one group of users, other groups will feel excluded.	- If a task is very simple and requires very little effort in terms of thinking and action, it is acceptable that it can be completed in one way only.

			- Most systems are not going to be used in just one situation. A system that cannot be used in different situations will lead users to seek alternatives.	- If the system is not used frequently, or only used for a very specific application, it might be necessary to limit the ways in which the system can be used.
	Efficiency: any system goal or task should be performed in the simplest way possible.	- There should be no extraneous steps in an action. - The effort required to complete an action should be minimal. - The time required to complete an action should be minimal.	- One of the main goals of technology is to make achieving user goals faster; therefore, the system should be as efficient as possible. - If the system is not fast and efficient, users can find alternatives with just one click.	- The actions taken are intentionally made challenging in some systems, such as video games; these have different measures for assessing efficiency of actions.
Aesthetic and minimalist design	Aesthetic: the design of the system should be aesthetically pleasing to the target audience.	- Visual elements should be used appropriately and carefully. - All elements in the system should be in harmony. - Audiovisual materials should be presented in an attractive manner.	- The beauty of the system adds to the positivity of the experience. - If the system is beautiful, the user will most likely be more forgiving with respect to usability issues and difficulties.	- Aesthetic is important, but it should not be the main goal of the system. - It should be noted that, while aesthetics might make users forgive minor issues, it will not make users forgive major issues.

				- Aesthetic is to some extent subjective, so it might be hard to assess.
	Organization: system content should be presented in an organized manner.	- All related elements should be organized and grouped together appropriately. - Different groups of elements should be distinct from one another.	- This makes the system easy to learn. - It will increase efficiency. - It will reduce user errors.	
	Simplicity: content presented by the system should be limited to necessary content.	- Any extraneous content should be removed. - Any content that could distract the user should be removed.	- This makes the system easy to learn. - It will increase efficiency. - It will reduce user errors.	
Help users recognize, diagnose, and recover from errors	Recognizing errors: the user should notice when an error occurs.	- Error indication should be presented in a way that the user will expect.	- If users do not know that they have made mistakes, they may repeat what they are doing, leading to a more serious problem.	
	Understanding errors: the user should understand	- Error information should be readable.	- The first step to rectifying an error is to know what it is wrong.	

	exactly what error has been made.	- It should be easy to interpret. - It should be presented in a way that matches the system's purpose. - Information should not be presented in a way that blames the user. - Information should not be presented in an intimidating manner.	- A user might change a correct input, thinking that the problem lies with it, which is going to create multiple problems instead of only the original one.	
	Recovering from errors: the user should know how to recover from any specific error.	- Recovery information should be presented in an actionable manner.	- If the user does not know how to solve the problem, they will waste time testing out different ways of solving the problem. - This may lead the user to change actions that were correct in the first place, doubling the issue.	- While it is always important to provide users with actionable solutions, this might not be needed where a solution is obvious.
Help and documentation	Help: the user should be able to contact a support person if they face difficulties or have questions.	- There should be multiple methods for providing help. - The expectations for each method should be stated clearly.	- Documentation cannot possibly cover all the problems that users might have on the system.	

			- People do not always read the documentation. - Some users are still uncomfortable in dealing with technology alone; they feel more comfortable in dealing with humans. - Even if users trust the system, talking to an advisor may feel more natural to many users.	
	Documentation: the whole system, or at least its most important aspects, should be documented and presented to the user in either written or visual form.	- The material should be complete. - It should be easy to understand. - It should be accessible. - It should be searchable. - It should be relevant. - The content of the material should be prioritized. - The content of the material should be categorized.	- Documentation helps in learning the system. - Without it, some users might be hesitant to use aspects of the system. - It can reduce the number of user errors. - It serves as a means to inform the users on what the system offers.	- If the system is small and easy to use, documentation may not be needed; it may be enough to provide help only for the unique cases.

		- Audiovisual documentation should not be too long; there should be audiovisual support for each problem or related problems. - Contextual documentation should be presented adjacent to frequent actions.		

4.1.2. Mapping seven stages of action to usability heuristics

Since one of the main goals of evaluation is to ensure that users can accomplish their goals on the system, it is crucial to know how humans accomplish goals in general. Moreover, understanding how adhering to usability principles will facilitate the process of accomplishing the goal and deepen the evaluator's understanding. Table 4 provides a list that shows each stage of action, and the role heuristics play in each.

Table 4. *Usability heuristics stages of action*

Stage of Action	*Heuristics Involved*
Form the goal (What do I want to accomplish?)	- Visibility of system status: knowing the state of the system helps the user know whether they can use it to pursue a specific goal. - Match between system and the real world: if users do not understand the content, they will not be able to know whether or not what they want is being offered. - Consistency and standards: by following standards, users can know how to navigate the system to see whether the goal they want is offered by the system. - Recognition rather than recall: when users open the system, they might forget or inaccurately remember what they originally wanted to do; if they are provided with full information and offered suggestions, they may remember the goal they originally wanted to accomplish. - Aesthetic and minimalist design: if the system is simple and organized, users can quickly know whether their goal can be accomplished. - Help and documentation: if users are unsure as to whether or not their goal can be accomplished on the system, they should be provided with either the material that explains the system so they can know this or the opportunity to consult with the support team.

Plan the action (What are the alternative action sequences?)	- Visibility of system status: because the best action will depend on where the user is located in the system, knowing their location will help users know what specific action to take. - Match between system and the real world: users will assume that the system follows a natural order, so they will base their planning on that assumption. - Consistency and standards: by following standards, users will know alternative options for accomplishing the goal. - Flexibility and efficiency of use: if the system provides different options, each with a specific sequence, users will be able to choose the option most suitable and efficient for their situation and their level of expertise. - Help and documentation: if users are unsure about possible ways to accomplish a goal, they should be provided with either explanatory material or consultation with the support team.
Specify the action sequence (What action can I do now?)	- Visibility of system status: by knowing where they are located in the system, the user will know the required number of steps and their sequence. - Match between system and the real world: following natural order will facilitate the process of listing the steps necessary to accomplish a goal. - Consistency and standards: by following standards, the user will know the steps required to accomplish the goal. - Help and documentation: if users are not sure of the necessary steps to perform a certain action, the documentation should give an explanation, or they should be permitted to contact the support team.
Perform the action sequence (How do I do it?)	- Visibility of system status: showing users the progress they are making is important in helping them make good use of their time while performing an action. - Match between system and the real world: clarity and understandability of content will help users in successfully moving from one step to the next. - User control and freedom: while performing an action, users should be able to reverse any action, exit from any undesirable situation, and if asked to enter personal or sensitive information, they should know why this is required and how it is going to be handled. - Consistency and standards: by following standards, users will know how to interact with the various elements they encounter, and if the system is consistent, users will learn from each step how to behave in subsequent steps.

	- Error prevention: since errors affect users' ability to accomplish their goals, the system should prevent users from making errors while performing an action. - Recognition rather than recall: while users are performing actions, all information that they need should be provided at each step; users should not be forced to recall this. - Aesthetic and minimalist design: having a simple and organized system means that the move from step to step will be easier and a user is less likely to become distracted. - Helping users recognize, diagnose, and recover from errors: since users might make mistakes while performing steps, recognizing mistakes, understanding them, and knowing how to recover from them is essential for getting back on track and accomplishing a goal. - Help and documentation: if a user is faced with an obstacle they do know how to overcome while performing an action, or if they have a certain question, either the documentation should provide an answer or they should be able to get the answer from the support team.
Perceive the state of the world (What happened?)	- Visibility of system status: providing the user with a closure will let the user know whether or not something has changed. - Consistency and standards: by following the standards of showing the users that something has changed, the user will most likely understand that something did happen. - Help and documentation: if the user is unsure whether or not something has happened, they should be provided with an answer either in the material or from the support team.
Interpret the perception (What does it mean?)	- Match between system and the real world: system changes should be presented to the user in an understandable manner. - Consistency and standards: if the system follows standards in the way information is presented, and if there is consistency in the way information is presented, a user should have no problem in interpreting what is being presented to them. - Help and documentation: if a user does not understand or cannot interpret what is being presented, they should be able to find an explanation in the documentation or from the support team.
Compare the outcome with the goal (Is this ok? Have I	- Recognition rather than recall: after finishing a goal, the user should be reminded exactly what has been accomplished that they might have forgotten; the user should not be forced to recall everything they have done.

accomplished my goal?)	- Help and documentation: if the user is still unsure whether or not a goal has been accomplished, they should be able to gain this information from the documentation or by contacting a support advisor.

4.1.3. Mapping usability components to usability heuristics

Users not only want to achieve goals, but they also want to accomplish goals as easily as possible. As such, systems should be usable as well as functional; an evaluator who understands usability components and knows how usability heuristics can be used for evaluation can help in reaching that goal. Table 5 presents a list which explains the role of usability heuristics in each of the usability components.

Table 5. *Usability components*

Usability Component	*Heuristics Involved*
Learnability	- Visibility of system status: without knowing the state of the system, a user will not be able to learn what they can and cannot do in the system. - Match between system and the real world: having understandable content and a natural flow of actions is key to learning; if a user does not understand the content, they will not be able to learn it. - User control and freedom: since trial and error can be a key to learning, users should be allowed to try things and reverse them as they wish. - Consistency and standards: following standards makes the learning process easier for users because they can use their previous experience with other systems; if the system is consistent, learning one part of the system will help in learning the other parts of the system. - Recognition rather than recall: suggestions and availability of information will ease the user's learning process; without them, the user might be forced to try many different ways of learning.

	- Aesthetic and minimalist design: simple system organization will make learning its use significantly easier because users will be less distracted and more able to identify the options. - Help and documentation: a user should be able to consult the documentation to understand the system and know what it offers.
Efficiency	- Visibility of system status: not knowing how long a certain action will take will not support using time efficiently; knowing location can help a user take the shortest path toward a goal. - Consistency and standards: following standards will help a user to quickly accomplish a goal; being consistent in knowing how to deal with one part will help a user to deal with the other parts more quickly. - Recognition rather than recall: suggestions increase the efficiency of the system because the user may only need to remember a few words; information availability will also lead to accomplishing a goal more quickly. - Flexibility and efficiency of use: accommodating different users and situations, and making each task as simple as possible, will increase system efficiency. - Aesthetic and minimalist design: simplicity in organization of the system will make goal accomplishment more efficient; if a user is distracted or cannot distinguish between the different system components, accomplishing goals will take longer.
Memorability	- Consistency and standards: if the system maintains consistency each time a user uses it, it will be much easier to quickly remember how to use it.
Errors	- Visibility of system status: not knowing the state of the system might cause a user to misinterpret what they can do in the system, possibly leading to mistakes. - Matching the system to the real world: users will assume a natural flow of actions, and if this is not the case, they might make mistakes; if the content is not understandable, the chance of making mistakes will also be higher. - User control and freedom: reversing any action will help users to handle errors efficiently as they will not get stuck with the consequences of their errors. - Consistency and standards: following standards and being consistent in each part of the system will significantly reduce the chance of user errors. - Error prevention: designed mainly to reduce the number of errors while using the system.

	- Recognition rather than recall: if users are forced to recall all required information, their limited memory may result in mistakes until they remember correctly. - Aesthetic and minimalist design: if the system is cluttered and crowded, users might mistakenly press an unwanted button or icon; by organizing the system and making it simple, the user will not face such an issue. - Help users recognize, diagnose, and recover from errors: errors might occur even with error prevention, so knowing how to handle them is vital in reducing the chances of errors persisting or recurring. - Help and documentation: providing opportunities to contact support advisors and ask them questions will reduce the chance of users making mistakes; by reading documentation and thereby learning more about the system, a user is also less likely to make mistakes.
Satisfaction	- Matching the system with the real world: using appropriate content will help increase user satisfaction. - Error prevention: committing errors can have an adverse impact on a user's mood, so preventing errors will enhance the user experience and increase user satisfaction. - Aesthetic and minimalist design: attractive systems will increase users' satisfaction and make them more forgiving of usability issues.

4.1.4. Operational usability heuristics

An evaluator must evaluate the system against the set of usability heuristics. Table 6 presents a list that can help to facilitate the process of detecting usability issues, providing an evaluator with the key questions to ask themselves.

Table 6. *Operational usability heuristics*

Usability Heuristic	*Concept*	*Questions*
Visibility of system status	State	- When looking at the different elements on the system (links, buttons, etc.), do you know what you can do with them? - When any changes happen in the system, are they immediately reflected in the system?

		- Do changes happen immediately or do they take time?
	Location	- Do you know in which system you are located, and can you determine this easily? - Do you know the page you are on? Can you find out easily? - Do you know where you are in relation to other parts of the system? Can you do this easily?
	Progress	- When you work on an action, does the system tell you how long it will take for completion? - When you complete a multi-page task, does the system tell you how many steps are left? - When there is a limited capacity in the system (storage, memory, etc.), does the system tell you how much is left? - Are progress measures accurate? If it says it takes one minute, does it really take one minute? - Are progress both active and passive measures provided?
	Closure	- When you finish a task, does the system provide you with feedback? - Does the system give you such feedback if you got either a right or a wrong result? - Does such feedback appear immediately, or does it take a long time? - Can you easily interpret the feedback? - If what you did was wrong, does the system offer an explanation?
Match between system and the real world	Understand-ability	- Is every piece of system content (text, icons, images, etc.) understandable? Specifically, by the target audience?
	Natural and logical order	- Do the steps required to complete a task follow a natural order, i.e., do they follow the order of how the task would be done in the real world? - If not, are the steps required to complete the task logical, i.e., can you predict what the next step should be?

	Appropriate-ness	- Is every content element in the system (text, icons, images, etc.) appropriate? Do they match the identity of the system? - Is there an aspect of the system that might be offensive to the target audience?
User control and freedom	Reversibility	- Can you undo any action (deleted, sent, placed, etc.) you have performed on the system? - Can you redo any action (deleted, sent, placed, etc.) you have performed on the system? - Are undo and redo easy to do? - How many steps back can you undo or redo?
	Emergency exit	- Can you escape from any situation in the system? - Is this easy to do?
	Informing users	- When the system asks you for personal or sensitive information, does it tell you why this is wanted? - Does the system tell you how this information is going to be stored and handled?
Consistency and standards	Consistency	- Does the system always refer to the same element with the same name across the system? - Do elements that appear identical across the system always do the same thing? - Does the interface layout and organization have a similar appearance across the system? - When you work on a multi-page task, is the same effort needed for each page, i.e., is there consistency in the effort needed for each page? - Do the multiple parts of the system have the same feel, i.e., do you perceive the different parts of the system as belonging to one unit?
	Standards	- Do you feel that your past experiences with similar systems helped you in using this system? - Does this help you in understanding content meaning? - Does it help in performing actions on the system? - Does the system organization look similar to that of other similar systems?

Error prevention	Instructions	- Does the system provide you with instructions on how to complete a certain task that requires a specific action to be performed? - Are instructions clear and easy to understand? - Are instructions clearly visible? - Are instructions too long?
	Constraints	- In situations where the action/input is clearly wrong or will lead to erroneous outcome, does the system prohibit that action/input from taking place? - When you are prohibited from performing a certain action/input, do you have an idea why?
	Confirmation	- When committing to a major action that has a long-lasting impact, does the system ask you to confirm the action?
	Notification	- When an important or serious event occurs, does the system notify you? - Does the system tell you what is going to happen if you do not take a recommended action?
	Autosaving	- When you are entering/writing an input that takes much time or effort, does the system automatically save your work? - Do you know when your input is being autosaved? - Is input retrieval easy or automatic?
	Flexible inputs	- When the system asks you to enter an input that comes in different forms (date, phone number, weight, etc.), does it allow you to enter it in the form you want? - When it takes a form you like and convert it, can you see the conversion?
	Defaults	- When you are in an empty state, are the system defaults the expected ones? - Is there any indication of exactly what the defaults are?
	Availability	- Is everything you would need to accomplish your goal clearly presented to you?

Recognition rather than recall		- When completing a multi-step task, is the information presented in one step and also needed in other steps presented to you?
	Suggestions	- When you are in the empty state, does the system provide you with suggestions on how to proceed? - When you search for something within the system and do not exactly recall it, does the system provide you with suggestions? - When you browse the system, does it provide you with suggestions of things similar to your browsing targets? - Are these suggestions accurate?
Flexibility and efficiency of use	Flexibility	- Does the system provide you with different paths toward accomplishing the same goal? - Will users with a variety of abilities and skills be able to use the system?
	Efficiency	- Could the steps/time/effort required for accomplishing a goal be reduced?
Aesthetic and minimalist design	Aesthetic	- Is the system aesthetically pleasing to you? - Do the interface elements appear to be in harmony? - Are audiovisual materials well-presented in the system?
	Organization	- When looking at the interface, are related elements organized in a way that shows their relationships? - When looking at the interface, could you easily distinguish among the different elements (menus, paragraphs, etc.)?
	Simplicity	- Is there any extraneous content on the system (features, icons, texts, etc.) that could be omitted? - Is there anything in the interface that distracts you from properly focusing?
Help users recognize, diagnose and	Recognizing errors	- When an error occurs, do you notice it? - Is the error indication the expected one, or did it take some time to notice it?

recover from errors	Understanding errors	- When you notice an error, do you know exactly what it is? - Can you easily read it? - Is error information written in an appropriate way? - Is it written in a way that might intimidate or try to blame you?
	Recovering from errors	- When you know of an error, do you know how to resolve it? - Is the solution presented to you in an appropriately actionable manner?
Help and documentation	Help	- Does the system provide you with capability for contacting the support team? - Are there multiple ways to contact the support team, or only one? - Do you receive an explanation of how long it will take to get a support team response?
	Documentation	- Is there documentation from which you can find solutions or learn more about the system? - Can you find all possible solutions? - Is the material easy to understand? - Can you easily find the material? - Can you easily search within the material? - Is the material prioritized by action frequency? - Is the material categorized? - If there are audiovisual tutorials, are they too long? - Is contextual documentation displayed next to the major tasks?

4.1.5. Usability report checklist

Since the most important element of the evaluation process is the report, which enumerates all findings of the evaluation, it is important to make the report as precise and intelligible as possible. Figure 21 serves as a checklist to remind the evaluator of all items that should be included in the report so that it will have the intended impact on the reader.

Report Checklist
• A problem number. • A problem name. • A problem description (what the problem is, why it is a problem, and how it is going to affect the user and the business?). • A problem screenshot. • Problem severity. • Usability heuristic/s violated. -- Optional -- • Articles showing the significance of similar issues. • Empirical studies to describe the consequence of such problems. • Any other material (text/audio/video) that could help in explaining the problem and its consequences. • Describing the problem in light of the seven stages of action and usability components. -- Tips -- • List the problems from the most severe to the least severe. • Be consistent in the way you list all the problems.

Figure 21. *Report checklist*

4.2. Understanding

The first stage of CoHE is the understanding stage, which consists of comprehending three points: first, how people go about accomplishing their goals, and what makes any system usable, to enable users to easily accomplish their goals on the given system; second, what the usability heuristics are in detail; and, third, how usability heuristics can help in the process of easily accomplishing the goals. The steps of this stage are as follows:

1. Reading and understanding Norman's seven stages of action to develop a more precise understanding of how humans go about accomplishing their goals.
2. Reading and understanding Nielsen's five usability components to achieve a better understanding of what makes any system usable, and how its users can accomplish their goals easily.
3. Reading and understanding usability heuristics to understand each concept, each component, its significance, and applicability, and providing examples to gain a better understanding of the elements to be used in evaluating the given system.
4. Reading and understanding the mapping of usability heuristics to Norman's seven stages of action to achieve a better understanding of how usability heuristics play a role in facilitating the process of accomplishing user goals.
5. Reading and understanding the mapping of usability heuristics to Nielsen's five usability components to achieve a better understanding of

how usability heuristics are involved in making accomplishment of user goals easier.

4.3. Inspecting

The second stage of CoHE is the inspecting stage, which involves knowing the several aspects of conducting an inspection: first, how the system should be approached; second, how much time should be allocated for the inspection; third, how many sessions should be allocated for the inspection; and, fourth, in what mood the inspector should be. The specific steps of the inspection are as follows:

1. Before beginning the inspection, the evaluator should know the target audience of the system and have general information about it, such as what the system is and what it is intended to do.
2. Since any given system is designed to meet some user goals, before starting the inspection, these goals should be identified and listed.
3. Before the inspection, the evaluator should plan to allocate between 30 to 45 minutes for each session, unless the system is small and likely to take less than this amount of time. If the system is extensive with many user goals and likely to take a long time to inspect, it is better to divide the evaluation into smaller sessions, each lasting for 30 to 45 minutes.
4. Begin inspecting the system without using usability heuristics. The evaluator should start by trying to accomplish predefined user goals on the system and writing down problems encountered. Along with predefined

user goals, several specific pages should be separately examined: the home page, the contact us page, the about us page and the FAQ page.

5. Evaluation without referring to usability heuristics should be performed twice, the first time when tried to imitate an actual user situation and a second time when fully alert to be critical. It is preferred that these steps be performed in this order to avoid learning effects that could occur during the alert/critical session.
6. Each usability problem found during these two sessions should be written down without rating its severity or without trying to link it to any usability heuristic. It should be written down along with some notes to provide a reference point for remembering the nature of the problem.
7. After the first round of inspection without the heuristics, the evaluator should conduct a second round using usability heuristics. The same predefined goals, along with the additional pages, will be inspected under the guidance of the usability heuristics.
8. As in the first session, the evaluator should document the usability problems found should along with severity ratings. Moreover, since these usability problems will have been found based on the heuristics, they will be linked to the appropriate heuristics, but only as an initial mapping.
9. Finally, if there is a list of common problems, the evaluator should examine the system to see whether or not these problems are found in the system, serving as a double-check review procedure. If any are found, they should be written down.

4.4. Documenting

The third stage of CoHE is the documentation stage, during which the evaluator prepares a document describing all usability problems found during their inspection. Preparing such a document requires several activities: first, the problems should be linked to usability heuristics; second, each problem should be rated in terms of its severity; and third, each problem should be described in a way that explains its significance. The steps involved in this stage are as follows:

1. The evaluator should assign each usability problem found to one or more heuristics. To develop a more accurate assignment, they should look to determine each stage of action affecting the problem, then locate all heuristics involved with this stage. Then, they should examine the usability component that the problem affects, followed by locating all heuristics involved with this component.
2. Each problem should next be rated in terms of its severity, keeping in mind the seven stages of action and related usability components. When the stage of action and usability components affecting the problem is known, the evaluator can determine the problem severity.
3. When reporting each problem, the evaluator should include a detailed problem description, providing screenshots that support this and offering recommendations on how to address the problem.
4. To make the description of the problem clearer and more detailed, the evaluator should explain it in terms of the seven stages of action and usability components. In other words, they should explain exactly how the

problem is going to hinder accomplishing a goal and the usability components it is likely to affect.

5. Finally, to add more depth to the report, it is recommended that the evaluator adds empirical studies, articles, and descriptions of similar problems if available.

CHAPTER 5. EXPERIMENT FOR TESTING COHE

The main goal of this study is to develop a means to improve the quality of evaluation by novices conducting HE. We can divide the process of reaching this goal into two parts: the first part, as presented in Chapters 3 and 4, involved developing a step-by-step protocol based on the knowledge, experience, and suggestions of expert evaluators; the second part, as addressed in this chapter, involves assessing whether or not the protocol successfully improves the performance of novice evaluators, using an experiment designed for this purpose. This chapter outlines the steps taken to examine the effects of CoHE on novice evaluation (Figure 22).

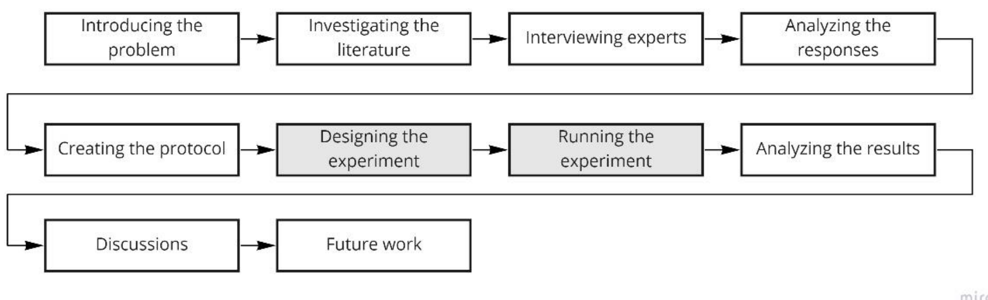

Figure 22. *Thesis organization (Chapter 5)*

5.1. Introduction

The main objective of the experiment was to measure the difference between using parts of CoHE, namely detailed usability heuristics and operational usability heuristics, and using traditional HE, in terms of the number of usability problems found, the severity of the usability problems, and validity, thoroughness, and effectiveness. The specific questions to be addressed are as follows:

RQ1. Do parts of CoHE, the detailed usability heuristics and operational usability heuristics, and traditional HE differ in terms of time, of the number of usability problems found, the severity of the usability problems, and validity, thoroughness, effectiveness and f-measure?

RQ2. Does the level of confidence of novice evaluators with respect to accuracy of their results differ when they use parts of CoHE compared to HE?

RQ3. What difficulties do novices face when conducting different parts of CoHE?

RQ4. How do the difficulties faced by novices when conducting parts of CoHE differ from those faced by novices when conducting HE?

RQ5. Does novices' judgment of their comprehension of usability heuristics differ when they use parts of CoHE compared to HE?

5.2. Design

To answer these questions, we designed an experiment to compare two groups of novice evaluators, with one group conducting evaluation using parts of our proposed CoHE protocol and a second group of novice evaluators conducting evaluation using traditional HE. The experiment also involved a benchmark group consisting of expert evaluators to help compare the groups.

5.2.1. Participants

The experiment involved two types of participants: novice evaluators and expert evaluators.

As discussed above, defining an “expert” is difficult, because the term has no standard definition, both within HCI/UX and across a broad range of fields. In line with the definition adopted to define expertise for the interviews, we chose to define a usability expert for this experiment as someone with at least 4 years of experience in the field (de Lima Salgado et al., 2016, pp. 2931–2946).

Defining a “novice” is arguably simpler and more intuitive; it is usually someone with little or no experience in the field, certainly less than 4 years. In this specific context, we defined a novice as someone with a general idea about the meaning of usability who works in the domain of computer science, information technology, software engineering, or related fields, but has never previously conducted HE outside of classroom settings. We chose this definition in an attempt to make the experiment as realistic as possible. Ideally, HE should be conducted by HCI/UX practitioners with experience in performing HE, but this is not always the case in the real world. When a company or firm does not employ HCI/UX practitioners or cannot afford to hire them to perform evaluation, members from an existing development team will usually conduct HE. The combination of novices and experts aimed to make the experiment both realistic and reliable.

To produce meaningful results, a large enough number of participants should be recruited for each condition; 10–12 participants per condition is usually considered sufficient to produce statistical significance (Sova & Nielsen, 2010). As such, we planned to recruit at least 10 participants per condition, with a total of at least 20 novice evaluators. As for experts, since three to five experts can detect about 85% of usability issues, we aimed to recruit at least five experts.

We recruited 20 novices, all of whom were from the field of computer science or related fields. All had less than 4 years of experience and had never performed heuristic evaluation outside of classroom settings.

The novices were from five different universities. Four were from DePaul University, four were from University of Cauca, four were from Autonomous University of Aguascalientes, four were from Autonomous University of Zacatecas, and four were from King Abdulaziz University. Four were undergraduates in telecommunication engineering, four were undergraduates in intelligent computing, one was an undergraduate in user experience design, four had a bachelor's degree in information system, one was a graduate student in information system, two were graduate students in human computer interaction, and four were graduate students in information processing. In terms of gender, 13 of the novices were male, six were female, and one was non-binary. Table 7 shows a summary of novice participant demographics.

Table 7. *Novice participants*

Participant	*Field*	*Level*	*Gender*	*University*
P1	Telecommunication engineering	Bachelor's student	Female	University of Cauca
P2	Telecommunication engineering	Bachelor's student	Male	University of Cauca
P3	Telecommunication engineering	Bachelor's student	Female	University of Cauca

P4	Telecommunication engineering	Bachelor's student	Female	University of Cauca
P5	Information Processing	Master's student	Male	Autonomous University of Zacatecas
P6	User Experience design	Bachelor's student	Non-binary	DePaul University
P7	Information systems	Master's student	Female	DePaul University
P8	Intelligent Computing	Bachelor's student	Male	Autonomous University of Aguascalientes
P9	Information Processing	Master's student	Male	Autonomous University of Zacatecas
P10	Information systems	Has a bachelor's degree	Male	King Abdulaziz University
P11	Information systems	Has a bachelor's degree	Male	King Abdulaziz University
P12	Information Processing	Master's student	Female	Autonomous University of Zacatecas
P13	Intelligent Computing	Bachelor's student	Male	Autonomous University of Aguascalientes
P14	Information systems	Has a bachelor's degree	Male	King Abdulaziz University
P15	Information systems	Has a bachelor's degree	Male	King Abdulaziz University
P16	Intelligent Computing	Bachelor's student	Male	Autonomous University of Aguascalientes
P17	Human computer interaction	Master's student	Male	DePaul University

P18	Human computer interaction	Master's student	Female	DePaul University
P19	Intelligent Computing	Bachelor's student	Male	Autonomous University of Aguascalientes
P20	Information Processing	Master's student	Male	Autonomous University of Zacatecas

These novices were divided into two conditions. Table 8 shows the participants in the first condition and. Table 9 shows the participants in the second condition.

Table 8. *First condition participants*

Participant	*Field*	*Level*	*Gender*	*University*
P1	Telecommunication engineering	Bachelor's student	Female	University of Cauca
P2	Telecommunication engineering	Bachelor's student	Female	University of Cauca
P3	Information Processing	Master's student	Male	Autonomous University of Zacatecas
P4	Information systems	Master's student	Female	DePaul University
P5	Information Processing	Master's student	Male	Autonomous University of Zacatecas
P6	Information systems	Has a bachelor's degree	Male	King Abdulaziz University
P7	Intelligent Computing	bachelor's student	Male	Autonomous University of Aguascalientes

P8	Information systems	Has a bachelor's degree	Male	King Abdulaziz University
P9	Human computer interaction	Master's student	Male	DePaul University
P10	Intelligent Computing	bachelor's student	Male	Autonomous University of Aguascalientes

Table 9. *Second condition participants*

Participant	*Field*	*Level*	*Gender*	*University*
P1	Telecommunication engineering	Bachelor's student	Male	University of Cauca
P2	Telecommunication engineering	Bachelor's student	Female	University of Cauca
P3	User Experience design	Bachelor's student	Non-binary	DePaul University
P4	Intelligent Computing	Bachelor's student	Male	Autonomous University of Aguascalientes
P5	Information systems	Has a bachelor's degree	Male	King Abdulaziz University
P6	Information Processing	Master's student	Female	Autonomous University of Zacatecas
P7	Information systems	Has a bachelor's degree	Male	King Abdulaziz University
P8	Intelligent Computing	Bachelor's student	Male	Autonomous University of Aguascalientes

P9	Human computer interaction	Master’s student	Female	DePaul University
P10	Information Processing	Master’s student	Male	Autonomous University of Zacatecas

We recruited eight expert participants as the benchmark. All of the expert participants had at least 4 years of experience in the field and had performed HE at least three times before.

Two expert participants had 6 years of experience in the field, three had 7 years of experience, one had 9 years of experience, one had 11 years of experience, and one had 12 years of experience. Four worked in academia and four worked in industry and their positions in academia and industry differ: three were UX researchers, one was a UX designer, two were assistant professors, and two were PhD students. In terms of gender, six were male and two were female. They came from different educational backgrounds: five were from computing related background; one was from an engineering related background; one was from a psychology related background; and one was from an art related background. The highest degree earned also differed: five had a master’s degree; two had a PhD; and one had a bachelor’s degree. Table 10 shows a summary of expert participant demographics.

Table 10. *Expert participants*

Particip-ant	*Years of Exper-ience*	*Type of Work*	*Current Position*	*Gender*	*Educational Background*	*Highest Degree Earned*
P1	7	Industry	UX researcher	Male	Engineering	Master's degree
P2	7	Industry	UX researcher	Male	Computing	Master's degree
P3	7	Academia	Assistant professor	Male	Computing	PhD
P4	12	Industry	UX designer	Female	Art	Bachelor's degree
P5	9	Academia	PhD student	Male	Computing	Master's degree
P6	11	Academia	Assistant professor	Female	Computing	PhD
P7	6	Industry	UX researcher	Male	Psychology	Master's degree
P8	6	Academia	PhD student	Male	Computing	Master's degree

5.2.2. System

Evaluators require a suitable system to evaluate. While there are a variety of different types of systems that could have been selected, we chose internet-based systems, as they are one of the most widely used system types (Sova & Nielsen, 2010) and their use has significantly increased over the last 10 years. In 2010, internet-based systems had more than 1.9 billion users, constituting almost 30% of the world's population, which increased to over 4.6 billion users by 2020, close to 60% of the global population (Miniwatts Marketing Group, n.d.).

Some systems, such as health-related sites, serve a very specific audience, while more general systems, such as mobile systems and event management systems, are used by a very wide range of audiences. We chose to use a general rather than a specific audience system for the experiment.

Functionality is another criterion affecting system choice. Some systems are concentrated around one function that users will mostly use to accomplish a single goal; for example, Google Search is mostly used for searching. However, we decided to use a multi-function system, which can be used to accomplish a variety of goals.

Well-known systems that are used by many people tend to conceal issues as users become very proficient in using them and no longer notice issues that at one time may have been troublesome. We sought to find a less widely used system so that participants would not struggle to find issues on this basis.

Finally, systems differ with respect to their level of usability. Since this experiment has been designed to compare two groups of evaluators, we decided to select a system with a high number of usability issues that requires significant improvement.

In summary, the system criteria were as follows:

- Internet-based system.
- Targets a general audience.
- Multi-function system.
- Not widely used.

- Needs significant improvement in terms of usability.

Based on the aforementioned criteria, we chose the system of a large telecommunications operator in the Middle East. Their internet-based system allows customers of different ages and backgrounds to accomplish a variety of goals, including paying bills, adding and removing services, sending points or minutes to others, keeping track of usage, redeeming points, filing complaints, etc. It was also selected because users had been complaining about the usability of the current version, showing that the system needs to be improved in terms of its usability.

5.2.3. Settings

We had three options for the setting for performing the evaluation: first, to bring the participants to the lab, show them the system for evaluation, give them instructions on how to evaluate it, and finally ask them to evaluate the system and produce their results in a limited time; second, to bring them to the lab, give them a brief introduction to the method and the system, and ask them to take it home with them, perform the evaluation, and send us the results; and, third, after recruiting participants and ensuring that the instructions and all required materials are available online, to give participants a link to the instructions and the materials and ask them to perform the evaluation at home and send us the results.

We chose the third option. The first option is somewhat unrealistic, because it would force evaluators to complete the evaluation in a short period of time, usually 2–3 hours, and also require them to perform the evaluation in a lab, which does not mirror the customary real-world environment. The second option is also unrealistic, because in some cases evaluators are asked to perform HE by only searching and reading the material themselves without any actual training. We therefore decided that the third option is the most realistic simulation. However, to control the process, we gave participants a limited amount of time, between 1–2 hours, to complete the main activities of understanding the heuristics and inspecting the system. Although they completed these activities online, we were also present via video call to monitor the process and the process itself was recorded.

We chose Zoom as the tool to conduct the sessions, meeting all the participants, experts, and novices using this platform. The sessions were recorded and saved on the researcher's computer.

5.2.4. Procedures

To recruit the two types of desired users, experts, and novices, we decided to use a snowball sampling technique, recruiting experts who had at least 4 years of experience in the field and had completed HE at least three times previously and novices from information technology, computer science, software engineering, and related fields who were familiar with usability but who had never completed HE before outside of classroom settings.

Experts would be provided with the system and asked to evaluate it using their usual methods without any specific instructions. Novices would be randomly assigned to one of two groups; the first group would evaluate the system based on parts of CoHE, namely detailed usability heuristics and operational usability heuristics, while the second group would evaluate the system based on HE.

The novices evaluating the system based on parts of CoHE would be given detailed usability heuristics and operational usability heuristics; they would have to start with the understanding phase, using the detailed usability heuristics, and then moved on to the inspecting phase, using the operational usability heuristics. After the understanding phase, evaluators would be asked to answer multiple questions to assess their understanding of the heuristics and, after finishing the inspecting phase, they would be asked to respond to a brief survey to assess their level of confidence regarding their results and to reflect on the process and talk about the difficulties they faced during evaluation.

Novices evaluating the system based on HE would not be provided with detailed usability heuristics nor operational usability heuristics; instead, they would only be given links to Nielsen's explanation of usability heuristics and HE. We chose Nielsen's heuristics because it is the most well-known; it appears as the first result when HE is typed into Google Search. Thus, we believed that this would provide a realistic simulation of the information that a novice evaluator would be most likely to receive about HE in the real world. After reading Nielsen's heuristics, the evaluators would be asked to answer multiple questions to assess their understanding of the heuristics and, after finishing the inspection, they would be asked to respond to a brief survey to assess their level of confidence in the results. They would also be asked to reflect on the process and talk about difficulties they may have faced during the evaluation.

For both groups of novices, the process would be divided into two main activities. The first activity would be understanding the usability heuristics. As outlined above, novices using CoHE would be provided with the required material, contacted over a video call, and asked to read the material; they would be given up to 2 hours to complete the reading and the session would be recorded. Novices using traditional HE would be provided with Nielsen's materials on his 10 usability heuristics; they would be given up to 2 hours to complete the reading. Experts would not be included in this point, because the assumption is that they would already be familiar with HE and would know how to apply it.

The second activity would be introducing the system to all three groups. They would be given up to 2 hours to finish the evaluation and all groups would be asked to evaluate the same goals on the system.

Novices using CoHE would be given operational usability heuristics, novices using traditional HE would be provided with Nielsen's explanation of the process, and experts would not be given any information; the latter would be reminded to use Nielsen's heuristics, with which they should already be familiar. The sessions would be recorded for analysis.

We designed the two sessions to be held separately, divided over two different days, to avoid overwhelming participants.

We started by preparing the materials for the experiment, which were also submitted to the IRB for approval. After approval was granted (Appendix E), we started the process of recruiting participants.

To recruit novices, we contacted HCI professors we already knew at DePaul University, University of Cauca, Autonomous University of Zacatecas, Autonomous University of Aguascalientes, and King Abdulaziz University. We explained to them the purpose of the study, informed them on the criteria for the participants, and provided them with the recruiting material (Appendix F), asking them to link us to any students who met the criteria and were willing to participate. We also used the CDM pool to reach participants.

To recruit experts, we contacted HCI/UX practitioners we knew who met the criteria. We explained to them the purpose of the study and asked if they would be willing to participate in the experiment (Appendix G).

After recruitment, we created a digital calendar using Calendly to enable participants to choose a convenient date and time for the sessions. Once selected, the Zoom links were automatically sent to them. 24 hours before the session, a reminder was sent to the participants via email (Appendix H).

Each novice participant signed up for two sessions, an understanding session and an inspecting session. The two sessions could not be selected for the same day, to avoid overwhelming the participants. Each expert signed up for one inspecting session only. Each session lasted for up to two hours.

Novices were randomly assigned to one of the two conditions: the first condition was a traditional heuristic evaluation, and the second condition was parts of the CoHE evaluation.

In the first session, we started by reading the information sheet to each participant (Appendix I) and they were asked some background questions. After that, they were given a link containing the usability heuristics: the first condition received a link containing Nielsen's 10 usability heuristics (Appendix J); the second condition received a link containing the detailed usability heuristics from CoHE and we added examples to each heuristic (Appendix K). Participants were encouraged to take their time reading the heuristics and to keep reading the heuristics carefully until they felt they fully understood them. Subsequently, participants were asked some questions regarding the heuristics (Appendix L). They were asked to explain the heuristics, give examples, and answer questions about the difficulties they faced during the reading (with the labels, descriptions, and examples).

In the second session, novices were asked to briefly review the list of heuristics. The first condition were again sent Nielsen's 10 usability heuristics to use in the inspection. The second condition were sent the operational usability heuristics from CoHE (Appendix M) to use in the inspection. After that, they were sent a link for the website for evaluation. First, they were encouraged to briefly explore the homepage of the website to gain an overall understanding of it. Then, they were assigned a series of 11 tasks, one by one, to perform on the website:

1. Transferring credits to another person.
2. Deactivating an active service.
3. Redeeming points to get free minutes of calls.

4. Paying their bill.
5. Changing the number, and then recharging the second number.
6. Viewing their service orders for the last year.
7. Activating the roaming service.
8. Adding international minutes.
9. Filing a complaint about a technical issue on the website.
10. Changing the current package to a different package.
11. Getting information on how to activate 5G.

The novices were told to perform the tasks and make note of any usability issue they found in the process. After each task, they were asked to review the heuristics to reflect on the task and list the issues they faced, linking them to the usability heuristics.

After finishing the 11 tasks, the novice participants were asked two sets of questions (Appendix N). The first set of questions was about their level of confidence in the evaluation they had performed. These questions asked how confident participants felt with regards to finding all the critical issues, linking the issues to the correct heuristics, giving recommendations, etc. The second set of questions was about their whole experience of the method. These questions sought to find out which parts they liked and disliked, which aspects were easiest and which were most difficult, etc.

Each expert participated in only one session. In this session, we started by reading the information sheet to each participant (Appendix O) and were then provided with a link to the website. Experts were given some time to explore the homepage and they were then assigned the same 11 tasks as the novices. The process was also similar; we asked them to identify any issue they encountered in the process and, after each task, they were given some time to reflect on the task and list the issues they found. The main difference is that experts were not given a list of heuristics and they were not asked any post-evaluation questions (Appendix P).

Overall, we organized 48 sessions: 40 sessions with novices, 20 of which were understanding sessions and 20 of which were inspecting sessions; and eight sessions with experts, all of which were inspecting sessions.

The sessions took place between April and August 2021: nine sessions took place in April, 16 sessions took place in May, 20 sessions took place in June, two sessions took place in July, and one session took place in August.

5.2.5. Analysis

After completing the evaluation, we planned to examine all the recordings and list all the problems for each participant. Then, we would begin analyzing the reports to answer the research questions, using the reports of the expert evaluators as the benchmark. Hartson et al. (2001) offered three ways of assessing the effectiveness of a usability evaluation method (pp. 373–410), all depending on making a list of real usability issues to compare against the usability issues that emerge from the usability evaluation method in question. These three ways are as follows:

1. Comparing against a list of standard usability problems; once all usability problems of a certain system are known and documented, they are consolidated into a single list of all the usability problems of that system.
2. Comparing against expert judgment; usability problems found by the usability evaluation method in question are examined by usability experts in order to judge whether they are realistic.
3. Comparing against end-user judgment; usability problems found by the usability evaluation method in question are examined by end-users or compared to problems produced by end-users to determine whether they are realistic.

Given that the aim of the experiment is to compare the quality of evaluation of CoHE versus traditional HE, the experts provide the reference point. As part of this comparison, we decided to use a list of problems produced by the experts as a benchmark. This would mean that any problem found by novices could be compared with the list of problems produced by the experts; if the problem found is on the experts' list, it would be considered a real problem, otherwise it would be considered a false positive (Hartson et al., 2003).

Babajo and Petrie (2012) suggest that there are two criteria with which usability problems can be compared: relaxed criteria and strict criteria. Relaxed criteria specify that two problems can be considered the same if they describe the same underlying usability problem, regardless of their level of abstraction. Strict criteria specify that two problems can be considered the same if they both describe the same design element and have the same level of abstraction. We designed this experiment to compare the methods against each of these criteria separately.

5.2.5.1 Usability problems measures

Usability problems can be described using one of three descriptions:

Real problems (Hits). Problems found by the usability method in question and also found by the benchmark.

Unfound problems (Misses). Problems found by the benchmark but not found by the usability method in question.

False positives (False alarms). Problems found by the usability method in question but not found by the benchmark.

According to Hartson et al. (2001), different usability methods use several different measures for comparison: thoroughness, validity, and effectiveness. Thoroughness and validity are analogues to recall and precision in the field of information retrieval. (pp. 373–410):

Thoroughness. Number of real problems found by the method in question/number of those found by the benchmark.

Validity. Number of real problems found by the method in question/all problems found by the method in question.

Effectiveness. Thoroughness * validity.

Since sometimes there is a valid reason for preferring one measure to the other, Hartson et al. (2001) used a weighted measure called an f-measure which was first introduced in the field of information retrieval (pp. 373–410):

F-measure. 1/α (1/validity) + (1-α) (1/thoroughness).

If thoroughness and validity are given the same weight, α would equal 0.5 and the f-measure would be: 2 *(validity * thoroughness)/(validity + thoroughness).

5.2.5.2 Severity of usability problems

The experiment was also designed to answer the question of whether there are differences between the three conditions in terms of the severity of usability problems found. To this end, we decided that two independent expert evaluators would be recruited to rate the severity of the problems found. According to Nielsen (1992), to obtain more reliable results, the mean of the judgment of several evaluators should be considered. In line with this, the evaluators would be handed an aggregated list of all the problems to rate the severity based on Nielsen's (1995) severity scale as follows:

0 = "I don't agree that this is a usability problem at all."

1 = "Cosmetic problem only: need not be fixed unless extra time is available on the project."

2 = "Minor usability problem: fixing this should be given low priority."

3 = “Major usability problem: important to fix, so should be given high priority.”

4 = “Usability catastrophe: imperative to fix this before the product can be released.”

5.2.5.3 Conducting analysis

All of the sessions were held and recorded via Zoom. After each session, the video recordings were downloaded to the researcher’s computer.

The analysis involved two rounds. The first round was performed immediately after the session; the researcher watched the entire session and took note of any significant occurrences. The second round was completed after finishing all the sessions; the researcher watched each, one by one, for more careful analysis.

Analysis of the understanding sessions focused on three main aspects. First, we measured the time it took each participant to finish reading the materials and announce that they were ready to answer questions about them. Second, we focused on the way participants described the heuristics. We asked: Were they able to fully explain them? Were they able to come up with examples of each heuristic? Were they able to differentiate between different heuristics? We watched the sessions carefully to note any issue of these kinds and we then aggregated all of the problems noted for each condition into one list of repeated problems. Third, we considered participants’ comments. During and after the reading, participants expressed opinions regarding the heuristics either through unsolicited comments or by answering the questions at the end of the session. We aggregated these comments for each condition into one list of repeated comments.

To inspect the sessions, we started with novices. We watched the videos twice, marking all the issues that the participants mentioned during the sessions. We created a spreadsheet for each participant, pasting screenshots of the problems, the design elements that have the problem, the page where the problem occurred, a description of the problem, and the recommendations given to solve the problem.

After this, we undertook the same process with expert sessions. We created a spreadsheet for each participant, containing the same information as the novices. Then, we created a master list merging all the problems found by experts, removing duplicates. The master list was used as the benchmark to decide whether a problem found by novices was a real problem or a false positive problem.

After creating the master list, we set out to rate the severity of the issues found by experts. We asked two usability experts to review the list of the problems, discuss the issues, and rate the severity of each usability issue found in the list. This process was completed online via Zoom; the two experts joined a session monitored by the researcher. They used Nielsen's 0–4 severity rating scale, as outlined above. However, given that each problem which an expert found was already assumed to be a real problem, the scale used excluded the rating 0, as this describes an issue considered as a false issue.

The final step was to examine the problems found by novices. Both the relaxed and strict criteria were used to establish the realness of any given problem. In the relaxed criteria, we considered the problem to be real if it matched an underlying problem found in the master list, or the same problem with a different level of abstraction. In the strict criteria, we considered the problem to be a real problem only if it matched the same problem found in the master list with the same level of abstraction.

After measuring the realness of these problems, we started to identify the severity of each problem. At this stage, we only looked at the real problems found by the strict criteria, since they describe exactly the same problems as those found in the master list. For each real problem found by novices, we looked at the problem that it matched in the master list and assigned it the same severity rating.

We also considered the time it took each participant to finish the inspection. We started measuring the time once the participant opened the website, until they logged out and announced that they did not have anything more to add.

After the inspection, we asked the novice participants two sets of questions. The first set was about their level of confidence in the evaluation they had performed; the questions were Likert scale questions. We created a table for each condition and noted the answers of each participant for each question. The second set of questions were open-ended questions about participants' overall experience with the method. We noted participants' comments about their overall experience and we consolidated the repeated comments into a single list.

CHAPTER 6. RESULTS

This chapter outlines the results of the experiment. Mirroring the experiment itself, it is divided into three sections: the first section discusses the understanding part, examining how well the participants understood the heuristics; the second section addresses the inspecting part, examining how effectively the participants inspected the system; and the third section explores participants' overall experience with the method, examining opinions about the entire experience (Figure 23).

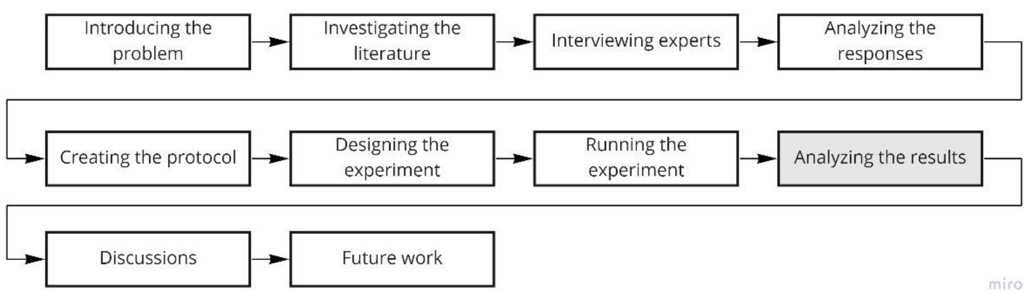

Figure 23. *Thesis organization (Chapter 6)*

6.1. Understanding the heuristics

Understanding is essential for utility. Without good understanding, any application might not be satisfactory. This section presents the results of the understanding sessions to determine how well the participants understood the heuristics.

6.1.1. Time

The time taken to read and understand the usability heuristics is critical, because one of the main advantages of HE is its low time requirements. During the understanding session, participants were asked to take sufficient time to read the usability heuristics until they felt they had fully understood them.

The participants of the first condition were given Nielsen's 10 usability heuristics, while the participants of the second condition were given detailed usability heuristics. To see the time it took each participant in each condition to read and understand the heuristics, see appendix Q.

In the first condition, the shortest time for a participant to read the heuristics was 6:17 minutes and the longest time was 13:06 minutes. On average, it took a participant in this condition 8:45.6 minutes to read and understand the heuristics. The accumulated time for all participants in the first condition was 1:27:36 hours. By contrast, the shortest time for a participant in the second condition to read the heuristics was 18:22 minutes and the longest time was 37:52 minutes. On average, it took a participant in this condition 28:26.1 minutes to read and understand the heuristics. The accumulated time for all participants in the second condition was 4:44:21 hours.

6.1.2. Missing concepts

As discussed previously, some of the usability heuristics have more than one concept within the same heuristic. These heuristics are:

- Consistency and standards.

- Flexibility and efficiency of use.
- Aesthetic and minimalist design.
- Help users recognize, diagnose and recover from errors.
- Help and documentation.

In some cases, participants explaining the heuristics explained one of the concepts, but they did not explain the other. This was an issue mainly with the participants in the first condition.

In the first condition, all of the participants missed at least one of the concepts in these heuristics, although they missed different concepts: five participants missed concepts in consistency and standards; nine participants missed concepts in flexibility and efficiency of use; eight participants missed concepts in aesthetic and minimalist design; eight participant missed concepts in help users recognize, diagnose and recover from errors; and nine participants missed concepts in help and documentation. In this second condition, this issue only occurred once: one participant missed a concept in help users recognize, diagnose and recover from errors. To see the missing concepts for participants in the first and second conditions, see appendix Q.

6.1.3. Confusing concepts

While some of the usability heuristics address apparently similar issues, this does not mean that they are same. However, some participants in both conditions confused some of the heuristics with each other.

In the first condition, two participants confused consistency and standards with recognition rather than recall; one participant confused visibility of system status with flexibility and efficiency of use; one participant confused error prevention with help users recognize, diagnose and recover from errors; one participant confused visibility of system status with match between system and the real world; one participant confused error prevention with recognition rather than recall; one participant confused user control and freedom with visibility of system status; and one participant confused consistency and standards with match between system and the real world. In the second condition, two participants confused aesthetic and minimalist design with flexibility and efficiency of use; one participant confused consistency and standards with match between system and the real world; one participant confused aesthetic and minimalist design with recognition rather than recall; and one participant confused user control and freedom with error prevention. To see the concepts that were confused by participants in the first and second conditions, see appendix Q.

6.1.4. Difficulty understanding concepts

Some of the heuristics can be difficult for novices to completely or partially understand. In both conditions, some participants experienced difficulties in understanding the concepts.

In the first condition, three participants did not fully understand flexibility and efficiency of use; three participants did not fully understand recognition rather than recall; two participants did not fully understand help and documentation; two participants did not fully understand match between system and the real world; one participant did not fully understand consistency and standards; and one participant did not fully understand aesthetic and minimalist design. In the second condition, two participants did not fully understand visibility of system status; two participants did not fully understand consistency and standards; one participant did not fully understand match between system and the real world; and one participant did not fully understand recognition rather than recall. To see the concepts which participants had difficulty understanding in the first and second conditions, see appendix Q.

6.1.5. Comments about the heuristics

After completing the reading, participants were asked multiple questions about the usability heuristics they had read. The main goal of these questions was to encourage participants to reflect on the process of reading the heuristics and to discuss what they liked and disliked about it and the challenges they faced.

In the first condition, repeated comments concerned the lack of examples, the need to clarify certain words or ideas in the descriptions, the similarities between certain heuristics, the need for more detailed descriptions, and the vagueness of certain labels.

With regards to examples, seven participants commented on the lack of examples in the usability heuristics, stating that, without having clear examples, some heuristics appeared very abstract and hard to grasp; most recommended that multiple examples needed to be given to clarify the concepts that the heuristics are trying to convey.

On clarifying descriptions, five participants highlighted that the ideas had multiple potential meanings and that the actual meaning in the description was unclear; they suggested that there should be extra explanations for these ideas or a separate list to define the ideas clearly.

Discussing the similarities between certain heuristics, seven participants felt that some heuristics were expressing the same underlying idea and should be merged into a single heuristic; they suggested that, if these heuristics are in fact different, the differences should be emphasized so that the reader can understand these differences easily.

In terms of providing more detailed descriptions, six participants stated that some of the descriptions were not as detailed as needed. They felt that the descriptions were very brief and insufficient for understanding such general concepts; they suggested that descriptions should be more extensive to give the reader a better understanding of the concepts.

On the labels, six participants mentioned that some of the labels did not appear to match the ideas that the heuristics were trying to convey or were too vague to help swift understanding; they recommended exchanging these labels for more meaningful ones.

Similarly, in the second condition, repeated comments discussed the lack of examples, the categorization of concepts, the vagueness of certain labels, and the organization of heuristics.

In relation to examples, three participants stated that some of the examples were unclear; they suggested that better or simpler examples could help to explain the concepts more effectively.

On the categorization of the concepts, four participants stated that some of the concepts seemed to be miscategorized; they agreed on the importance of the concepts, but they felt that they belonged to different heuristics than those under which they are currently placed.

With regards to labels, five participants mentioned that some of the labels were not clear enough to understand the concepts immediately; they recommended changing the labels to make them clearer and more understandable.

For the organization of heuristics, three participants suggested that, while they agreed with the content, they felt that the order of the points needed changing, proposing that certain elements should be moved from one place to another to facilitate the reading process.

To see a summary of the comments of participants in the first and second conditions, see appendix Q.

6.2. Inspecting the system

The main activity in HE is to evaluate a system. Good evaluation is the key to enhancing the quality of the system. This section thus discusses the results of the inspecting session, showing how long the participants took in evaluating the system and how many problems they found.

6.2.1. Time

Given that HE is considered a discount usability method, it is important to measure how long it takes an evaluator to evaluate a system.

As outlined previously, the participants of the first condition were asked to evaluate the system based on Nielsen's 10 usability heuristics, while the participants of the second condition were asked to evaluate the system based on operational usability heuristics from CoHE. To see the time it took each participant from each condition to perform the evaluation, see appendix Q.

The maximum time it took a participant in the first condition to perform the evaluation was 1:40:36 hours and the minimum time was 1:10:12 hours. The average time for the participants in the first condition to perform the evaluation was 1:24:06.7 hours. The accumulated time for the first condition was 14:01:07 hours. Meanwhile, the maximum time it took a participant in the second condition to perform the evaluation was 1:58:33 hours and the minimum time was 1:20:51 hours. The average time for the participants in the second condition to perform the evaluation was 1:42:01.1 hours. The accumulated time for the second condition was 17:00:11 hours.

6.2.2. Number of problems found

The main goal of HE is to find as many usability problems as possible so that these problems can be fixed before real users encounter them on the system. As such, it is important to discuss the number of problems found by each participant in each condition. To see the number of problems found by the participants in the first and second condition, see appendix Q.

The maximum number of problems found by a participant in the first condition was 29 problems and the minimum number was 17 problems; the mean of the problems found by the first condition was 22 problems and the median was 20.5 problems. By contrast, the maximum number of problems found by a participant in the second condition was 37 problems and the minimum number was 18 problems; the mean of the problems found by the second condition was 27.8 problems and the median was 27.5 problems.

6.2.3. Number of problems found (Relaxed criteria)

However, not all problems found by novices in HE are real problems. Thus, the problems found by novices needed to be filtered to eliminate the false positive problems. To see the number of real problems found by the participants in the first and second conditions, after they had been filtered through the relaxed criteria, see appendix Q; in the relaxed criteria, two problems are considered the same if they describe the same underlying problem or the same problem with different levels of abstraction.

The maximum number of real problems found by a participant in the first condition was 28 real problems and the minimum number was 16 real problems; the mean of the problems found by the first condition was 19.6 real problems and the median was 18.5 real problems. The maximum number of problems found by a participant in the second condition was 32 real problems and the minimum number was 17 real problems. The mean of the problems found by the second condition was 24.6 real problems and the median was 25 real problems.

6.2.4. Number of problems found (Strict criteria)

Unlike the relaxed criteria, the strict criteria consider two problems to be the same only if they describe the same problem with the same level of abstraction. To see the number of real problems found by the participants in the first and second conditions, after the problems were filtered using the strict criteria, see appendix Q.

The maximum number of real problems found by a participant in the first condition was 23 real problems and the minimum number was 13 real problems; the mean of the problems found by the first condition was 16.5 real problems and the median was 15.5 real problems. However, the maximum number of problems found by a participant in the second condition was 27 real problems and the minimum number was 17 real problems; the mean of the problems found by the second condition was 20.9 real problems and the median was 20.5 real problems.

6.2.5. Severity of problems found

Not all problems are considered equally important; fixing some problems is more urgent than others. It is important to know the severity of each problem found to know where to place it on a list of priorities. See appendix Q. It breaks down all the problems found by each participant in the first and second condition into cosmetic, minor, major, and catastrophe problems; this only includes real problems, as filtered by the strict criteria.

In the first condition, the maximum number of cosmetic problems found was 1, the minimum was 0, the mean was 0.2, and the median was 0; the maximum number of minor problems found was 6, the minimum was 1, the mean was 3.4 and the median was 3; the maximum number of major problems found was 10, the minimum was 3, the mean was 6.1 and the median was 5.5; and the maximum number of catastrophe problems was 11, the minimum was 4, the mean was 6.8 and the median was 6.5. By contrast, in the second condition, the maximum number of cosmetic problems found was 3, the minimum was 0, the mean was 0.7, and the median was 0.5; the maximum number of minor problems found was 6, the minimum was 1, the mean was 3.5 and the median was 3.5; the maximum number of major problems found was 10, the minimum was 5, the mean was 6.6 and the median was 6; and the maximum number of catastrophe problems was 13, the minimum was 8, the mean was 10.1 and the median was 10.

6.2.6. Weighted problems

In performing an inspection, the main concern is not only how many problems are found but also how many problems are missed and how many problems are false positives. Therefore, using the equations outlined previously, we can show the thoroughness, validity, effectiveness, and f-measure for each participant in each condition: thoroughness measures the percentage of problems found out of all of the existing problems; validity measures the percentage of all real problems found by an evaluator out of all of the problems found by the evaluator; effectiveness is the product of thoroughness and validity; and the f-measure gives equal weight to validity and thoroughness.

Table 11 shows a breakdown of the aggregated problems found by experts, which is used here as a golden list.

Table 11. *List of problems found by experts*

Cosmetic	*Minor*	*Major*	*Catastrophe*	*Total*
27	55	72	42	196

Table 12 – Table 13 show the thoroughness, validity, effectiveness and f-measure of each participant in the first and second conditions, using the real problems filtered by the relaxed criteria, while Table 14 – Table 15 show the thoroughness, validity, effectiveness and f-measure of each participant in the first and second conditions, using the real problems filtered by the strict criteria.

Table 12. *Weighted problems for condition one (Relaxed)*

Participant	*Thoroughness*	*Validity*	*Effectiveness*	*F-measure*
P1	0.107	0.955	0.102	0.192
P2	0.087	1	0.087	0.160
P3	0.097	0.905	0.088	0.175
P4	0.102	0.769	0.078	0.180
P5	0.092	0.9	0.083	0.167
P6	0.112	0.786	0.088	0.196
P7	0.082	0.8	0.066	0.149
P8	0.092	1	0.092	0.168
P9	0.087	0.895	0.078	0.159
P10	0.143	0.966	0.138	0.249

Table 13. *Weighted problems for condition two (Relaxed)*

Participant	*Thoroughness*	*Validity*	*Effectiveness*	*F-measure*
P1	0.128	0.962	0.123	0.226
P2	0.133	0.897	0.119	0.232
P3	0.133	0.765	0.102	0.227
P4	0.117	0.958	0.112	0.209
P5	0.128	0.833	0.107	0.222
P6	0.087	0.944	0.082	0.159
P7	0.163	0.865	0.141	0.274
P8	0.117	0.885	0.104	0.207
P9	0.122	0.96	0.117	0.216
P10	0.128	0.862	0.110	0.223

Table 14. *Weighted problems for condition one (Strict)*

Participant	*Thoroughness*	*Validity*	*Effectiveness*	*F-measure*
P1	0.092	0.818	0.075	0.165
P2	0.066	0.765	0.050	0.122
P3	0.082	0.762	0.062	0.148
P4	0.071	0.538	0.038	0.125
P5	0.077	0.75	0.058	0.139
P6	0.102	0.714	0.073	0.179
P7	0.071	0.7	0.049	0.129
P8	0.087	0.944	0.082	0.159

P9	0.077	0.789	0.061	0.140
P10	0.117	0.793	0.093	0.204

Table 15. *Weighted problems for condition two (Strict)*

Participant	*Thoroughness*	*Validity*	*Effectiveness*	*F-measure*
P1	0.092	0.692	0.064	0.162
P2	0.097	0.655	0.064	0.169
P3	0.122	0.706	0.086	0.208
P4	0.102	0.833	0.085	0.182
P5	0.117	0.767	0.089	0.203
P6	0.087	0.944	0.082	0.159
P7	0.138	0.729	0.101	0.232
P8	0.097	0.731	0.071	0.171
P9	0.107	0.84	0.089	0.189
P10	0.107	0.724	0.077	0.186

Using the relaxed criteria for the first condition, the maximum thoroughness was 0.143, the minimum was 0.082, the mean was 0.1001, the median was 0.0945; the maximum validity was 1, the minimum was 0.769, the mean was 0.8976, the median was 0.9025; the maximum effectiveness was 0.138, the minimum was 0.066, the mean was 0.09, the median was 0.0875; and the maximum f-measure was 0.249, the minimum was 0.149, the mean was 0.1795, and the median was 0.1715. Using the relaxed criteria for the second condition, the maximum thoroughness was 0.163, the minimum was 0.087, the mean was

0.1256, the median was 0.128; the maximum validity was 0.962, the minimum was 0.765, the mean was 0.8931, the median was 0.891; the maximum effectiveness was 0.141, the minimum was 0.082, the mean was 0.1117, the median was 0.111; the maximum f-measure was 0.274, the minimum was 0.159, the mean was 0.2195, and the median was 0.2225.

Using the strict criteria for the first condition, the maximum thoroughness was 0.117, the minimum was 0.066, the mean was 0.0842, the median was 0.0795; the maximum validity was 0.944, the minimum was 0.538, the mean was 0.7573, the median was 0.7635; the maximum effectiveness was 0.093, the minimum was 0.038, the mean was 0.0641, the median was 0.0615; and the maximum f-measure was 0.204, the minimum was 0.122, the mean was 0.151, and the median was 0.144. Using the strict criteria for the second condition, the maximum thoroughness was 0.138, the minimum was 0.087, the mean was 0.1066, the median was 0.1045; the maximum validity was 0.944, the minimum was 0.655, the mean was 0.7621, the median was 0.73; the maximum effectiveness was 0.101, the minimum was 0.064, the mean was 0.0808, the median was 0.0835; and the maximum f-measure was 0.232, the minimum was 0.159, the mean was 0.1861, and the median was 0.184.

Moreover, we wanted to measure the percentage of the problems found in each severity level out of the whole number of problems in that type. Therefore, we divided the number of problems found in each level for each participant by the number of problems found in each level in the golden list.

Table 16 – Table 17 show the percentage of problems in each level by each participant in the first and second conditions; this includes the problems filtered by the strict criteria only.

Table 16. *Percentage of problems by severity for condition one*

Participant	*% Cosmetics*	*% Minors*	*% Majors*	*% Catastrophes*
P1	0	0.091	0.111	0.119
P2	0	0.018	0.083	0.143
P3	0	0.055	0.083	0.166
P4	0	0.036	0.069	0.166
P5	0.037	0.109	0.055	0.095
P6	0.037	0.055	0.138	0.143
P7	0	0.073	0.069	0.119
P8	0	0.073	0.069	0.190
P9	0	0.055	0.042	0.214
P10	0	0.055	0.125	0.262

Table 17. *Percentage of problems by severity for condition two*

Participant	*% Cosmetics*	*% Minors*	*% Majors*	*% Catastrophes*
P1	0	0.055	0.097	0.190
P2	0	0.055	0.083	0.238
P3	0.037	0.109	0.083	0.262
P4	0	0.073	0.069	0.262
P5	0.037	0.036	0.097	0.309
P6	0	0.018	0.083	0.238

P7	0.111	0.073	0.139	0.238
P8	0.037	0.091	0.069	0.190
P9	0	0.073	0.083	0.262
P10	0.037	0.055	0.111	0.214

In the first condition, the maximum percentage of cosmetic problems found was 0.037, the minimum was 0, the mean was 0.0074, the median was 0; the maximum percentage of minor problems found was 0.109, the minimum was 0.018, the mean was 0.062, the median was 0.055; the maximum percentage of major problems found was 0.138, the minimum was 0.042, the mean was 0.0844, the median was 0.076; and the maximum percentage of catastrophe problems found was 0.262, the minimum was 0.095, the mean was 0.1617, and the median was 0.1545. In the second condition, the maximum percentage of cosmetic problems found was 0.111, the minimum was 0, the mean was 0.0259, the median was 0.0185; the maximum percentage of minor problems found was 0.109, the minimum was 0.018, the mean was 0.0638, the median was 0.064; the maximum percentage of major problems found was 0.139, the minimum was 0.069, the mean was 0.0914, the median was 0.083; and the maximum percentage of catastrophe problems found was 0.309, the minimum was 0.190, the mean was 0.2403, and the median was 0.238.

6.3. Overall experience

After finishing the sessions, we asked participants in both conditions two sets of questions to find out how they felt about the whole experience: the first set was about their level of confidence in the evaluation they had performed; the second set was about their opinion on the method they had used.

6.3.1. Confidence level

To measure their level of confidence in the evaluation they had performed, we asked participants to rate their level of agreement (Strongly disagree, Disagree, Neutral, Agree, Strongly Agree) with each of the following statements, which were inspired by Friess (2015):

- Q1: I am confident that I found all the critical problems on the website.
- Q2: I am confident that I provided useful recommendations that addressed the problems of the website.
- Q3: I am confident the overall problems that I found will be one of the most useful lists of problems for the company running the website.
- Q4: I am confident that real people who will use the website will encounter the problems that I identified.
- Q5: I am confident that I found all the related problems to each heuristic in the list.
- Q6: I am confident that I linked all the problems to the correct heuristic.
- Q7: I am confident that the problems I found covered all the problems that users will encounter.

Table 18 – Table 19 present the responses of the participants in the first and second conditions.

Table 18. *Confidence level for condition one*

Partici-pants	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*	*Q6*	*Q7*
P1	Agree	Agree	Neutral	Agree	Agree	Strongly Agree	Neutral
P2	Disagree	Agree	Neutral	Agree	Disagree	Disagree	Disagree
P3	Agree	Strongly Agree	Agree	Strongly Agree	Agree	Agree	Agree
P4	Strongly Disagree	Strongly Agree	Agree	Strongly Agree	Neutral	Neutral	Strongly Disagree
P5	Disagree	Strongly Agree	Neutral	Strongly Agree	Neutral	Agree	Disagree
P6	Strongly Disagree	Strongly Agree	Neutral	Agree	Neutral	Agree	Strongly Disagree
P7	Agree	Strongly Agree	Strongly Agree	Strongly Agree	Neutral	Agree	Neutral
P8	Disagree	Agree	Neutral	Agree	Neutral	Agree	Disagree
P9	Agree	Strongly Agree	Agree	Neutral	Neutral	Neutral	Disagree
P10	Agree	Strongly Agree	Neutral	Agree	Disagree	Neutral	Neutral

Table 19. *Confidence level for condition two*

Partici-pants	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*	*Q6*	*Q7*
P1	Agree	Agree	Agree	Strongly Agree	Agree	Strongly Agree	Neutral
P2	Disagree	Agree	Neutral	Strongly Agree	Neutral	Neutral	Strongly Disagree
P3	Strongly Agree	Agree	Agree	Strongly Agree	Agree	Agree	Agree

P4	Agree	Strongly Agree	Agree	Strongly Agree	Neutral	Strongly Agree	Strongly Disagree
P5	Strongly Agree	Strongly Agree	Strongly Agree	Strongly Agree	Agree	Strongly Agree	Strongly Disagree
P6	Neutral	Agree	Agree	Agree	Neutral	Agree	Neutral
P7	Strongly Agree	Strongly Agree	Agree	Strongly Agree	Agree	Agree	Agree
P8	Agree	Strongly Agree	Agree	Neutral	Disagree	Agree	Disagree
P9	Agree	Agree	Neutral	Agree	Agree	Agree	Agree
P10	Strongly Agree	Agree	Neutral	Agree	Neutral	Neutral	Disagree

For the first condition, in the first question, two participants responded with “strongly disagree”, three with “disagree”, and five with “agree” and the mode was “agree”; in the second question, three participants responded with “agree” and seven with “strongly agree” and the mode was “strongly agree”; in the third question, six participants responded with “neutral”, three with “agree” and one with “strongly agree” and the mode was “neutral”; in the fourth question, one participant responded with “neutral”, five with “agree” and four with “strongly agree” and the mode was “agree”; in the fifth question, two participants responded with “disagree”, six with “neutral” and two with “agree” and the mode was “neutral”; in the sixth question, one participant responded with “disagree”, three with “neutral”, five with “agree” and one with “strongly agree” and the mode was “agree”; and, in the last question, two participants responded with “strongly disagree”, four with “disagree”, three with “neutral” and one with “agree” and the mode was “disagree”.

For the second condition, in the first question, one participant responded with “disagree”, one with “neutral”, four with “agree” and four with “strongly agree” and the mode was “agree” and “strongly agree”; in the second question, six participants responded with “agree” and four with “strongly agree” and the mode was “agree”; in the third question, three participants responded with “neutral”, six with “agree” and one with “strongly agree” and the mode was “agree”; in the fourth question, one participant responded with “neutral”, three with “agree”, six with “strongly agree” and the mode was “strongly agree”; in the fifth question, one participant responded with “disagree”, four with “neutral” and five with “agree” and the mode was “agree”; in the sixth question, two participants responded with “neutral”, five with “agree” and three with “strongly agree” and the mode was “agree”; and, in the last question, three participants responded with “strongly disagree”, two with “disagree”, two with “neutral” and three with “agree” and the mode was “strongly disagree” and “agree”.

6.3.2. Participants’ experience

After finishing both the reading and the inspecting sessions, we asked the participants in both conditions some questions about their experience with the method. The main goal of these questions was to find out what they liked and disliked about the method and which aspects they found hardest and easiest to perform.

In the first condition, repeated comments were about the role of heuristics in finding usability problems, the generality of heuristics, and the reasons behind using heuristics. By contrast, in the second condition, the only repeated comment was about length of the heuristics.

With regards to the role of heuristics in finding usability problems, four participants in the first condition felt that the usability heuristics did not directly help in finding usability problems on the system; instead, they viewed it more as a retrospective process, where they had already found problems and then tried to link them to the heuristics, rather than using the heuristics as a guide to find usability problems.

On the generality of heuristics, five participants in the first condition stated that the usability heuristics were too general and were not put into context, making them difficult to understand; they recommended adding examples, pictures, videos and real world scenarios to show these heuristics in play.

Discussing the reasons behind using heuristics, four participants in the first condition reflected that there was no explanation of why these heuristics should be used; they said that knowing the reason behind the heuristics would reinforce their understanding and motivate them to use the heuristics more carefully.

With regards to linking the problems to the correct heuristics, six participants in the first condition suggested that the most difficult part of the process was to link the problems they found to the correct heuristics.

On the length of the heuristics, five participants in the second condition believed that the heuristics were too long and require too much time to read; they felt that asking them to read the materials in one sitting was slightly too demanding. Table 20 – Table 21 summarize the comments of the participants in the first and second conditions.

Table 20. *Comments about the process for condition one*

Comment	*No. of Participants*	*Participants*
Retrospective process	4	P1, P4, P8, P9
Lack of context	5	P1, P3, P6, P8, P9
Lack of reason	4	P4, P6, P8, P9
Linking process	6	P2, P5, P6, P7, P8, P10

Table 21. *Comments about the process for condition two*

Comment	*No. of Participants*	*Participants*
Length of heuristics	5	P1, P2, P3, P5, P7

However, there were other repeated comments from participants in the first and second conditions about aspects of the process that they liked.

In the first condition, three participants said that, although they did not provide direct guidance on how to detect usability problems, the heuristics did help them to think harder and to be more critical when looking at the system. Three participants also mentioned that they liked that the heuristics gave them certain tasks to follow while evaluating the system, stating that it helped them to gain focus rather than simply exploring the system without any specific aim.

In the second condition, five participants suggested that the way that the operational usability heuristics is designed as questions to be asked during the evaluation helped directly in detecting usability problems on the system. Three participants stated that they liked the way the heuristics were organized in terms of definitions, significance, exceptions, and examples; they suggested that this helped them to fully understand the concepts. Finally, two participants mentioned that having tasks helped in directing the evaluation and made it easier to detect usability issues on the system.

Table 22 – Table 23 summarize the aspects of the heuristics that participants in the first and second conditions liked.

Table 22. *Likes about the process for condition one*

Comment	*No. of Participants*	*Participants*
Being critical	3	P1, P2, P6
Specific tasks	3	P3, P7, P8

Table 23. *Likes about the process for condition two*

Comment	*No. of Participants*	*Participants*
Guidance	5	P1, P4, P5, P6, P7
Organization	3	P2, P3, P9
Specific tasks	2	P7, P10

Overall, all participants in both conditions described their experience as a positive learning experience. When participants in the first condition were asked if they think that would use the method in the future, seven participants said that they think they are going to use it in the future, one participant said that they might use it alongside other heuristics, and two participants said that they do not think they will use it in the future. By contrast, when participants in the second condition were asked the same question, all the ten participants responded that they think they will use the method in the future.

Table 24 – Table 25 summarize the responses of participants in each condition on the likelihood of using the method in the future.

Table 24. *Likelihood of using the process for condition one*

Use it or not?	*No. of Participants*	*Participants*
Yes	7	P1, P3, P5, P6, P7, P8, P10
Yes, but not alone	1	P2
No	2	P4, P9

Table 25. *Likelihood of using the process for condition two*

Use or Not?	*No. of Participants*	*Participants*
Yes	10	P1, P2, P3, P4, P5, P6, P7, P8, P9, P10

6.4. Answering the research questions

Before conducting the experiment, we had multiple research questions. Here, we present the questions as well as the answers to these questions after we conducted the experiment.

RQ1. Do parts of CoHE, the detailed usability heuristics and operational usability heuristics, and traditional HE differ in terms of time, of the number of usability problems found, the severity of the usability problems, and validity, thoroughness, effectiveness and f-measure?

First, the time between the two groups differed significantly. The group that used traditional HE spent significantly less time than the group that used parts of CoHE to read the heuristics and to do the evaluation.

Second, the number of problems found by the two groups also differed significantly, the group that used parts of CoHE found significantly more problems than the group that used traditional HE. They found more unfiltered problems, more problems under the relaxed criteria and more problems under the strict criteria.

Third, the severity of the issues found by the two groups didn't significantly differ for cosmetic, minor and major problems. However, it did significantly differ for the catastrophe problems. The group that used parts of CoHE found significantly more catastrophe problems than the group the used traditional HE.

Finally, for thoroughness, effectiveness and f-measure, there was a significant difference between the two groups. The group that used the parts of CoHE scored significantly higher in all of them. For validity, there was no significant difference between the two groups.

RQ2. Does the level of confidence of novice evaluators with respect to accuracy of their results differ when they use parts of CoHE compared to HE?

There was a difference between the two groups in terms of how confident they were about the results they produced. The group that used parts of CoHE were more confident in their responses to four statements out of seven. While the group that used traditional HE were more confident in their responses to one statement. While both groups were at the same level of confidence in their responses to one statement.

RQ3. What difficulties do novices face when conducting different parts of CoHE?

Mainly, there was a major difficulty faced by participants who used parts of CoHE. This difficulty was related to the length of the heuristics. A number of participants believed that the heuristics were slightly too long which makes reading them a demanding endeavor which leads to a difficulty in applying them in conducting the evaluation.

RQ4. How do the difficulties faced by novices when conducting parts of CoHE differ from those faced by novices when conducting HE?

As we mentioned above, the length of the heuristics was the main issue novices faced when conducting parts of CoHE. This was different from the difficulties faced by novices who used traditional HE. The participants who used traditional HE had multiple issues. First, that the heuristics don't lend themselves to find usability issues on the system. Second, they are too general which makes understanding them difficult. Third, they lack any explanation of why they should be used. Finally, linking the usability problems was difficult since there was no clear way of how to do such linking.

RQ5. Does novices' judgment of their comprehension of usability heuristics differ when they use parts of CoHE compared to HE?

In general, both groups were positive about their comprehension of the usability heuristics. However, they mentioned multiple points that negatively affected their understanding of the heuristics and these points differed between the groups. Both groups agreed that some of the labels were vague and don't necessarily convey the true meaning of the heuristics. However, the participants who used traditional HE complained about the lack of examples which negatively affected their ability to fully understand the heuristics and that some heuristics seem to potentially have different meanings which was also confusing to them. Moreover, they felt the some of the descriptions were incomplete and contained words that are not clearly defined. Finally, they felt that some of the heuristics were similar which makes them confused and not able to clearly know the difference between them.

On the other hand, the participants who used parts of CoHE didn't complain about the lack of examples. However, they felt that some the examples were not clear. Also, the categorization and the organization of some of the concepts under the heuristics, stating that some of the concepts were not ordered properly and are not placed under the correct heuristics.

6.5. Summary

This section draws out the most important results of the experiment. With regards to the time it took participants to read and inspect the system, the average of the first condition was less than the average of the second condition for both reading and inspection. It took participants in the first condition, on average, 8 minutes and 46 seconds to read and understand the heuristics, while it took participants in the second condition, on average, 28 minutes and 26 seconds to read and understand the heuristics. The difference between the means was significant, $t(18) = 10.519$, $p < .00001$. It took the participants in the first condition, on average, 1 hour, 24 minutes and 7 seconds to inspect the system, while it took the participants in the second condition, on average, 1 hour, 42 minutes and 1 second to inspect the system. The difference between the means was significant, $t(18) = 3.794$, $p = 0.0013$. Thus, the overall time spend on reading and inspecting in the first condition was, on average, 1 hour, 32 minutes and 52 seconds, while, in the second condition, it was, on average, 2 hours, 10 minutes and 27 seconds. The difference between the means was significant, $t(18) = 6.916$, $p < .00001$. Figure 24-Figure 26 illustrate the difference between the two conditions in terms of the time spent.

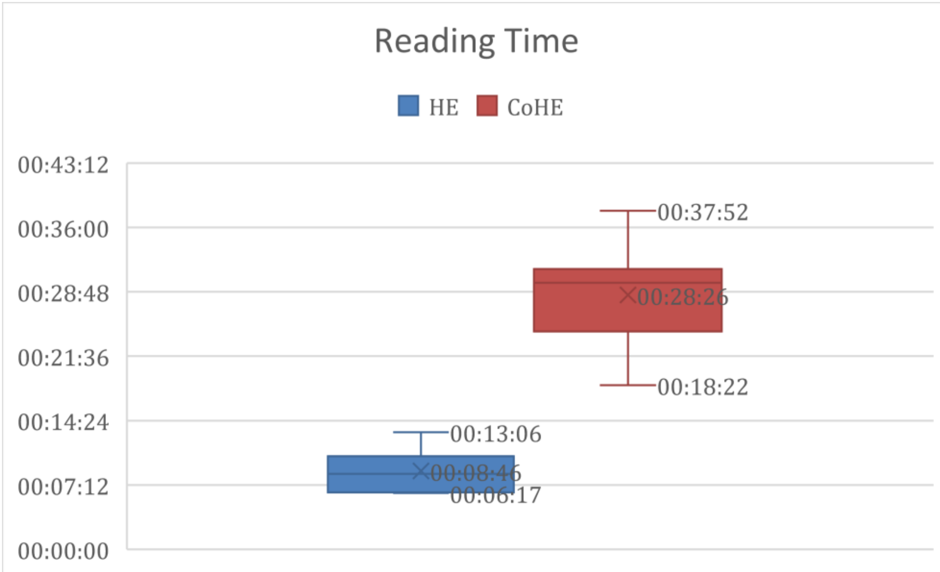

Figure 24. *Reading time*

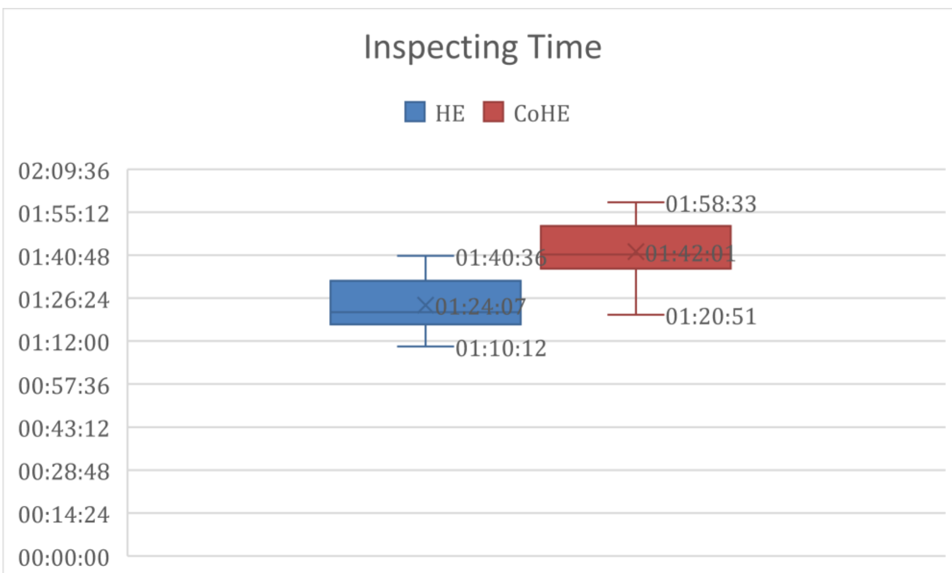

Figure 25. *Inspecting time*

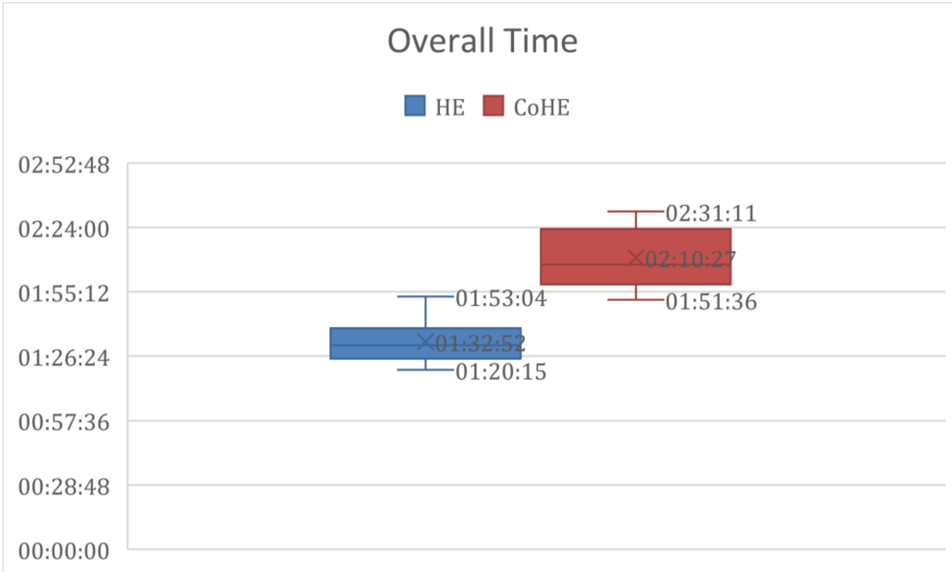

Figure 26. *Overall time*

However, participants in the second condition found more problems on average than participants in the first condition. For all problems found, without any filtration, the mean for participants in the second condition was 27.8 problems; this was higher than the mean for participants in the first condition, which was 22 problems. The median for participants in the second condition was also higher; the median for participants in the second condition was 27, while the median for participants in the first condition was 20.5 problems. The difference between the means was significant, $t(18) = 2.699$, $p = 0.0147$.

For problems filtered by the relaxed criteria, participants in the second condition also found, on average, more problems. The mean of problems found by participants in the second condition was 24.6 problems; this was higher than the mean of problems found by participants in the first condition, which was 19.6 problems. The median of the second condition was also higher; the median of the second condition was 25 problems and the median of the first condition was 18.5 problems. The difference between the means was significant, $t(18) = 3.109$, $p = 0.0061$.

For problems filtered by the strict criteria, participants in the second condition likewise found, on average, more problems. The mean of problems found by participants in the second condition was 20.9 problems; this was higher than the mean of problems found by participants in the first condition, which was 16.5 problems. The median of the second condition was also higher; the median of the second condition was 20.5 problems and the median of the first condition was 15.5 problems. The difference between the means was significant, $t(18) = 3.2071$, $p = 0.0049$. Figure 27–Figure 29 demonstrate the difference between the two conditions in terms of the number of problems found.

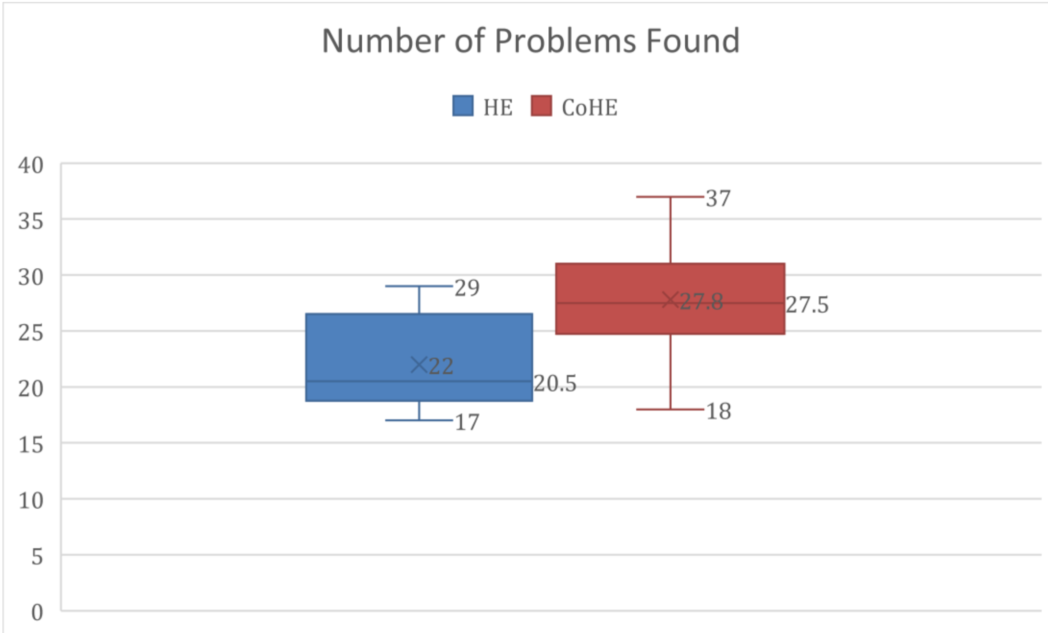

Figure 27. *Number of problems found*

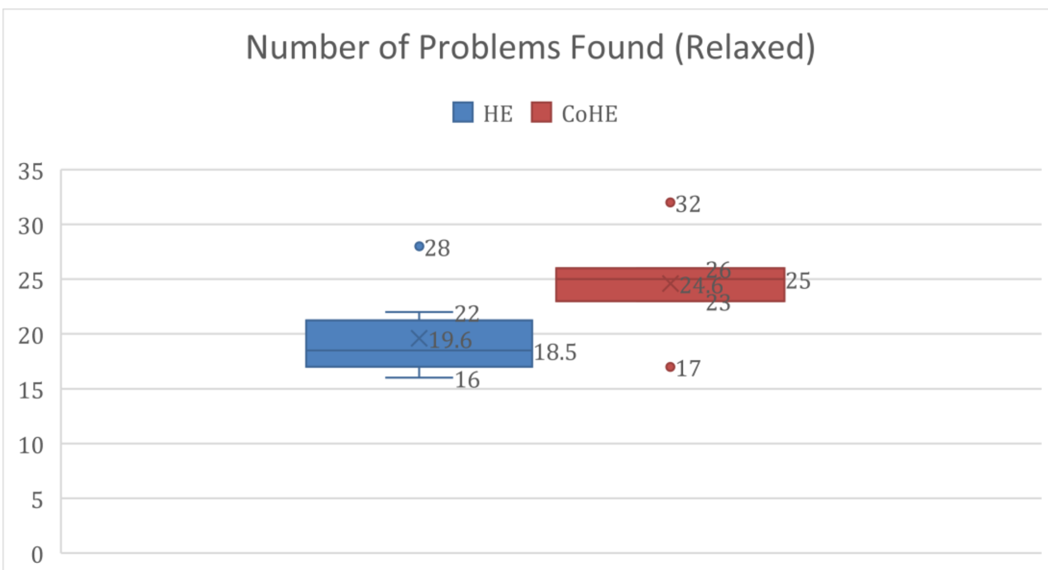

Figure 28. *Number of real problems found (Relaxed)*

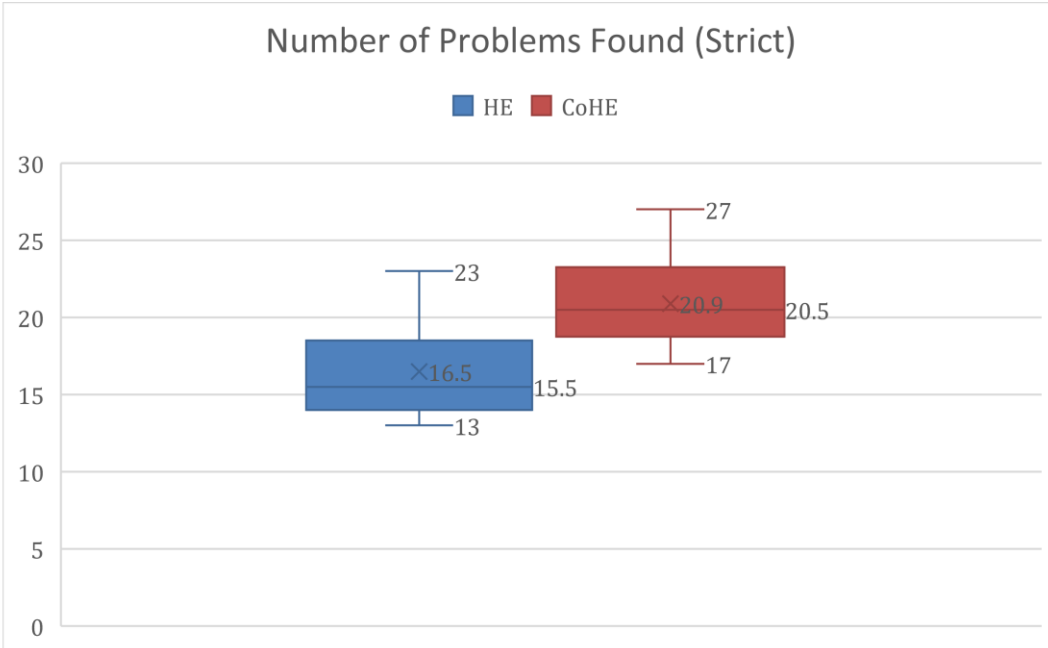

Figure 29. *Number of real problems found (Strict)*

For thoroughness, on average, participants in the second condition were more thorough than participants in the first condition. The mean of the thoroughness of the second condition, for the problems that were filtered by the relaxed criteria, was 12.6%; this was higher than the mean of the first condition, which was 10%. The median of the second condition was also higher than the first condition; the median of the second condition was 12.8% and the median of the first condition was 9.5%. The difference between the means was significant, $t(18) = 3.1176$, $p = 0.0059$.

For the problems filtered by the strict criteria, on average, participants in the second condition were also more thorough than the participants in the first condition. The mean of the participants in the second condition was 10.7%; this was higher than the mean of the participants in the first condition, which was

8.4%. The difference between the means was significant, $t(18) = 3.2095$, $p = 0.0049$. Figure 30–Figure 31 show the difference between the participants in the two conditions in terms of thoroughness.

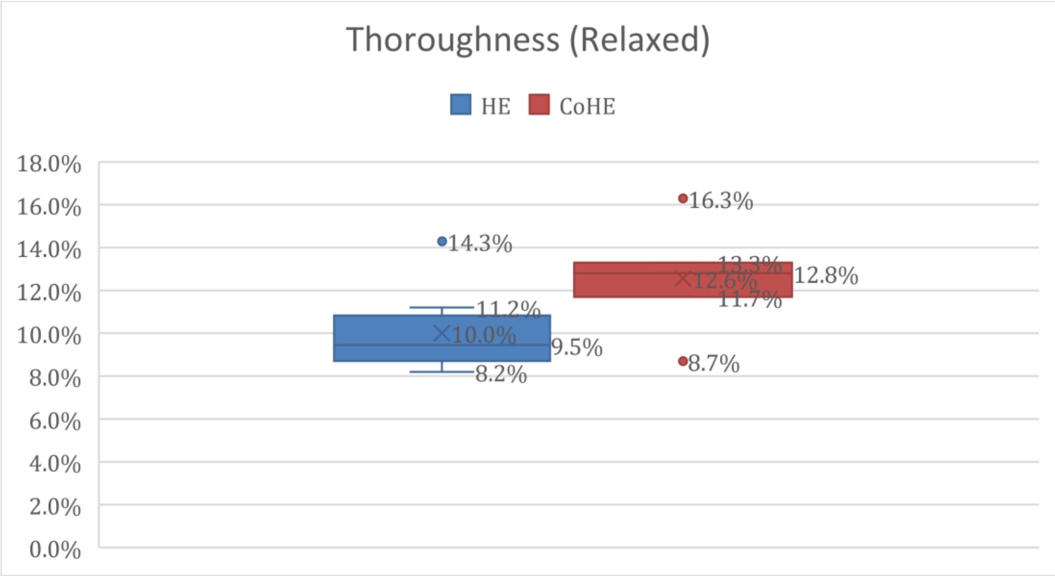

Figure 30. *Thoroughness (Relaxed)*

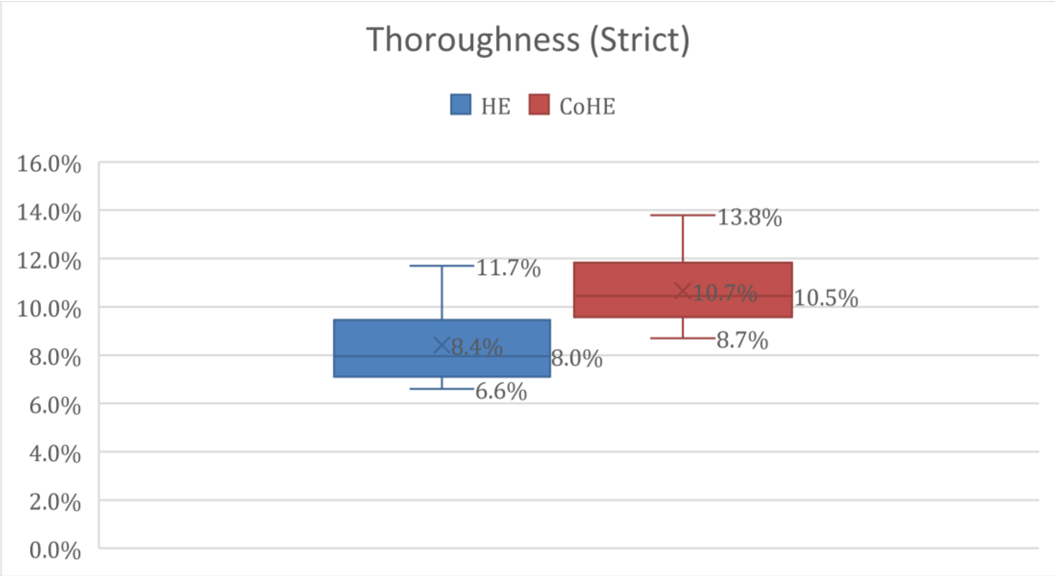

Figure 31. *Thoroughness (Strict)*

For validity, participants in the first condition found, on average, more valid problems than the participants in the second condition, when filtered by the relaxed criteria. The mean of the validity for the first condition was 89.8%; this was higher than the mean of the second condition, which was 89.3%. The median of the first condition, which was 90.3%, was also higher than the median of the second condition, which was 89.1%. The difference between the means was not significant, $t(18) = 0.1315$, $p = 0.8968$.

When the problems were filtered by the strict criteria, the mean of the validity of the second condition was also higher than the mean of the first condition. However, the median of the first condition was higher than the median of the second condition. The mean of the first condition was 75.7%, which was less than the mean of the second condition, which was 76.2%. The median of the first condition was 76.4%, which was higher than the median of the second condition, which was 73%. The difference between the means was not significant, $t(18) = 0.1133$, $p = 0.9110$. Figure 32–Figure 33 illustrate the difference between the two conditions in terms of validity.

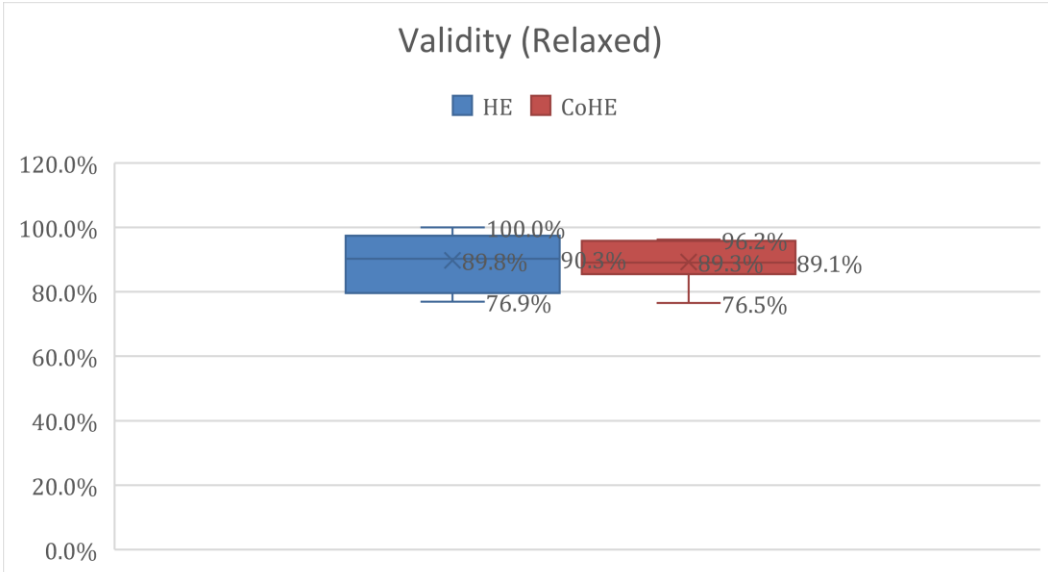

Figure 32. *Validity (Relaxed)*

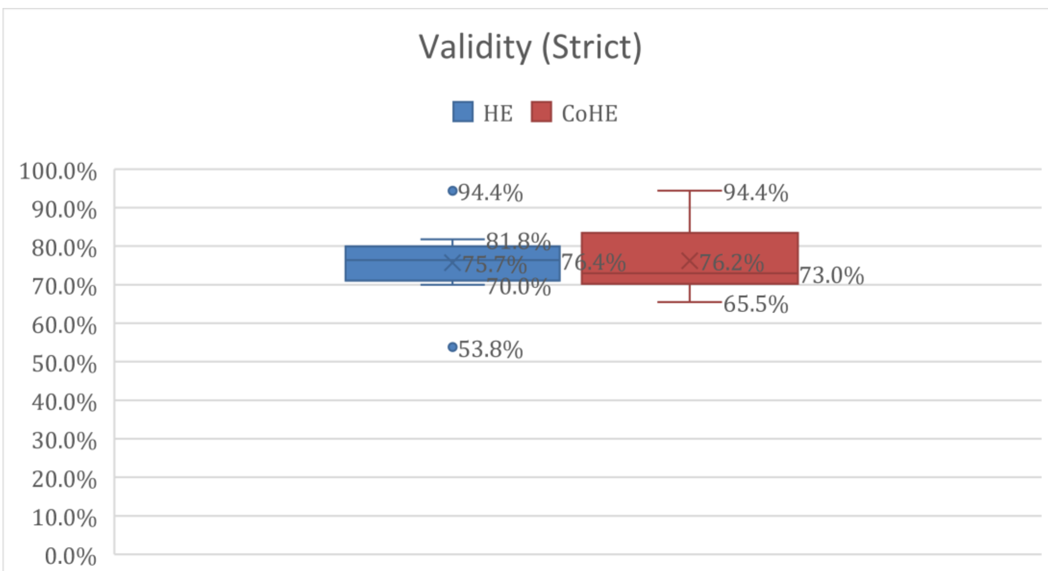

Figure 33. *Validity (Strict)*

For effectiveness, the participants in the second condition were, on average, more effective than the participants in the first condition. When the problems were filtered by the relaxed criteria, the mean of the effectiveness for the second condition was 11.2%; this was higher than the mean of the second condition, which was 9%. The median was also higher; the median of the second condition was 11.1% whereas the median of the first condition was 8.8%. The difference between the means was significant, $t(18) = 2.7733$, $p = 0.0125$.

When the problems were filtered by the strict criteria, the mean and median of the second condition were still higher than the first condition. The mean of the second condition was 8.1%, while the mean of the first condition was 6.4%. Likewise, the median of the second condition was 8.4%, while the median of the first condition was 6.2%. The difference between the means was significant, $t(18) = 2.5781$, $p = 0.0190$. Figure 34-Figure 35 show the difference between the two conditions in terms of effectiveness.

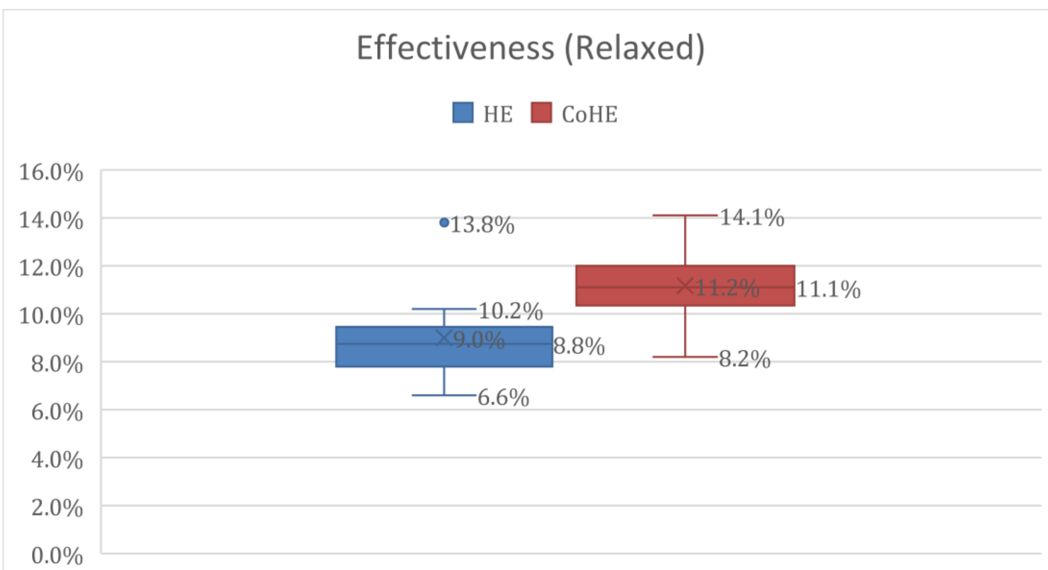

Figure 34. *Effectiveness (Relaxed)*

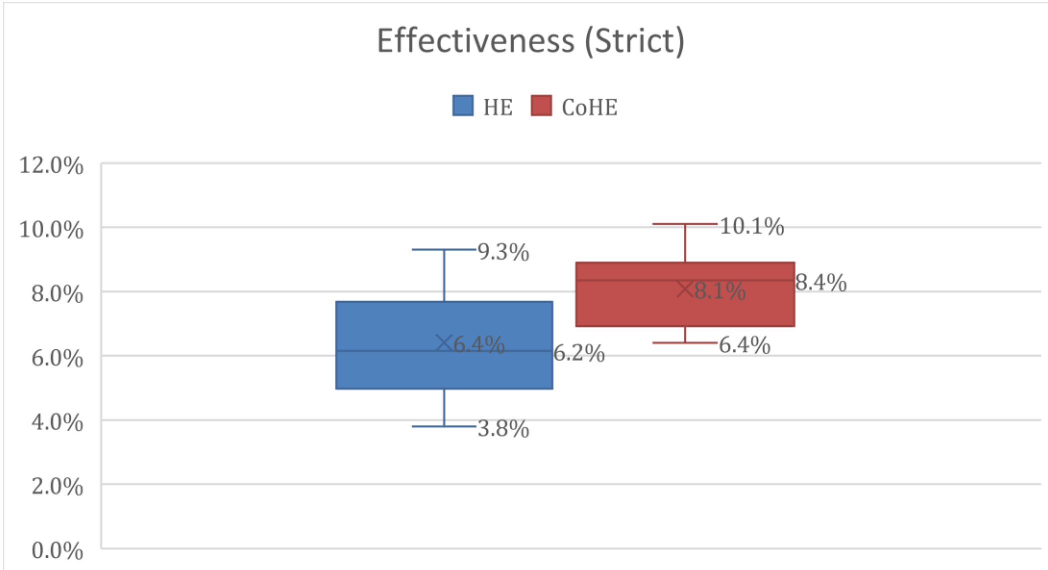

Figure 35. *Effectiveness (Strict)*

For the f-measure, participants in the second condition scored higher, on average, than participants in the first condition. The mean of the f-measure, when filtered by the relaxed criteria, was 21.9% for the second condition; this was higher than the mean of the first condition, which was 17.9%. The median of the second condition, which was 22.3%, was also higher than the median of the first condition, which was 17.2%. The difference between the means was significant, $t(18) = 3.1534$, $p = 0.0055$.

When filtered by the strict criteria, participants in the second condition still scored higher, on average, than participants in the first condition. The mean of the f-measure for the second condition was 18.6%; this was higher than the mean of the first condition which was 15.1 %. The median of the second condition, which was 18.4%, was higher than the median of the second condition, which was 14.4%. The difference between the means was significant, $t(18) =$

3.1952, p = 0.0050. Figure 36-Figure 37 present the difference between the two conditions in terms of f-measure.

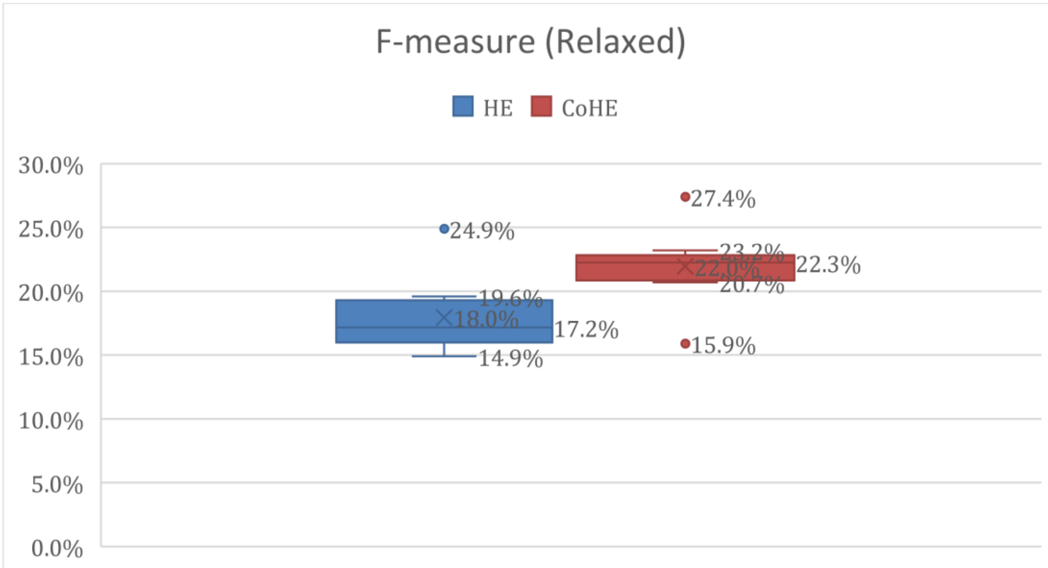

Figure 36. *F-measure (Relaxed)*

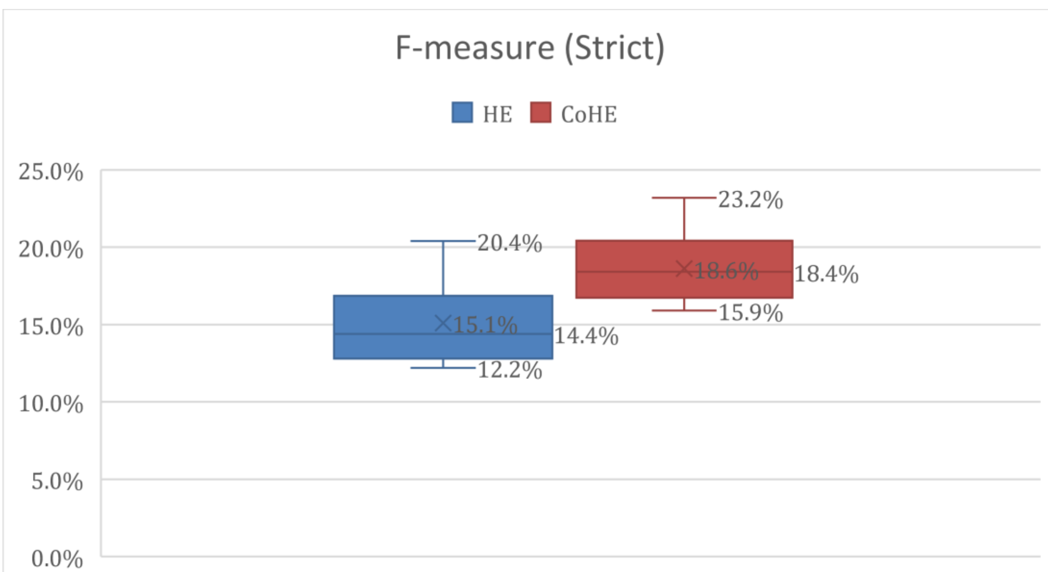

Figure 37. *F-measure (Strict)*

Finally, the number of problems found in each level of severity differed between the two conditions, with participants in the second condition finding, on average, more problems in each severity level.

The mean of the cosmetic problems found by the second condition, which was 0.7 problems, was higher than the mean of the first condition, which was 0.2 problems, and the median of the cosmetic problems of the second condition, which was 0.5 problems, was also higher than the first condition, which was 0 problems. However, The difference between the means was not significant, $t(18) = 1.5230$, $p = 0.1451$.

The mean of the minor problems found by the second condition, which was 3.5 problems, was higher than the mean of the first condition, which was 3.4 problem, and the median of the second condition, which was 3.5 problem, was also higher than the median of the first condition, which was 3 problems. However, The difference between the means was not significant, $t(18) = 0.1562$, $p = 0.8776$.

The mean of the major problems found by the second condition, which was 6.6 problems, was higher than the mean of the first condition, which was 6.1 problems, and the median of the second condition, which was 6 problems, was also higher than the median of the first condition, which was 5.5 problems. However, The difference between the means was not significant, $t(18) = 0.5870$, $p = 0.5645$.

The mean of the catastrophe problems found by the second condition, which was 10.1 problems, was higher than the mean of the first condition, which was 6.8 problems, and the median of the second condition, which was 10 problems, was also higher than the median of the first condition, which was 6.5 problems. The difference between the means was significant, $t(18) = 4.0249$, $p = 0.0008$. Figure 38-Figure 41 show the difference between both conditions in terms of the number of problems found broken down by severity.

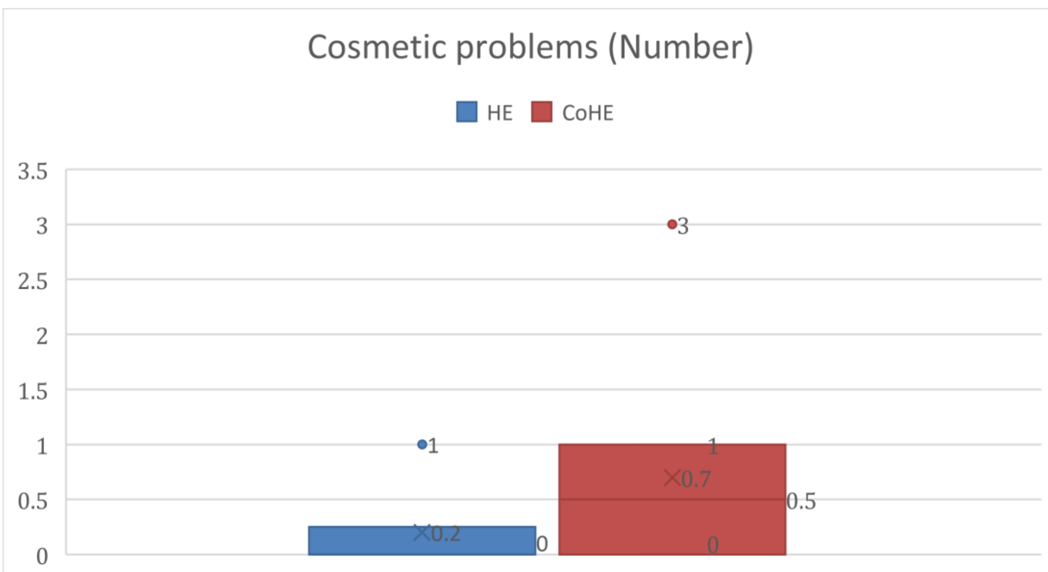

Figure 38. *Number of cosmetic problems*

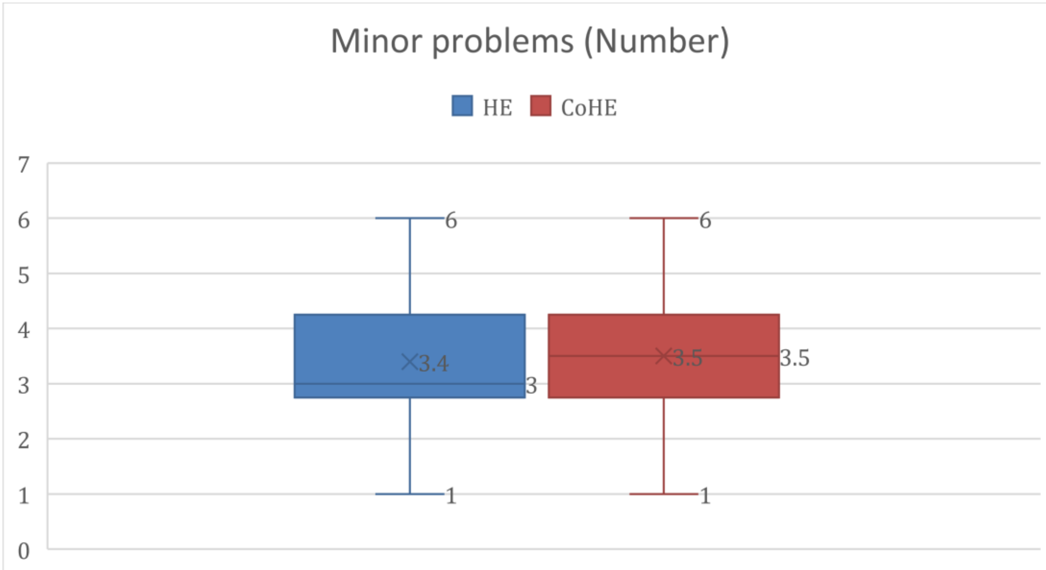

Figure 39. *Number of minor problems*

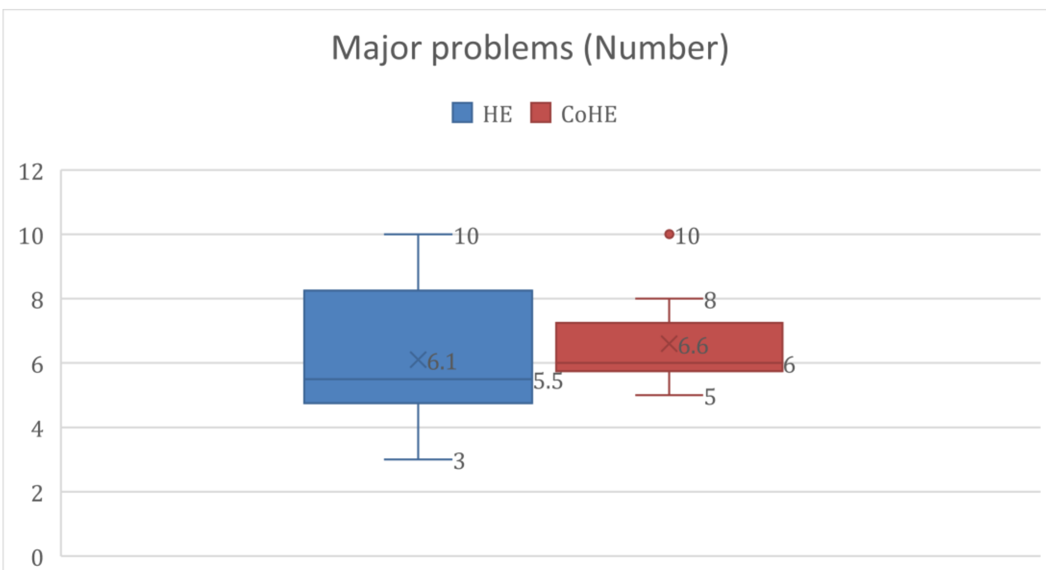

Figure 40. *Number of major problems*

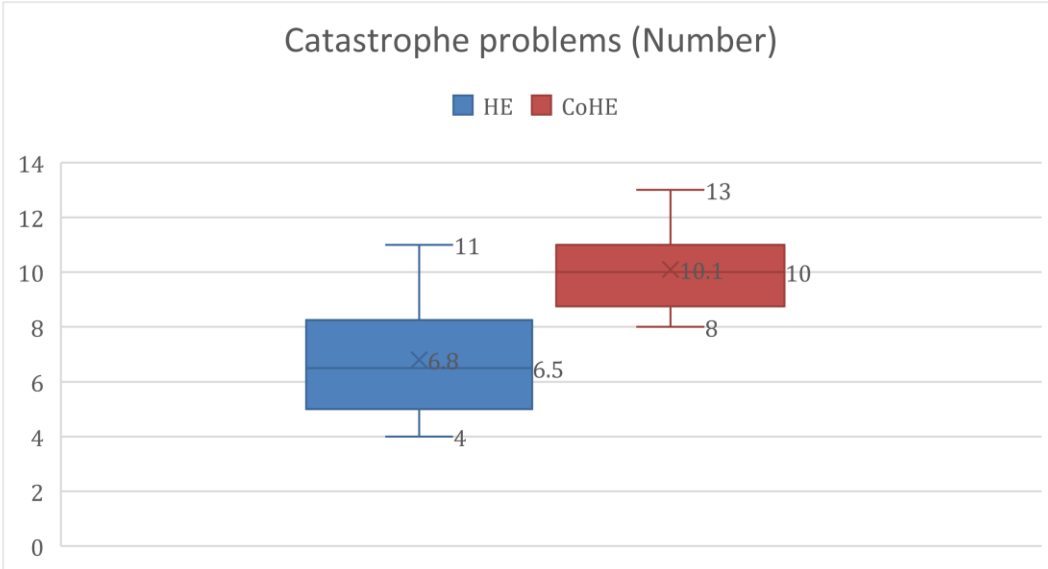

Figure 41. *Number of catastrophe problems*

Calculating the number of problems each participant in each condition found out of all the problems identified in the golden list, participants in the second condition, on average, found more problems in each level than participants in the second condition.

The mean of the second condition for cosmetic problems, which was 2.6%, was higher than the mean of the first condition, which was 0.7%, and the median of the second condition, which was 1.9%, was also higher than the median of the first condition, which was 0%. However, The difference between the means was not significant, $t(18) = 1.5230$, $p = 0.1451$.

The mean of the second condition for minor problems, which was 6.4%, was higher than the mean of the first condition, which was 6.2%, and the median of the second condition, which was 6.4%, was also higher than the median of the first condition, which was 5.5%. However, The difference between the means was not significant, $t(18) = 0.1544$, $p = 0.8791$.

The mean of the second condition for major problems, which was 9.1%, was higher than the mean of the first condition, which was 8.4%, and the median of the second condition was 8.3% which was higher than the median of the first condition which was 7.6%. However, The difference between the means was not significant, $t(18) = 0.5915$, $p = 0.5615$.

Finally, the mean of the second condition for catastrophe problems, which was 24%, was higher than the mean of the first condition, which was 16.2%, and the median of the second condition, which was 23.8%, was also higher than the median of the first condition, which was 15.5%. The difference between the means was significant, $t(18) = 4.0246$, $p = 0.0008$.

Figure 42 – Figure 45 show the difference between the two conditions in terms of the percentage of problems found for each level of severity.

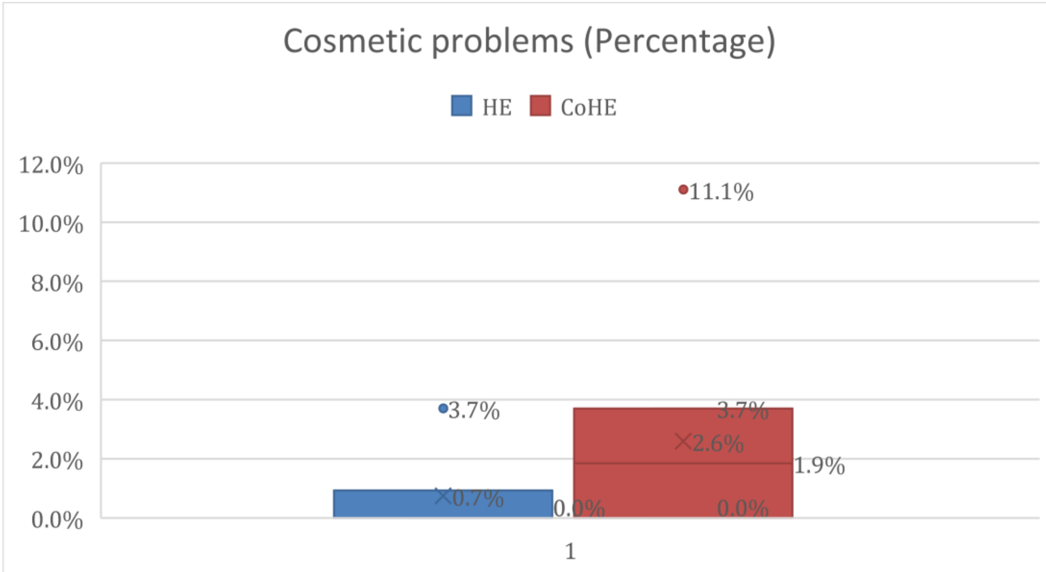

Figure 42. *Percentage of cosmetic problems*

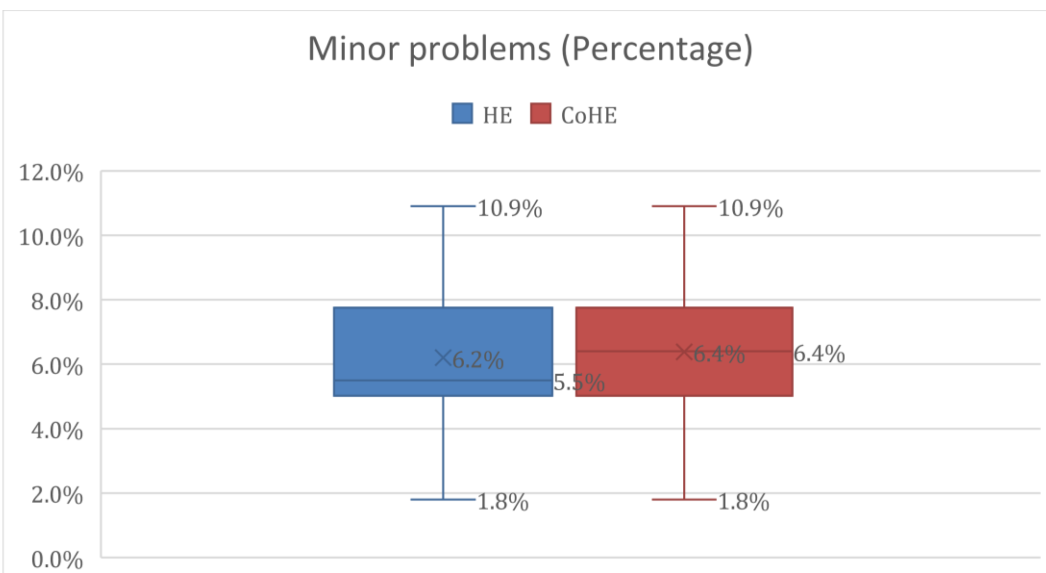

Figure 43. *Percentage of minor problems*

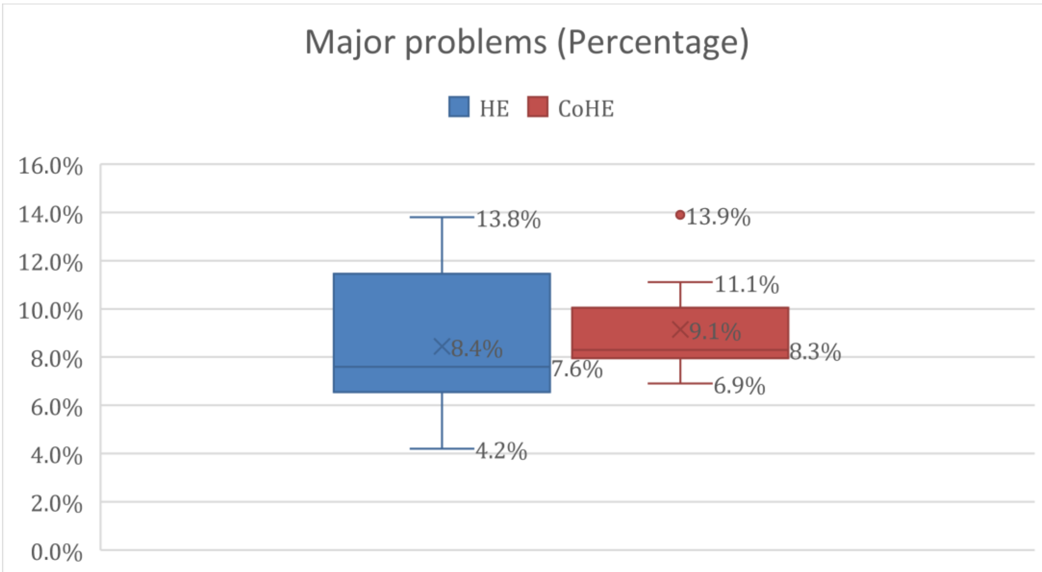

Figure 44. *Percentage of major problems*

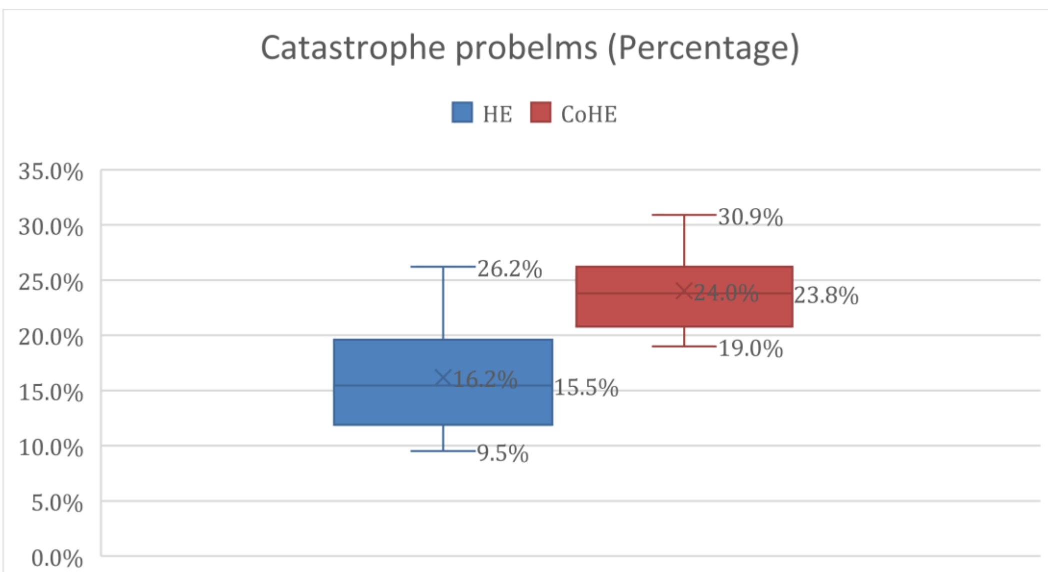

Figure 45. *Percentage of catastrophe problems*

CHAPTER 7. DISCUSSION

Having presented the results of the experiment, there are several points that deserve to be discussed with regards to this research. This chapter discusses these points, shedding light on the limitations of the research and the difficulties faced during the research (Figure 46).

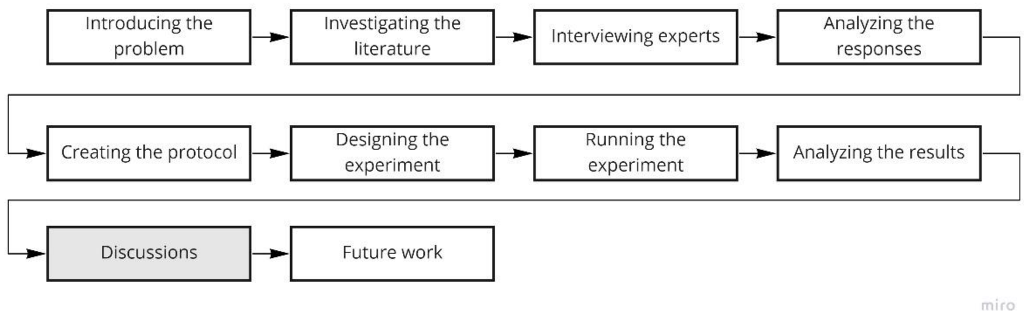

Figure 46. *Thesis organization (Chapter 7)*

7.1. Limited time for the evaluation

In real life contexts, doing a complete and careful evaluation takes a significant amount of time. The recommended time for an evaluation is 2 hours, but evaluations often take more time than this in reality. In this experiment, we followed the recommended 2-hour time limit for several reasons. First, it would have been difficult to ask participants to spend more than two hours evaluating a system, as this would be a large time commitment to ask from them. Second, even if participants agreed in principle to spend more than two hours evaluating a system, it would be extremely overwhelming for them to do this in practice. They would be likely to become tired and lose focus, which would have damaged the quality of their performance. Third, analyzing all the sessions proved to be very demanding and having sessions longer than 2 hours would have made this process even more difficult.

For these reasons, we asked participants to perform an evaluation in 2 hours, but it is worth recognizing that it is not necessarily possible for participants to do a careful evaluation within 2 hours. Given this limitation, we expect that the results of these evaluations are not the best quality possible from the participants and thus might not reflect the true potential of the participants, and the methods, if more time was available to inspect the system more carefully.

7.2. Performing the evaluation in one session

Not all evaluators prefer to perform their tasks in one session; some people prefer to divide their tasks and complete them separately, taking breaks as they go. With this in mind, as part of CoHE, we recommend that evaluators, especially novices, should take breaks every 45 minutes. However, given the difficulty of scheduling sessions, this was not possible in the context of the experiment. As participants had other responsibilities in their lives, scheduling the sessions proved to be difficult, so asking them to schedule multiple sessions to facilitate a longer process would have been even more difficult. This was the reason for scheduling two sessions only, one for reading and one inspecting.

In line with the limited time overall, this approach may have affected the quality of the evaluations produced by the evaluators. If they were given more time to take brakes and return to the evaluation fresh, we believe this could have improved the evaluations and further demonstrated the potential of the methods.

7.3. Performing parts of CoHE

CoHE is a comprehensive protocol, meaning that it was designed to guide novice evaluators from the moment they decide to perform an evaluation until the submission of their evaluation report. Given that this is a lengthy process, and we were confined to a 2-hour time frame, we were not able to ask evaluators to follow the whole protocol step-by-step. Instead, we presented evaluators with the most relevant parts of the protocol, namely the detailed usability heuristics and the operational usability heuristics. Likewise, the participants who evaluated the system based on Nielsen's heuristics were only presented with the heuristics rather than Nielsen's broader guidance on evaluation.

Despite these limitations, this experiment offers an indication of the quality of the protocol. Nevertheless, it does not necessarily reflect the true potential of it. We believe that, if both conditions were given the full materials, meaning the whole CoHE protocol and Nielsen's wider process, the results might have been different.

7.4. The realism of the method

When conducting evaluations in real life, evaluators do not usually depend on only one source for their evaluation; they depend on multiple sources. They may use certain heuristics but search for complementary material to find more detailed explanations or look up some terms to understand them better. However, in the experiment, participants in both conditions were instructed to use only the materials provided, without searching for complementary material. This aimed to help control the experiment and make sure the participants were performing the same tasks in a comparable way. If participants were allowed to perform the evaluation in their own way, we think they may have looked for complementary material. If this had been the case, we believe that the results of the evaluation may have been different.

7.5. Monitoring the sessions

As outlined in Chapter 5, all of the sessions were monitored and recorded, which required the researcher to be present during the reading and inspecting sessions. The presence of the researcher may have influenced the evaluation process, as some evaluators, particularly novices, may have felt uncomfortable with being observed while reading or evaluating. Although the presence of the researcher was necessary for noting participants' comments, asking questions when necessary, and observing the flow of the process, this may have affected the evaluation process and, if the participants were not being monitored or recorded, the results may have been different.

7.6. Giving participants certain tasks

In HE, there are different ways of approaching a system when evaluating it. Some people take a page-by-page approach, some people follow certain tasks, some people evaluate certain pages only, and so on. In this experiment, we gave participants certain tasks to follow, rather than allowing participants to evaluate the system in their own way. As with only using one source, the reason for this was to control the research process and make sure that all participants were exposed to the same design elements when conducting their evaluation. We expect that, if the participants were free to evaluate the system in their own way, they may have performed the evaluation differently, which might have lead them to detect different problems.

7.7. Participants trying different approaches

Although participants were assigned the same tasks, they nevertheless took different approaches. They used different inputs or they performed the tasks following different routes. This means that, ultimately, they were not necessarily exposed to exactly the same design elements. This is difficult to control, since most tasks have multiple ways they can be completed and we did not want to put further controls on the evaluation given that figuring out the task is part of completing the evaluation itself. We suspect that, if we had given the participants the exact same routes and the exact same inputs, the results might have been different.

7.8. Number of participants

As outlined above, the minimum number of participants considered to produce meaningful results is between 10–12 participants for each condition. We recruited 10 participants for each condition; while we wanted to recruit more participants, the duration of the research, requiring each participant to undergo 4 hours in total, made recruitment very difficult. Participants had to find time in their busy schedule for two sessions on two separate days and commit to spending two hours concentrating on a task. While having 28 participants, with 10 in each condition and eight experts, was not ideal, we believe it was a reasonable sample given the difficulties faced during recruitment and in accommodating different availabilities when scheduling the sessions.

7.9. Heterogeneous groups

With the difficulties in recruitment, we were unable to recruit participants for both conditions from a homogenous group. We could not recruit participants from the same class or the same university. Therefore, we recruited participants from multiple universities who met the same criteria defined prior to recruitment. To reduce the influence of this issue on the results, we recruited an equal number of participants from each university and divided them randomly and equally into both conditions. As a result, we had two participants from each university in each group.

7.10. Golden list

In defining the real problems, we used a golden list composed of all the problems found by the experts, comparing these with the problems found by the novices. This involved an assumption that the experts found all of the problems on the system, which might not have been the case. However, there is no simple way to produce a golden list containing all the problems existing on a system. Indeed, it is a common practice in other research to use a golden list either consisting of problems found by usability experts or found in usability testing sessions. Nevertheless, in both cases, there is no guarantee that all the issues are identified. As such, there is a possibility that a real problem that was found by a novice evaluator that was considered to be a false positive since it was not found on the golden list.

CHAPTER 8. CONCLUSION AND FUTURE WORK

This research is understood as a first step in investigating the subject of facilitating heuristic evaluation for novice evaluators. We intend to build upon this work in the future, either by modifying and updating the CoHE protocol or by organizing more comprehensive testing to measure the impact it has on novice evaluators. In this final chapter, we offer a conclusion for the research undertaken and outline our intentions for future work (Figure 47).

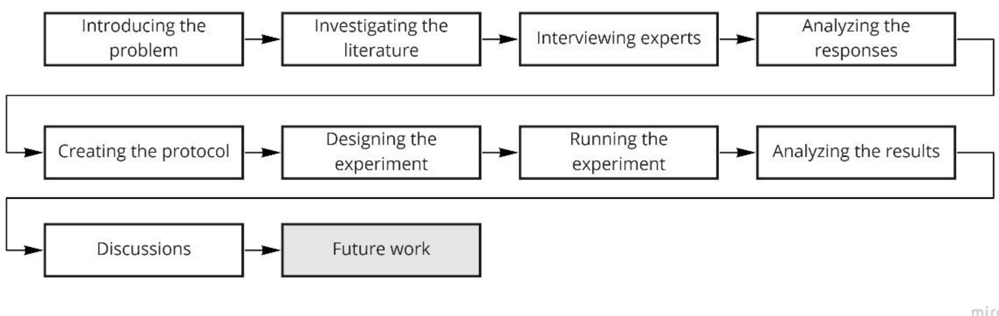

Figure 47. *Thesis organization (Chapter 8)*

8.1. Conclusion

Heuristic evaluation is one of the most widely used usability evaluation methods. The idea of heuristic evaluation as a method is simple: an evaluator evaluates a system guided by a set of usability heuristics. Despite this, HE suffers from a major problem: the expertise effect. This means that the quality of evaluations produced by experts differs significantly from those produced by novices. While this discrepancy has made some researchers question the utility of the method, claiming that the results are the product of the evaluator's experience rather than the method itself, the aim of this research was to challenge this perspective by enhancing the performance of novice evaluators when using HE.

To this end, we investigated the issues that make HE difficult for novices. We interviewed 15 usability experts, asking them to explain in detail how they perform HE. We also asked them to reflect on the issues they faced when they started using the method and how they overcame these issues. Moreover, we asked them to discuss in detail the Nielsen's 10 usability heuristics, getting them to explain them, give examples, and describe their significance, their applicability, and the consequences of ignoring them.

Based on analyzing the responses of the experts, we created a protocol called Coherent Heuristic Evaluation (CoHE). This step-by-step protocol aimed to support novices perform HE, from the moment they decide to undertake an evaluation until the point of submitting the evaluation report. The protocol we developed consists of three stages; the understanding stage, which guides the evaluators through a process of reading and comprehending the usability heuristics; the inspecting stage, which supports the evaluators through the process of inspecting the system and finding usability problems; and the reporting stage, which guides the evaluators through preparing the report and explaining the problems they found and the reasons they should be fixed.

Having created this protocol, we set out to test it, designing an experiment to compare CoHE with unguided HE. We recruited 28 participants, including 20 novices and eight experts, and we divided the novices into two groups: the first group were given Nielsen's 10 usability heuristics, while the second group were given parts of CoHE, namely detailed usability heuristics for understanding and operational usability heuristics for inspection. Each participant in each group performed two sessions, one focused on understanding and another involving evaluation, while the experts only performed one session, during which they evaluated the system to create a golden list.

After analyzing the sessions, noting all the problems found by the novices in both groups and using the golden list as a benchmark, we found that participants who used parts of CoHE took significantly more time reading and evaluating the system than those who used Nielsen’s 10 usability heuristics. However, participants who used parts of CoHE were able to find significantly more problems than those who used Nielsen’s 10 usability heuristics, whether problems were unfiltered, filtered by the relaxed criteria, or filtered by the strict criteria. The problems found by participants who used parts of CoHE were significantly more thorough and effective, and scored higher on the f-measure, than those who used Nielsen’s 10 usability heuristics, but the there was no significant difference between both groups in terms of validity. When it comes to severity levels, participants who used parts of CoHE were able, on average, to find more problems in each level of severity than participants who used Nielsen’s 10 usability heuristics and the thoroughness of the problems found for each level of severity was also higher for participants who used parts of CoHE than for the participants who used Nielsen’s 10 usability heuristics. However, the difference was only significant for the catastrophe level.

8.2. Future work

As stated above, we consider this research a first step in our work on facilitating heuristic evaluation for novice evaluators. In the future, we intend to follow multiple directions to investigate this matter further and continue developing the CoHE protocol.

First, we want to examine the CoHE protocol more carefully. The current version of CoHE is the result of interviewing 15 usability experts. The interviews were around 2 hours long and the questions focused on their experience. In interviews, sometimes interviewees forget or confuse information or offer suggestions they do not necessarily follow exactly. Therefore, we intend to undertake contextual inquiries, visiting usability experts and observing how they perform the evaluation themselves, to gain deeper insights into the process. The results of this could involve modifying, updating and enriching the protocol.

Second, we want to design experiments to test the whole CoHE protocol. In this research, we were only able to test certain parts of the protocol. Therefore, we want to test the entire protocol to observe its effects on novice evaluators and investigate its full potential.

Third, as shown in the experiment, using parts of CoHE took more time than using Nielsen's 10 usability heuristics. We want to conduct a longitudinal study to ascertain whether the time required to use the protocol subsides over time. This would also enable us to note any issues or advantages that come from using the protocol over a long period of time.

Fourth, we want to test the protocol on different populations. The beauty of HE is its simplicity; therefore, we want to see whether different populations, such as developers, designers, technicians, and even possibly people with no formal training in IT, could benefit from it. This would also give us the opportunity to observe the differences between the different populations when using the protocols. Moreover, we want to find out whether usability experts could also benefit from using CoHE.

Fifth, we want to see the relation between the time spent in the evaluation, and the number of problems found. We want to have two groups, one using traditional HE and one using the CoHE do the evaluation with exactly the same duration. Then, we want to see if the difference between the two is still significant or not.

Finally, given that CoHE is general, it can be used with any set of usability heuristics, so we want to test using CoHE alongside different sets of usability heuristics. This would allow us to gain an idea of how well CoHE works with different heuristics and which ones are the best to use with it.

REFERENCES

Alqurni, J., Alroobaea, R., & Alqahtani, M. (2018). Effect of user sessions on the heuristic usability method. *International Journal of Open Source Software and Processes*, *9*(1), 62–81. https://doi.org/10.4018/IJOSSP.2018010104

Alsumait, A., & Al-Osaimi, A. (2010). Usability heuristics evaluation for child e-learning applications. *Journal of Software*, *5*(6), 654–661. https://doi.org/10.4304/jsw.5.6.654-661

Ariffin, S. A., & Dyson, L. E. (2015). Culturally appropriate design of mobile learning applications in the Malaysian context. In *Cross-cultural design methods, practice and impact* (pp. 3–14). Springer.

Atkinson, B. F., Bennett, O. T., Bahr, G. S., & Nelson, M. M. (2007). In C. Stephanidis (Ed.), *Universal access in human-computer interaction* (pp. 563–572). Springer.

Aydin, B., Palikhe, H., & Beruvides, M. (2012). The impact of usability on the cost of quality. *International annual conference of the American society for engineering management* (pp. 632–637). ASEM.

Babajo, A., & Petrie, H. (2012). *The effectiveness of collaborative heuristic evaluation* [Unpublished master’s thesis]. The University of York.

Babbie, E. R. (2013). *The practice of social research* (13th ed.). Wadsworth Cengage Learning.

Baker, K., Greenberg, S., & Gutwin, C. (2002). Empirical development of a heuristic evaluation methodology for shared workspace groupware. *Proceedings of the 2002 ACM conference on computer supported cooperative work*. ACM Press. https://doi.org/10.1145/587078.587093

Al-Banna, M., Benatallah, B., & Barukh, M. C. (2016). Software security professionals: Expertise indicators. *IEEE 2nd international conference on collaboration and internet computing (CIC)* (pp. 139–148). IEEE. https://doi.org/10.1109/CIC.2016.030

Barnum, C. M. (2011). *Usability testing essentials: Ready, set . . . test!* Morgan Kaufmann Publishers.

Bennett, J. (1984). *Managing to meet usability requirements: Establishing and meeting software development goals*. Prentice-Hall.

Blandford, A., Furniss, D., & Makri, S. (2016). *Qualitative HCI research: Going behind the scenes* [PowerPoint slides]. Synthesis lectures on human-centered informatics.

Bond, C. (2007). Time is money: Impacts of usability at a call center. *Ergonomics in Design: The Quarterly of Human Factors Applications*, *15*(4), 17–22. https://doi.org/10.1518/106480407X255224

Botella, F., Alarcon, E., & Peñalver, A. (2014). How to classify to experts in usability evaluation. *Proceedings of the 15th international conference on human computer interaction*. ACM Press. https://doi.org/10.1145/2662253.2662278

Botella, F., Rusu, C., Rusu, V., & Quiñones, D. (2018). How novel evaluators perceive their first heuristic evaluation. *Proceedings of the 19th international conference on human computer interaction*. ACM Press. https://doi.org/10.1145/3233824.3233835

Bruun, A., & Stage, J. (2014). Barefoot usability evaluations. *Behaviour & Information Technology*, *33*(11), 1148–1167. https://doi.org/10.1080/0144929X.2014.883552

Bruun, A., & Stage, J. (2015). New approaches to usability evaluation in software development: Barefoot and crowdsourcing. *Journal of Systems and Software*, *105*, 40–53. https://doi.org/10.1016/j.jss.2015.03.043

Chanco, C., Moquillaza, A., & Paz, F. (2019). Development and validation of usability heuristics for evaluation of interfaces in ATM. In P. Zaphiris & A. Ioannou (Eds.), *Learning and collaboration technologies: Human and Technology Ecosystems* (pp. 3–18). Springer.

Cockton, G., & Woolrych, A. (2001). Understanding inspection methods: Lessons from an assessment of heuristic evaluation. In A. Blandford (Ed.), *People and Computers — Interaction without Frontiers: Joint Proceedings of HCI 2001 and IHM 2001* (pp. 171–191). Springer.

Cockton, G., & Woolrych, A. (2002). Sale must end: Should discount methods be cleared off HCI's shelves? *Interactions*, *9*(5), 13–18. https://doi.org/10.1145/566981.566990

Cockton, G., Woolrych, A., Hall, L., & Hindmarch, M. (2004). Changing analysts' tunes: The surprising impact of a new instrument for usability inspection method assessment. *Proceedings of the ASME design engineering technical conferences and computers and information in engineering conference 2004* (pp. 145–161). ASME.

Cope, M. E., & Uliano, K. C. (1995). Cost-justifying usability engineering: A real world example. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *39*(4), 263–267. https://doi.org/10.1177/154193129503900409

Cronholm, S. (2009). The usability of usability guidelines: A proposal for meta-guidelines. *Proceedings of the 21st annual conference of the Australian computer-human interaction special interest group on design*. ACM Press. https://doi.org/10.1145/1738826.1738864

Desurvire, H., & Thomas, J. C. (1993). Enhancing the performance of interface evaluators using non-empirical usability methods. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *37*(17), 1132–1136. https://doi.org/10.1177/154193129303701702

Desurvire, H., Caplan, M., & Toth, J. A. (2004). Using heuristics to evaluate the playability of games. *Extended abstracts of the 2004 conference on human factors and computing systems*. ACM Press. https://doi.org/10.1145/985921.986102 van Dijk, J., Pieterson, W., van Deursen, A., Ebbers, W., Wimmer, M. A., Scholl, J., & Grönlund, A. (2007). E-services for citizens: The Dutch usage case. *Proceedings of the international conference on electronic government* (pp. 155–166). Springer. https://doi.org/10.1007/978-3-540-74444-3_14s

Dumas, J. S. (2003). User-based evaluations. In J. A. Jacko & A. Sears (Eds.), *Human factors and ergonomics — The human-computer interaction handbook: Fundamentals, evolving technologies, and emerging applications* (pp.1093–1117). Lawrence Erlbaum Associates.

Ericsson, K. A., Prietula, M. J., & Cokely, E. T. (2007, July 1). The making of an expert. *Harvard Business Review*. https://hbr.org/2007/07/the-making-of-an-expert

Friess, E. (2015). Personas in heuristic evaluation: An exploratory study. *IEEE Transactions on Professional Communication*, *58*(2), 176–191. https://doi.org/10.1109/TPC.2015.2429971

Gerhardt-Powals, J. (1996). Cognitive engineering principles for enhancing human-computer performance. *International Journal of Human-Computer Interaction*, *8*(2), 189–211. https://doi.org/10.1080/10447319609526147

Gibbs, W. W. (1997). Taking computers to task. *Scientific American*, *277*(1), 82–89. https://doi.org/10.1038/scientificamerican0797-82

Granollers, T. (2018). Usability evaluation with heuristics. In W. Wang (Ed.), *Design, user experience, and usability: Theory and practice* (pp. 396–340). Springer.

Gray, W. D., & Salzman, M. C. (1998). Damaged merchandise? A review of experiments that compare usability evaluation methods. *Human-Computer Interaction*, *13*(3), 203–261. https://doi.org/10.1207/s15327051hci1303_2

Guest, G., Bunce, A., & Johnson, L. (2006). How many interviews are enough? An experiment with data saturation and variability. *Field Methods*, *18*(1), 59–82. https://doi.org/10.1177/1525822X05279903

Hambrick, D. Z., Oswald, F. L., Altmann, E. M., Meinz, E. J., Gobet, F., & Campitelli, G. (2014). Deliberate practice: Is that all it takes to become an expert? *Intelligence*, *45*, 34–45. https://doi.org/10.1016/j.intell.2013.04.001

Han, S. H., Yun, M. H., Kwahk, J., & Hong, S. W. (2001). Usability of consumer electronic products. *International Journal of Industrial Ergonomics*, *28*(3–4), 143–151. https://doi.org/10.1016/S0169-8141(01)00025-7

Han, Y. Y. (2005). Unexpected increased mortality after implementation of a commercially sold computerized physician order entry system. *Pediatrics*, *116*(6), 1506–1512. https://doi.org/10.1542/peds.2005-1287

Harrison, M. C., Henneman, R. L., & Blatt, L. A. (1994). Design of a human factors cost-justification tool. In R. G. Bias & D. J. Mayhew (Eds.), *Cost-justifying usability* (pp. 203–241). Academic Press.

Hartson, H. R., Andre, T. S., & Williges, R. C. (2001). Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer*

Interaction, *13*(4), 373–410. https://doi.org/10.1207/S15327590IJHC1304_03

Hartson, H. R., Andre, T. S., & Williges, R. C. (2003). Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction*, *15*(1), 145–181. https://doi.org/10.1207/S15327590IJHC1501_13

Hussein, I., Mahmud, M., & Tap, A. O. (2014). A survey of user experience practice: A point of meet between academic and industry. In S.-F. Yang (Ed.), *Information technology applications in industry III* (pp. 62–67). Trans Tech Publications.

Hvannberg, E. T., Law, E. L.-C., & Lárusdóttir, M. K. (2007). Heuristic evaluation: Comparing ways of finding and reporting usability problems. *Interacting with Computers*, *19*(2), 225–240. https://doi.org/10.1016/j.intcom.2006.10.001

Iivari, N. (2006). Understanding the work of an HCI practitioner. *Proceedings of the 4th Nordic conference on human-computer interaction changing roles* (pp. 185–194). ACM Press. https://doi.org/10.1145/1182475.1182495

Inostroza, R., Rusu, C., Roncagliolo, S., & Rusu, V. (2013). Usability heuristics for touchscreen-based mobile devices: Update. *Proceedings of the 2013 Chilean conference on human-computer interaction*. ACM Press. https://doi.org/10.1145/2535597.2535602

International Organization for Standardization. (2018). *Ergonomics of human-system interaction — Usability: Definitions and concepts* (ISO No. 9241–9251). Retrieved from https://kebs.isolutions.iso.org/obp/ui#iso:std:iso:9241:-11:ed-2:v1:en

Issa, T., & Isaias, P. (2015). *Sustainable design: HCI, usability and environmental concerns*. Springer.

Issa, T., & Isaias, P. (2015). Usability and human computer interaction. In Y. Xiao (Ed.), *International conference on computer graphics, visualization, computer vision and image processing*. IADIS Press.

Karat, C.-M. (2005). A business case approach to usability cost justification for the web. In *Cost-justifying usability* (pp. 103–141). Elsevier. https://doi.org/10.1016/B978-012095811-5/50004-3

Al-Khalifa, H. S., Al-Twaim, B., & AlHarbi, B. (2016). A heuristic checklist for usability evaluation of Saudi government mobile applications. *Proceedings of the 18th international conference on information integration and web-based applications and services* (pp. 375–378). ACM Press. https://doi.org/10.1145/3011141.3011189

Korhonen, H., & Koivisto, E. M. I. (2006). Playability heuristics for mobile games. *Proceedings of the 8th conference on human-computer interaction with mobile devices and services*. ACM Press. https://doi.org/10.1145/1152215.1152218

Koutsabasis, P., Spyrou, T., Darzentas, J. S., & Darzentas, J. (2007). On the performance of novice evaluators in usability evaluations. *Proceedings of the Pan-Hellenic conference on human-computer interaction.* PCI.

Kurosu, M., Matsuura, S., & Sugizaki, M. (1997). Categorical inspection method-structured heuristic evaluation (SHEM). *1997 IEEE international conference on systems, man, and cybernetics: computational cybernetics and simulation* (pp. 2613–2618). IEEE. https://doi.org/10.1109/ICSMC.1997.635329

Kurosu, M., Sugizaki, M., & Matsuura, S. (1999). In H.-J. Bullinger & J. Ziegler (Eds.), *Human-computer interaction: Proceedings of the 8th international conference on human-computer interaction* (pp. 938–942). Erlbaum.

Kushniruk, A. W., Triola, M. M., Borycki, E. M., Stein, E., & Kannry, J. L. (2005). Technology induced error and usability: The relationship between usability problems and prescription errors when using a handheld application. *International Journal of Medical Informatics*, *74*(7–8), 519–526. https://doi.org/10.1016/j.ijmedinf.2005.01.003

Lanzilotti, R., Ardito, C., Costabile, M. F., & De Angeli, A. (2011). Do patterns help novice evaluators? A comparative study. *International Journal of Human-Computer Studies*, *69*(1–2), 52–69. https://doi.org/10.1016/j.ijhcs.2010.07.005

Lazar, J., Feng, J. H., & Hochheiser, H. (2017). *Research Methods in Human-Computer Interaction*. https://doi.org/10.1016/B978-0-12-805390-4.09991-X

Lewis, C., Polson, P. G., Wharton, C., & Rieman, J. (1990). Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. *Proceedings of the SIGCHI conference on human factors in computing systems empowering people*. ACM Press. https://doi.org/10.1145/97243.97279

Lewis, C., & Wharton, C. (1997). Cognitive walkthroughs. In *Handbook of Human-Computer Interaction* (2nd ed., pp. 717–732).

Liljegren, E. (2006). Usability in a medical technology context assessment of methods for usability evaluation of medical equipment. *International Journal of Industrial Ergonomics*, *36*(4), 345–352. https://doi.org/10.1016/j.ergon.2005.10.004

de Lima Salgado, A. (2017). *Adaptations to the heuristic evaluation (HE) method for novice evaluators* [Unpublished master's thesis]. University of São Paulo.

de Lima Salgado, A., de Lara, S. M. A., Pimenta Freire, A., & de Mattos Fortes, R. P. (2016). What is hidden in a heuristic evaluation: Tactics from the experts. *13th international conference on information systems and technology management* (pp. 2931–2946). CONTECSI. https://doi.org/10.5748/9788599693124-13CONTECSI/PS-4068 de Lima Salgado, A., & de Mattos Fortes, R. P. (2016). Heuristic evaluation for novice evaluators. In A. Marcus (Ed.), *International conference of design, user experience, and usability* (pp. 387–398). Springer.

Maguire, M. (2001). Methods to support human-centered design. *International Journal of Human-Computer Studies*, *55*(4), 587–634.

Mandel, T. (1997). The golden rules of user interface design. In *The elements of user interface design* (pp. 1–28). John Wiley & Sons.

Maulsby, D., Greenberg, S., & Mander, R. (1993). Prototyping an intelligent agent through Wizard of Oz. *Proceedings of the INTERACT'93 and CHI'93 conference on human factors in computing systems* (pp. 277–284). ACM Press.

Mi, N., Cavuoto, L. A., Benson, K., Smith-Jackson, T., & Nussbaum, M. A. (2014). A heuristic checklist for an accessible smartphone interface design. *Universal Access in the Information Society*, *13*(4), 351–365.

Miniwatts Marketing Group. (n.d.). *Internet world stats: Usage and population statistics*. Retrieved September 25, 2016, from https://www.internetworldstats.com/

Molich, R., & Nielsen, J. (1990). Improving a human-computer dialogue. *Communications of the ACM*, *33*(3), 338–348.

Moraveji, N., & Soesanto, C. (2012). Towards stress-less user interfaces: 10 design heuristics based on the psychophysiology of stress. *CHI'12* *extended abstracts on human factors in computing systems* (pp. 1643–1648). ACM Press.

Muller, M. J., Matheson, L., Page, C., & Gallup, R. (1998). Methods and tools: Participatory heuristic evaluation. *Interactions*, *5*(5), 13–18. https://doi.org/10.1145/285213.285219

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 373–380). ACM Press. https://doi.org/10.1145/142750.142834

Nielsen, J. (1992). Reliability of severity estimates for usability problems found by heuristic evaluation. *Posters and short talks of the 1992 SIGCHI conference on human factors in computing systems*. ACM Press. https://doi.org/10.1145/1125021.1125117

Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. *Conference companion on human factors in computing systems*. ACM Press. https://doi.org/10.1145/259963.260333

Nielsen, J. (1994). Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier. In R. G. Bias & D. J. Mayhew (Eds.), *Cost-justifying usability*. Academic Press.

Nielsen, J. (1994). Usability inspection methods. *Conference companion on human factors in computing systems* (pp. 413–414). ACM Press. https://doi.org/10.1145/259963.260531

Nielsen, J. (2012, January 3). *Usability 101: Introduction to usability*. Nielsen Norman Group. Retrieved September 25, 2020, from https://www.nngroup.com/articles/usability-101-introduction-to-usability/

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Proceedings of the SIGCHI Conference on human factors in computing systems empowering people*. ACM Press. https://doi.org/10.1145/97243.97281

Norman, D. A. (1988). *The psychology of everyday things*. Basic Books.

Norman, D. A. (2013). *The design of everyday things: revised and expanded edition*. Basic Books.

Paz, F., & Paz, F. A. (2015). Experimental case study of new usability heuristics. In A. Marcus (Ed.), *International conference of design, user experience, and usability* (p. 212–223). Springer.

Peterson, M. R. (2007, June). UX increases revenue: Two case studies. *User Experience Professional Association*. http://uxpamagazine.org/ux_increases_revenue/

Petrie, H., & Buykx, L. (2010). Collaborative heuristic evaluation: Improving the effectiveness of heuristic evaluation. *2010 ICEEE'10 and the ICME'10: World Congress on Engineering*. IAENG.

Po, S., Howard, S., Vetere, F., & Skov, M. B. (2004). Heuristic evaluation and mobile usability: Bridging the realism gap. *Proceedings of the 10th annual* *international conference on mobile computing and networking* (pp. 49–60). Association for Computing Machinery.

Preece, J., & Rombach, H. D. (1994). A taxonomy for combining software engineering and human-computer interaction measurement approaches: Towards a common framework. *International Journal of Human-Computer Studies*, *41*(4), 553–583. https://doi.org/10.1006/ijhc.1994.1073

Rajanen, D., Clemmensen, T., Iivari, N., Inal, Y., Rızvanoğlu, K., Sivaji, A., & Roche, A. (2017). UX professionals' definitions of usability and UX: A comparison between Turkey, Finland, Denmark, France and Malaysia. In R. Bernhaupt, G. Dalvi, A. Joshi, D. K. Balkrishan, J. O'Neill, & M. Winckler (Eds.), *Human-computer interaction: INTERACT 2017* (pp. 218–239). Springer. https://doi.org/10.1007/978-3-319-68059-0_14

Al-Razgan, M. S., Al-Khalifa, H. S., & Al-Shahrani, H. H. (2012). Touch-based mobile phone interface guidelines and design recommendations for elderly people: A survey of the literature. In J. Shawe-Taylor (Ed.), *Advances in neural information processing systems: 25th annual conference on neural information processing systems*. Curran.

Al-Razgan, M. S., Al-Khalifa, H. S., & Al-Shahrani, H. H. (2014). Heuristics for evaluating the usability of mobile launchers for elderly people. *International conference of design, user experience, and usability* (pp. 415–424). Springer.

Renzi, A. B., Chammas, A., Agner, L., & Greenshpan, J. (2015) Startup Rio: User experience and startups. In A. Marcus (Ed.), *Design, user experience, and usability: Design discourse.* Springer. https://doi.org/10.1007/978-3-319-20886-2_32

Robinson, D., & Fitter, M. (1992). Supportive evaluation methodology: A method to facilitate system development. *Behaviour & Information Technology*, *11*(3), 151–159. https://doi.org/10.1080/01449299208924332

Robson, C., & McCartan, K. (2016). *Real world research: A resource for users of social research methods in applied settings* (4th ed.). Wiley.

Rosenbaum, S., Rohn, J. A., & Humburg, J. (2000). A toolkit for strategic usability: Results from workshops, panels, and surveys. *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 337–344. ACM Press. https://doi.org/10.1145/332040.332454

Rusu, C., Botella, F., Rusu, V., Roncagliolo, S., & Quiñones, D. (2018) An online travel agency comparative study: Heuristic evaluators perception. In G. Meiselwitz (Ed.), *Social computing and social media: User experience and behavior*. Springer. https://doi.org/10.1007/978-3-319-91521-0_9

Saavedra, M. J., Rusu, C., Quiñones, D., & Roncagliolo, S. (2019) A set of usability and user experience heuristics for social networks. In G. Meiselwitz (Ed.), *Social computing and social media: Design, human behavior and analytics*. Springer. https://doi.org/10.1007/978-3-030-21902-4_10

Salian, K., & Sim, G. (2014). Simplifying heuristic evaluation for older children. *Proceedings of the India HCI 2014 conference on human computer interaction*. ACM Press. https://doi.org/10.1145/2676702.2676704

Shneiderman, B. (1987). Designing the user interface strategies for effective human-computer interaction. *ACM SIGBIO Newsletter*, *9*(1). https://doi.org/10.1145/25065.950626

Sivaji, A., Abdullah, M. R., Downe, A. G., & Ahmad, W. F. (2013). Hybrid usability methodology: Integrating heuristic evaluation with laboratory testing across the software development lifecycle. *International conference on information technology: New generations* (pp. 375–383). IEEE Computer Society.

Slavkovic, A., & Cross, K. (1999). Novice heuristic evaluations of a complex interface. *CHI '99 extended abstracts on human factors in computing systems*. ACM Press. https://doi.org/10.1145/632716.632902

Sohl, M. (2017). *Comparing two heuristic evaluation methods and validating with usability test methods* [Unpublished master's thesis]. Linköping University. https://liu.diva-portal.org/smash/get/diva2:1241260/FULLTEXT01.pdf

Sova, D. H., & Nielsen, J. (2010). *234 tips and tricks for recruiting users as participants in usability studies*. NN Group. https://media.nngroup.com/media/reports/free/How_To_Recruit_Participants_for_Usability_Studies.pdf

Sripathi, V., & Sandru, V. (2013). Effective usability testing: Knowledge of user centered design is a key requirement. *International Journal of Emerging Technology and Advanced Engineering*, *3*(1), 627–635. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.7315&rep=rep1&type=pdf

Tan, W.-S., Liu, D., & Bishu, R. (2009). Web evaluation: Heuristic evaluation vs. user testing. *International Journal of Industrial Ergonomics*, *39*(4), 621–627. https://doi.org/10.1016/j.ergon.2008.02.012

Thomas, R. P., & Lawrence, A. (2018). Assessment of expert performance compared across professional domains. *Journal of Applied Research in Memory and Cognition*, *7*(2), 167–176. https://doi.org/10.1016/j.jarmac.2018.03.009

Tognazzini, B. (2003). *First principles of interaction design (revised and expanded).* Ask Tog. https://asktog.com/atc/principles-of-interaction-design/

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

University of Washington Computing & Communications. (2005). *Communications, graphics and web design based on Edward Tufte's principles*. Retrieved September 25, 2020, from http://chnm.gmu.edu/digitalhistory/links/pdf/chapter4/4.13b%20and%204.21.pdf

Usability First (2007). *Usability ROI case study: Breastcancer.org discussion forums*. http://www.usabilityfirst.com/documents/U1st_BCO_CaseStudy.pdf

Varsaluoma, J. (2009). Scenarios in the heuristic evaluation of mobile devices: Emphasizing the context of use. *2009 4th ACM/IEEE proceedings*. IEEE.

Virzi, R. A. (1992). Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *34*(4), 457–468. https://doi.org/10.1177/001872089203400407

Wodike, O. A., Sim, G., & Horton, M. (2014). Empowering teenagers to perform a heuristic evaluation of a game. *Proceedings of the 28th international BCS human-computer interaction conference* (pp. 353–358). https://doi.org/10.14236/ewic/HCI2014.57

Zhang, D., & Adipat, B. (2005). Challenges, methodologies, and issues in the usability testing of mobile applications. *International Journal of Human-Computer Interaction*, *18*(3), 293–308. https://doi.org/10.1207/s15327590ijhc1803_3

Zhang, Z., Basili, V., & Shneiderman, B. (1999). [No Title Found]. *Empirical Software Engineering*, *4*(1), 43–69. https://doi.org/10.1023/A:1009803214692

APPENDIX A Interview questions

My name is Anas Abulfaraj. Thank you for talking with me today.

We are doing research with the aim of making heuristic evaluation easier to apply by novice evaluators.

We have an outline of information we would like to learn about from you. But we would like for you to think about this more as a conversation than a question-and-answer session. We'll start out talking about some demographics. Then we will want to learn about your previous experiences with heuristic evaluation in general. After that, we'll ask some questions about Nielsen's heuristics. Finally, we will ask about your thoughts on how to improve heuristic evaluation for novice evaluators. Does that sound OK to you?

There are no right or wrong answers to my questions; I just want to listen to what you have to say.

We will be recording this session for the purposes of the study explained earlier. Your responses will be confidential and only used to help us in our research about facilitating heuristic evaluation for novice evaluators.

This interview is voluntary. You have the right not to answer any question, and to stop the interview at any time or for any reason. Any questions before I begin?

Screening (Asked before, but repeated to make sure)

- At least 4 years as a UX professional?

- Conducted at least three heuristic evaluations?
- Familiar with Nielsen's heuristics?

Prep

Tell us about yourself

- Age? (Are you over 21?)
- Gender? (Male, female, or other?)
- Years of experience in the field?
- Your job title?
- Occupation? (Industry vs. Academia).
- What do you usually work on?

Heuristic evaluation

- Tell us about your experience with heuristic evaluation.
- Based on your experience in the field, which set of heuristics are the most popular?
- Which heuristics have you used? And which do you prefer?
- How do you compare them? (In terms of time, ease of use, effectiveness).
- What is your process in conducting heuristic evaluation?
    - One-set? Multiple sets?
    - On average, how long does the whole evaluation take?
    - Do you divide the application page by page, tasks, scenarios, aspects?
    - How do you document the usability problems?

    - How do you map the usability problems to the heuristics?
    - Could you walk us through your process?
- How do you rate the severity of the issue?
    - Which scale do you use?
    - Could you walk us through your process?
- Could you think of any exercises or practices that might help you empathize with the user and sharpen your mind to find critical issues while conducting the evaluation?
- Do you have any suggestions on how to improve the understandability of the usability guidelines for novice evaluators?
- Do you have any suggestions on how to facilitate the use of heuristic evaluation for novice evaluators?

Nielsen's heuristics

- How would you describe Nielsen's heuristics?
- Have you encountered any difficulties using Nielsen's heuristics?
    - If yes, what are the difficulties? And how do you overcome them?
    - Is there any overlap between certain guidelines?
- If you were to divide the guidelines into sub guidelines, which guidelines would you divide?
- Are all the labels easy to understand?
    - If no, could you rename those unclear labels?
- Let's go through each guideline and see if the description of the guideline addresses the label clearly and completely?

    - How would you describe each one of the heuristics?
- For each guideline, can you think of any examples other than the ones that are mentioned in the description of the guideline and can clarify the meaning of the guideline?
- For each guideline, and based on your experience, what are the most common usability problems that the guideline help in finding?
- Based on your experience, are all the guidelines applicable in all kinds of systems and situations?
    - If no, in which systems and situations the guideline would be inapplicable?
    - Could you give some examples from your experience?
- For each guideline, can you explain its importance?
    - What are the consequences of ignoring it?

Wrap-up questions

- Finally, Is there anything else you want to add?

Thank you again for taking the time to speak with me today.

APPENDIX B: Recruitment script

My name is Anas Abulfaraj. I am a PhD student in the College of Computing and Digital Media at DePaul University, USA. We are conducting a research study about Heuristic Evaluation (HE).

We want to learn more about the tactics and techniques expert evaluators use when they conduct heuristic evaluation. Moreover, we want to know the challenges they face during the evaluation and how they overcome them. The goal is to present this information to novice evaluators, so they can apply them when they conduct heuristic evaluation.

As part of the research, we want to interview usability experts who have at least 4 years of experience in the field, have done at least three HE sessions, and familiar with Nielsen’s Heuristics.

If you meet the aforementioned criteria, and agreed to participate in this research, you will be asked to complete an audio recorded interview. The interview will include questions about your experience with HE, Nielsen’s heuristics, and your opinions on how to improve them. The interview will take about 90–120 minutes of your time.

I will be the main researcher. My email is aabulfa2@mail.depaul.edu and my phone number is 812-369-5879.

My faculty advisor is Adam Steele, PhD. His email is asteele@cs.depaul.edu and his phone number is 312-362-6247.

If you know anyone else who might be interested in being in the research and who meets the criteria, please forward this email to them. When we talk to you, we might ask if you know of anyone else you can refer to us.

APPENDIX C IRB approval

DEPAUL
UNIVERSITY

Office of Research Services
Institutional Review Board
1 East Jackson Boulevard
Chicago, Illinois 60604-2287
312-362-7593
Fax: 312-362-7574

Research Involving Human Subjects

NOTICE OF INSTITUTIONAL REVIEW BOARD ACTION

To: Anas Abulfaraj, Graduate Student, College of Computing and Digital Media

Date: May 7, 2019

Re: Research Protocol # AA041819CDM
"Facilitating Heuristic Evaluation (HE) for Novice Evaluators"

Please review the following important information about the review of your proposed research activity.

Review Details
This submission is an initial submission.

Your research project meets the criteria for Exempt review under 45 CFR 46.104 under the following category:

(2) Research that only includes interactions involving educational tests (cognitive, diagnostic, aptitude, achievement), survey procedures, interview procedures or observation of public behavior (including visual or auditory recording) if at least one of the following criteria is met:

(i) *The information obtained is recorded by the investigator in such a manner that the identity of the human subjects cannot readily be ascertained, directly or through identifiers linked to the subjects;*

(ii) ***Any disclosure of the human subjects' responses outside the research would not reasonably place the subjects at risk of criminal or civil liability or be damaging to the subjects' financial standing, employability, educational advancement, or reputation;*** *or*

(iii) *The information obtained is recorded by the investigator in such a manner that the identity of the human subjects can readily be ascertained, directly or through identifiers linked to the subjects, and an IRB conducts a limited IRB review to make the determination required by 45 CFR 46.111 (a)(7). (REQUIRES LIMITED IRB REVIEW OF PRIVACY AND CONFIDENTIALITY)*

Approval Details
Your research was reviewed and approved on May 7, 2019.

Number of approved participants: 20 Total
You should not exceed this total number of subjects without prospectively submitting an amendment to the IRB requesting an increase in subject number.

Funding Source: 1) None

Approved Performance sites: 1) DePaul University

DePaul IORG#0000628, FWA#00000099, IRB Registration#00000964 Page **1** of **2**

Reminders

- Under DePaul's current institutional policy governing human research, research projects that meet the criteria for an exemption determination may receive administrative review by the Office of Research Services Research Protections staff. Once projects are determined to be exempt, the researcher is free to begin the work and is not required to submit an annual update (continuing review). As your project has been determined to be exempt, your primary obligation moving forward is to resubmit your research materials for review and classification/approval when making changes to the research, but before the changes are implemented in the research. **All changes to the research must be reviewed and approved by the IRB or Office of Research Services staff.** Changes requiring approval include, but are not limited to, changes in the design or focus of the research project, revisions to the information sheet for participants, addition of new measures or instruments, increasing the subject number, and any change to the research that might alter the exemption status (either add additional exemption categories or make the research no longer eligible for an exemption determination).

- **Once the project is complete, you should submit a final closure report to the IRB.**

The Office of Research Services would like to thank you for your efforts and cooperation and wishes you the best of luck on your research. If you have any questions, please contact me by telephone at (312) 362-7593 or by email at sloesspe@depaul.edu.

For the Board,

Susan M. Loess-Perez

Susan Loess-Perez, MS, CIP, CCRC
Director of Research Compliance
Office of Research Services

Cc: Adam Steele, PhD, Faculty Sponsor, College of Computing and Digital Media

APPENDIX D Information sheet for participation in research study

Interview: Facilitating Heuristic Evaluation (HE) for Novice Evaluators

Principal Investigator: Anas Abulfaraj, PhD Student, College of Computing and Digital Media

Institution: DePaul University, USA

Faculty Advisor: Adam Steele, PhD, College of Computing and Digital Media.

We are conducting a research study because we are trying to learn more about the tactics and techniques experts use when they conduct heuristic evaluation. Moreover, we want to know the challenges they face during the evaluation and how they overcome them. The goal is to present this information to novice evaluators, so they can apply them when they conduct heuristic evaluation.

We are asking you to be in the research because you have at least 4 years of experience in Human-Computer Interaction (HCI), have conducted at least three sessions of heuristic evaluation and are familiar with Nielsen's heuristics.

If you agree to be in this study, you will be asked to complete an interview. The interview will include questions about your experience with Heuristic Evaluation (HE), Nielsen's heuristics, and your opinions on how to improve them. We will also collect some personal information about you such as age, gender, education, and occupation. If there is a question you do not want to answer, you may skip it. The interview will be audio recorded. We are recording this interview in order to make accurate written notes of what you have said. Once the interview is transcribed, the recording will be deleted. This interview will take about 90–120 minutes of your time.

Research data collected from you will be kept confidential. Voice recordings are considered identifiable, but we will not put your name on the recording file or use it during the recorded interview. The recording file will be labeled with a random number, such as participant 1, participant 2, etc., and then when we transcribe the recording into written notes, there will be no identifiers present on the transcripts.

Your participation is voluntary, which means you can choose not to participate. There will be no negative consequences if you decide not to participate or change your mind later after you begin the study. You can withdraw your participation at any time, by letting me know before the interview ends or shortly after the interview is completed. Once I transcribe the interview and delete the recordings, I will not know which transcript belongs to you.

If you have questions, concerns, or complaints about this study or you want to get additional information or provide input about this research, please contact me at aabulfa2@mail.depaul.edu.

If you have questions about your rights as a research subject, you may contact Susan Loess-Perez, DePaul University's Director of Research Compliance, in the Office of Research Services at 312-362-7593 or by email at sloesspe@depaul.edu. You may also contact DePaul's Office of Research Services if:

- Your questions, concerns, or complaints are not being answered by the research team.
- You cannot reach the research team.
- You want to talk to someone besides the research team.

You may keep [or print] this information for your records.

I have explained the study to you, and by signing the document below, you are indicating your affirmative agreement to be in the research.

Signature:

____

Printed Name:

____

Date:

____

APPENDIX E IRB approval

DEPAUL
UNIVERSITY

Office of Research Services
1 East Jackson Boulevard
Chicago, Illinois 60604-2287

Research Involving Human Subjects
NOTICE OF INSTITUTIONAL REVIEW BOARD ACTION

To: Anas Abulfaraj, CDM

Date: April 09, 2021

Re: Research Protocol # IRB-2021-283

Title: Facilitating Heuristic Evaluation for Novice Evaluators

Please review the following important information about the review of your proposed research activity.

Review Details
This submission is an initial submission. Your research project meets the criteria for Exempt review under 45 CFR 46.104.

Approval Details

Approval date: April 09, 2021

Approved Study Documents: See the attachments tab in the protocol application online.

Number of Approved Subjects: See the approved protocol application online.

You should not exceed the total number of subjects without prospectively submitting an amendment to the IRB requesting an increase in subject number.

Findings:1. Exempt Category 3 i C - (3) (i) Research involving benign behavioral interventions in conjunction with the collection of information from an adult subject through verbal or written responses (including data entry) or audiovisual recording if the subject prospectively agrees to the intervention and information collection and at least one of the following criteria is met: (C) The information obtained is recorded by the investigator in such a manner that the identity of the human subjects can readily be ascertained, directly or through identifiers linked to the subjects, and an IRB conducts a limited IRB review to make the determination required by 45 CFR 46.111(a)(7). (REQUIRES LIMITED IRB REVIEW OF PRIVACY AND CONFIDENTIALITY) (ii) For the purposes of this provision, benign behavioral interventions are brief in duration, harmless, painless, not physically invasive, not likely to have a significant adverse lasting impact on the subjects, and the investigator has no reason to think the subjects will find the interventions offensive or embarrassing. Provided all such criteria are met, examples of such benign interventions would include having the subjects play an online game, having them solve puzzles under various noise conditions or having them decide how to allocate a nominal amount of received cash between themselves and someone else. (iii) If the research involves deceiving the subjects regarding the nature or purposes of the research, this exemption is not applicable unless the subject authorizes the deception through a prospective agreement to participate in the research in circumstances in which the subject is informed that he or she will be unaware of or misled regarding the nature or purposes of the research.
2. Exemption-Limited IRB Review - A limited IRB review of the privacy and confidentiality protections in place, and where applicable the Broad Consent for categories 7 and 8, has been completed and the privacy and confidentiality protections in place have been determined to be appropriate.

Includes: Performance Site: University of del Cauca in Colombia, Cesar Collazos, PhD.

Reminders
* Under DePaul's current institutional policy governing human research, research projects that meet the criteria for an exemption determination may receive administrative review by the Office of Research Services Research Protections staff. Once projects are determined to be exempt, the researcher is free to begin the work and is not required to submit an annual update (continuing review). As your project has been determined to be exempt, your primary obligation moving forward is to resubmit your research materials for review and classification/approval when making changes to the research, but before the changes are implemented in the research. **All changes to the research must be reviewed and approved by the IRB or Office of Research Services staff**. Changes requiring approval include, but are not limited to, changes in the design or focus of the research project, revisions to the information sheet for participants, addition of new measures or instruments, increasing the subject number, and any change to the research that might alter the exemption status (either add additional exemption categories or make the research no longer eligible for an exemption determination).

*** Once the project is complete, you should submit a final closure report to the IRB.**

The Board would like to thank you for your efforts and cooperation and wishes you the best of luck on your research. If you have any questions, please contact Jessica Bloom, Assistant Director of Research Compliance by telephone at (312) 362-6168 or by email at jbloom8@depaul.edu.

For the Board,

DePaul U IORG#: 0000628 FWA#: 00000099 IRB Registration#: 00000964

Jessica Bloom

APPENDIX F Recruitment material (Novices)

Recruitment Script: Facilitating Heuristic Evaluation (HE) for Novice Evaluators

My name is Anas Abulfaraj. I am a PhD student in the College of Computing and Digital Media at DePaul university, USA. We are conducting a research study about Heuristic Evaluation (HE) which is a method used to evaluate the usability of any given system. We want to learn more about the differences between a new way to conduct Heuristic Evaluation called Coherent Heuristic Evaluation (CoHE) and a traditional Heuristic Evaluation. We want to know how they quantitively and qualitatively differ when used by novices. We want to know which one of them is easier to understand, easier to implement and help in finding more usability issues. The ultimate goal of this research effort is to facilitate the use of Heuristic Evaluation and make it more accessible for novice evaluators.

As part of the research, we want HCI/UX students and other related fields to read the heuristics and conduct a Heuristic Evaluation on a system that we chose. The students we want to recruit should be HCI/UX, Computer Science, Information Technology, Information Systems, Software Engineering, or Computer Engineering students, have less than 4 years of experience in the field of HCI/UX and never done Heuristic Evaluation before outside of a class settings.

If you meet the aforementioned criteria, and agreed to participate in this research, you will be asked to read a usability heuristic and evaluate a system. The reading and the evaluation will be done in separate sessions. Both sessions will be done via a conference video call with the presence of the main researcher. Moreover, both sessions will be video recorded. In the reading session, you will be given a link to a list of usability heuristics and you will be asked to read them. After that you will be asked some questions regarding the understandability of the heuristics. In the evaluation session, you will be given a link to the system along with a number of tasks. You will be asked to go through the tasks and evaluate the system along the way. After that you will be asked some questions regarding your experience with the method. Each session will take about 90-120 minutes of your time for a potential total of up to four hours for both sessions.

I will be the main researcher. My email is aabulfa2@mail.depaul.edu and my phone number is 812-369-5879.

My faculty advisor is Adam Steele, PhD. His email is asteele@cs.depaul.edu and his phone number is 312-362-6247.

APPENDIX G Recruitment material (Experts)

Recruitment Script: Facilitating Heuristic Evaluation (HE) for Novice Evaluators

My name is Anas Abulfaraj. I am a PhD student in the College of Computing and Digital Media at DePaul university, USA. We are conducting a research study about Heuristic Evaluation (HE). We want to learn more about the differences between a new way to conduct Heuristic Evaluation called Coherent Heuristic Evaluation (CoHE) and a traditional Heuristic Evaluation. We want to know how they quantitively and qualitatively differ when used by novices. However, novices tend to find false positive usability issues when evaluating systems. Therefore, we want usability experts to participate by evaluating a system so we can use their results as a benchmark to assess the usability issues found by novices. The ultimate goal of this research effort is to facilitate the use of Heuristic Evaluation and make it more accessible for novice evaluators.

As part of the research, we want usability experts to conduct a Heuristic Evaluation on a system that we chose. The usability experts we want to recruit should have at least four years of experience in the field and have done at least three Heuristic Evaluation sessions before.

If you meet the aforementioned criteria, and agreed to participate in this research, you will be asked to evaluate a system. The evaluation session will be done via a conference video call with the presence of the main researcher. Moreover, the evaluation session will be video recorded. In the session, you will be given a link to the system along with a number of tasks. You will be asked to go through the tasks and evaluate the system along the way. The evaluation session will take about 90-120 minutes of your time.

I will be the main researcher. My email is aabulfa2@mail.depaul.edu and my phone number is 812-369-5879.

My faculty advisor is Adam Steele, PhD. His email is asteele@cs.depaul.edu and his phone number is 312-362-6247.

If you know anyone else who might be interested in being in the research and who meets the criteria, please forward this email to them.

APPENDIX H 24 hours reminder

24 Hours Reminder: Facilitating Heuristic Evaluation (HE) for Novice Evaluators

Heuristic Evaluation (HE) research study.

Greetings! Thanks for taking part in our research study.

This is a friendly reminder that our meeting will take place tomorrow. Here is our meeting's information:

Date: ------------

Time: ------------

Place: zoom link

Duration: 2-hours

Again, thank you for your participation.

Anas Abulfaraj.

APPENDIX I Novices information sheet

Information Sheet for Participation in Research Study

Facilitating Heuristic Evaluation (HE) for Novice Evaluators

Principal Investigator: Anas Abulfaraj, PhD Student, College of Computing and Digital Media.

Institution: DePaul University, USA.

Faculty Advisor: Adam Steele, PhD, College of Computing and Digital Media.

We are conducting a research study about Heuristic Evaluation (HE) which is a method used to evaluate the usability of any given system. We want to learn more about the differences between a new way to conduct Heuristic Evaluation called Coherent Heuristic Evaluation (CoHE) and a traditional Heuristic Evaluation. We want to know how they quantitively and qualitatively differ when used by novices. We want to know which one of them is easier to understand, easier to implement and help in finding more usability issues. The ultimate goal of this research effort is to facilitate the use of Heuristic Evaluation and make it more accessible for novice evaluators.

We are asking you to be in the research, because you are a HCI/UX, Computer Science, Information Technology, Information Systems, Software Engineering, or Computer Engineering student, have less than 4 years of experience in the field of HCI/UX and never done Heuristic Evaluation before outside of a class settings.

If you agree to be in this study, you will be asked to read usability heuristics and evaluate a system. The reading and the evaluation will be done in two separate sessions. In the reading session, you will be given a link to a list of usability heuristics and you will be asked to read them. After that you will be asked some questions regarding the understandability of the heuristics. In the evaluation session, you will be given a link to the system along with a number of tasks. You will be asked to go through the tasks and evaluate the system along the way. After that you will be asked some questions regarding your experience with the method. We will also collect some personal information about you such as age, gender, education, and occupation. If there is a question you do not want to answer, you may skip it. The sessions will be video recorded. We are recording this session in order to make accurate written notes of what you have said and done. Once the sessions are transcribed, the videos will be deleted. Each of the two sessions will take about 90-120 minutes of your time, for a total of up to 4 hours.

Research data collected from you will be kept confidential. Video recordings are considered identifiable, but we will not put your name on the recording file or use it during the recorded sessions. The recording files will be labeled with a random number, such as participant 1, participant 2, etc. and then when we transcribe the recordings into written notes, there will be no identifiers present on the transcripts.

Your participation is voluntary, which means you can choose not to participate. There will be no negative consequences if you decide not to participate or change your mind later after you begin the study. You can withdraw your participation at any time, by letting me know before the sessions ends or shortly after the sessions are completed. Once I transcribe the sessions and delete the recordings, I will not know which transcript belongs to you.

If you have questions, concerns, or complaints about this study or you want to get additional information or provide input about this research, please contact me at aabulfa2@mail.depaul.edu.

If you have questions about your rights as a research subject, you may contact Jessica Bloom in the Office of Research Services at (312) 362-6168 or via email at jbloom8@depaul.edu. You may also contact DePaul's Office of Research Services if:

1. Your questions, concerns, or complaints are not being answered by the research team.
2. You cannot reach the research team.
3. You want to talk to someone besides the research team.

You may keep [or print] this information for your records.

I have explained the study to you, and by participating in the interview session, you are indicating your affirmative agreement to be in the research.

APPENDIX J Nielsen's 10 usability heuristics

List of Nielsen's 10 Usability Heuristics

1. Visibility of system status:

    Description: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

2. Match between system and the real world:

    Description: The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

3. User control and freedom:

    Description: Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. Consistency and standards:

    Description: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5. Error prevention:

    Description: Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6. Recognition rather than recall:

Description: Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. Flexibility and efficiency of use:

Description: Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. Aesthetic and minimalist design:

Description: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. Help users recognize, diagnose, and recover from errors:

Description: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. Help and documentation:

Description: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

APPENDIX K Detailed usability heuristics

List of 10 Usability Heuristics

- Visibility of system status:

There are four concepts under it:

1. State: the user should know the state of the system and what he or she is capable of doing in the system at any given time.
    a. How to make it work?
        i. There should be a visible and easy to interpret indicator to inform the user about the state of the system.
        ii. Any changes in the state of the system should be reflected immediately.
    b. Why does it matter?
        i. Lets the user know what to expect and what to do.
        ii. This reduces the time, since the user needn't spend time to figure out the system state.
        iii. A lack of this quality will give users false hope and possibly lead to errors.
    c. Examples:
        i. If there is a link in the page, it should appear in a different color and be underscored so that the user knows that he/she can click on it.
        ii. If in an e-shop, a certain item was available, and then suddenly went out of stock, the status of the item should immediately change to "out of stock".
2. Location: the user should know which part of the system they are in, i.e., his or her location in relation to other parts of the system.
    a. How to make it work?
        i. There should be an indicator to tell the user where he or she is in relation to other parts of the system.
        ii. The indicator should be clear and visible.
        iii. The indicator should be present throughout the whole system.
    b. Why does it matter?
        i. Knowing where they are, would help them know where to go.
        ii. The steps a user takes to accomplish any goal is mostly dependent upon where he or she is in the system.
        iii. When the user leaves the system for a while and returns, they might not remember where they were before leaving the system.

iv. The logo of the system tells a user where he or she is and also conveys the system purpose and quality.

c. Exception:

i. There are some cases where it is relatively unimportant for a user to know where he or she is located on the system, most often when the system consists of a very small number of pages.

d. Examples:

i. The logo of the system on the top of the page lets users know in which system they are.

ii. The title of the page lets them know in which part of the system they are.

iii. The navigation bar lets them know where they are in relation to other parts of the system.

3. <u>Progress: a user should know how far he or she is from accomplishing their goal.</u>

a. How to make it work?

i. The indicator should be appropriate (time, capacity, steps).

ii. The indicator should be accurate.

iii. The indicator should be present both in active and passive situations.

b. Why does it matter?

i. Makes the user feel in control.

ii. Helps the user make decisions about how to best use their time.

c. Exception:

i. The only exception to showing progress is when the action takes little time to be accomplished. If it takes only a few seconds, generally less than 10, it is relatively unimportant to show progress.

d. Examples:

i. Active situations occur when the user is completing a multi-step task. For example, when a user is completing a multipage form, he/she should know how many pages have been completed and how many pages are left.

ii. Passive situations are when a user takes an action and waits for the system to complete it. For example, when the user downloads a file, the system shows him/her how long the file will take to download.

4. <u>Closure: the user should know that the task at hand is completed, whether or not it has been completed with desirable outcomes.</u>

a. How to make it work?

i. It should be offered immediately after completion.

ii. It should be offered no matter what the results are.
iii. It should be clear and understandable.
iv. If the action wasn't successful, an explanation should be offered.

b. Why does it matter?
   i. Not knowing whether or not the goal is completed defeats the purpose of the system.
   ii. If the user doesn't know whether or not a goal is completed, he or she might repeat the action.
c. Examples:
   i. When the user performs a financial transaction, he/she should know whether it went through.
   ii. When the user fills in a form and then submits it, the system should tell the user that the form was submitted.

- <u>Match between system and the real world:</u>

There are three concepts under it:

1. <u>Understandability: the content presented on the system should be understandable by the target audience.</u>
   a. How to make it work?
      i. The content should be understandable by the target audience, target audience being the key word.
   b. Why does it matter?
      i. Not understanding the content makes learning how to use it difficult.
      ii. Mistakes can occur if content is not understood.
   c. Exceptions:
      i. It should be emphasized that understandability depends heavily on the target audience. Something considered understandable to one group of users might not be considered understandable to another.
   d. Examples:
      i. Content refers not only to text but generally anything presented in the system, such as pictures, icons, or metaphors. All of these should be understandable to the target audience.
2. <u>Natural and logical order: the content and actions presented on the system should follow a logical/natural order.</u>
   a. How to make it work?
      i. The action sequence should follow the same order as similar tasks in the real world.

    ii. If not possible, it should be logical and intuitive.
  b. Why does it matter?
    i. Following natural/logical order makes learning faster.
    ii. Not following the natural/logical order could lead to errors.
  c. Exceptions:
    i. Following natural order could be challenging if the system operates unconventionally or in a completely new way. In such a case, while it is possible the natural order should not be followed, effort should be made to make the order as logical and intuitive as possible.
    ii. In some cases, the system could suggest a new way to do certain tasks in an order that may not seem natural, but it should still be logical and easy to use.
  d. Examples:
    i. In an e-shop, the steps to buy an item should be similar to the steps that users follow when they buy an item from a physical shop.

3. <u>Appropriateness: the content presented on the system should be acceptable and appropriate to the target audience.</u>
  a. How to make it work?
    i. The content should match the system identity and purpose.
    ii. The content should be appropriate and not offensive.
  b. Why does it matter?
    i. Using inappropriate content could offend the users.
  c. Exceptions:
    i. The appropriateness of the content also depends on the target audience and the nature of the system.
  d. Examples:
    i. If the system is expected to be used by children, then certain words or phrases should not be used.
    ii. If the system is expected to be used by users from a certain culture, then content that might be perceived as offensive should not be displayed.
    iii. If the system is formal, like government websites, then the language used should be formal as well.

- <u>User control and freedom:</u>

There are three concepts under it:

1. <u>Reversibility: the user should be able to undo and redo any action performed on the system.</u>

   a. How to make it work?
      i. The user should be able to reverse any action at any time and any place.
      ii. Reversing actions should be easy.
      iii. Users should be able to go back as far as they want in reversing the action.
   b. Why does it matter?
      i. Reversibility capability facilitates the learning process.
      ii. It helps in effectively handling errors.
      iii. It helps reduce the user's stress and anxiety.
      iv. Inability to reverse actions could be risky and embarrassing.
   c. Exceptions:
      i. Sometimes the system intentionally doesn't allow the user to reverse some actions. An example would be a survey in which the questions should be answered in a specific order.
      ii. sometimes a user should be given only a limited time to reverse an action, after which they can't.
   d. Examples:
      i. If the user deletes a certain file, he/she should be able to retrieve the deleted file if desired.

2. <u>Emergency exit: the user should be able to escape from any undesirable situation in the system.</u>
   a. How to make it work?
      i. The user should be able escape any situation regardless of time and place.
      ii. This should be easy to do.
   b. Why does it matter?
      i. A user might have done something and changed his or her mind or made a mistake, so escape from such a situation should be possible.
      ii. The user should feel that he or she is in control, not the system.
      iii. Lack of an emergency exit could cause privacy issues.
   c. Examples:
      i. On certain websites, there are continuous pop ups that the user does not know how to block, which is a violation of this heuristic.
3. <u>Informing users: before asking the user to enter any input or take any action; the user should be presented with enough information to let him or her make an informed decision.</u>
   a. How to make it work?
      i. Why information is asked should be explained.

    ii. How information is going to be stored and handled should be explained.
  b. Why does it matter?
    i. If the user doesn't know why they are asked to provide certain information, they might be hesitant to use the system.
    ii. There is also an ethical obligation if the information is sensitive.
  c. Exceptions:
    i. Only sensitive and personal information needs to be given a reason.
    ii. if the reasons for asking for personal/sensitive information is obvious, it might not be necessary to explain why.
  d. Examples:
    i. When the system asks the user to enter personal information, the system should explain to the user why he/she is being asked to enter this information and how the system is going to handle this information.
    ii. When an application asks to get permission to the user's contact list, it should explain why.

- <u>Consistency and standards:</u>

  There are two concepts under it:

1. <u>Consistency: once a certain element of the system is used in one place of the system, it should be presented in the same way throughout the whole system.</u>
  a. How to make it work?
    i. Meanings should be consistent.
    ii. Functions should be consistent.
    iii. Organization and layout should be consistent.
    iv. The effort should be consistent.
    v. The feeling should be consistent.
  b. Why does it matter?
    i. This makes learning the system easier.
    ii. It also increases the memorability of the system.
    iii. Lack of consistency could lead to errors.
  c. Exceptions:
    i. If certain things should stand out, this should be accomplished in a way inconsistent with how other things are done.
  d. Examples:

i. In interface, if the word “send” is used in one place, it should be used across the website. Changing it to “submit” in other places might confuse the user.
ii. If every page in the website is designed differently, or has a different color, then the user will feel like they left the site to another site.

2. <u>Standards: the design of the system should follow common practices and conventions of similar systems.</u>
    a. How to make it work?
        i. Follow standards in meanings.
        ii. Follow standards in action sequences.
        iii. Follow standards in organization.
    b. Why does it matter?
        i. This facilitates the learning process.
        ii. It helps in reducing errors.
    c. Exceptions:
        i. If there are no agreed-upon standards for certain things on the system.
        ii. If there is something easier and more intuitive than that currently being done.
    d. Examples:
        i. In interface, the standard is to place the logo in the upper left side. So when you get the website you know quickly how to use it.
        ii. Most websites place the search bar at the top of the page and users are used to that placement, then the system should follow that and place the search bar at the top of the page.

- <u>Error prevention:</u>

There are seven concepts under it:

1. <u>Instructions: give users sufficient guidance before taking any action to avoid making errors.</u>
    a. How to make it work?
        i. It should be visible.
        ii. It should be understandable
        iii. It should be concise.
    b. Why does it matter?
        i. It makes task completion longer.
        ii. It will negatively affect the user experience and mood.
    c. Examples:

        i. When the system asks the user to enter a username and password, there should be instructions next to this request to inform the user about what the username and the password should and should not contain.

2. <u>Constraints: placing constraints on some types of inputs that are clearly invalid to save user time and effort.</u>
    a. How to make it work?
        i. Give users hints on why they are being constrained.
    b. Why does it matter?
        i. It makes task completion longer.
        ii. It will negatively affect the user experience and mood.
    c. Examples:
        i. If the user is booking a flight, the user should not be allowed to enter a return flight date that precedes the date of the departure flight.
3. <u>Confirmation: asking users to confirm an action before doing it to make sure they really want the action to take place.</u>
    a. How to make it work?
        i. Make sure the user knows that you are asking for confirmation.
    b. Why does it matter?
        i. It makes task completion longer.
        ii. It will negatively affect the user experience and mood.
    c. Examples:
        i. If the user is about to send a very large amount of money to someone, the system should ask the user to confirm the transfer to ensure that the correct amount is being sent and that the right person is going to receive it.
4. <u>Notification: the user should be notified about any changes in the system, especially those with serious consequences.</u>
    a. How to make it work?
        i. Tell the user clearly what the notification is about.
        ii. Explain the consequences of ignoring the notification.
        iii. Notify the user about only important things.
    b. Why does it matter?
        i. It makes task completion longer.
        ii. It will negatively affect the user experience and mood.
    c. Examples:
        i. If the user is using his/her phone, he/she might not notice that the battery is running low, so the system should notify him/her at a certain point that the battery is about to die to allow the user to take action.

5. <u>Auto-saving: user inputs should be saved in case something has gone wrong. In such a case, the user would not lose the effort they had expended.</u>
    a. How to make it work?
        i. Let the user know that their input is being auto-saved.
        ii. Saved input should be retrieved easily or automatically.
    b. Why does it matter?
        i. It makes task completion longer.
        ii. It will negatively affect the user experience and mood.
    c. Examples:
        i. On e-learning websites, if the user is writing an essay on the site, the site should autosave the user's inputs so if anything happens, the effort put in by the user does not go to waste.
6. <u>Flexible inputs: the user shouldn't be forced to enter input in a certain form; he or she should be able to enter the input in any form they wish.</u>
    a. How to make it work?
        i. Conversion from form to form should be clearly visible to the user so he or she can assess its accuracy.
    b. Why does it matter?
        i. It makes task completion longer.
        ii. It will negatively affect the user experience and mood.
    c. Examples:
        i. When entering a date, some users are comfortable entering the name of the month, while others are comfortable entering the month as a number; the system should accept both forms.
7. <u>Defaults: the default state of the system should be used carefully to prevent users from making mistakes.</u>
    a. How to make it work?
        i. The defaults should be the most expected things.
        ii. There should be an indication as to what the default state is.
    b. Why does it matter?
        i. It makes task completion longer.
        ii. It will negatively affect the user experience and mood.
    c. Examples:

        If the default of a phone is to not ring when someone calls, then the user would miss calls, as users do not expect that state to be the default when they use any new phone.

- <u>Recognition rather than recall:</u>

  There are two concepts under it:

  1. <u>Availability: anything the user will need to accomplish a certain goal on the system, should be presented to them; the user should have to rely on their memory as little as possible.</u>
     a. How to make it work?
        i. All information should be clear to the user.
        ii. If there is a multi-step task, the information needed should be presented at each step.
     b. Why does it matter?
        i. This increases system efficiency.
        ii. Users shouldn't be expected to recall everything.
        iii. It helps in reducing errors.
     c. Examples:
        i. When people go to a supermarket, there are signs on every aisle to tell them what every aisle contains so they do not need to remember what every aisle contains every time they go to the supermarket.
  2. <u>Suggestions: it is not possible to know what every user wants to accomplish on the system, so, to facilitate the process the user should be provided with suggestions.</u>
     a. How to make it work?
        i. Provide the user with suggestions when they start from an empty state.
        ii. Give the user suggestions when they are searching.
        iii. Provide the user with suggestions of similar or related things when they are browsing.
        iv. Suggestions should be as accurate as possible.
     b. Why does it matter?
        i. It increases the efficiency of the system.
        ii. It reduces the errors.
     c. Exceptions:
        i. It is a key that suggestions be as accurate as possible; if they are not, or couldn't be, then it shouldn't be offered.
     d. Examples:
        i. The suggestions that Google gives when one starts to type in the search bar help the user access what he/she wants.
        ii. In e-shops, when the user is browsing a certain item, the system gives suggestions of items that are frequently purchased with the item the user is browsing, which might remind the user to purchase an item he/she wanted

- Flexibility and efficiency of use:

There are two concepts under it:

1. Flexibility: every major goal or task on the system should be accessible and capable of implementation in more than one way.
    a. How to make it work?
        i. It should accommodate all types of users who are expected to use the system.
        ii. It should accommodate all expected situations of the system use.
    b. Why does it matter?
        i. Most systems are designed to be used by multiple users, and if one only accommodates one group of users, other groups will be excluded.
        ii. Most systems are not going to be used in just one situation. A system that can't be used in different situations will lead users to seek alternatives.
    c. Exceptions:
        i. If a task is very simple and requires very little effort in terms of thinking and doing, it is acceptable that it be done in one way only.
        ii. If the system is not used frequently, or only for a very specific application, it might be necessary to limit the ways in which the system is used.
    d. Examples:
        i. Providing a text reader to accommodate the user when he/she is driving
        ii. Providing shortcuts for expert users.
2. Efficiency: any system goal or task should be performed in the simplest way possible.
    a. How to make it work?
        i. There should be no extraneous steps in an action.
        ii. The effort required to complete an action should be minimum.
        iii. The time required to complete an action should be minimum.
    b. Why does it matter?
        i. One of the main goals of technology is to make achieving user goals faster. Therefore, the system should be as efficient as possible.
        ii. If the system isn't fast and efficient, users can find alternatives with just one click.
    c. Exceptions:

i. The actions taken are intentionally made challenging in some systems, like video games for example. They have different measures for assessing efficiency of actions.

d. Examples:

i. If the user is signing up for a website and is asked to enter his/her phone number, but the phone number will not serve any purpose, then the request for the phone number should be removed because it is just going to require additional unnecessary effort on the part of the user.

- <u>Aesthetic and minimalist design:</u>

There are three concepts under it:

1. <u>Aesthetic: the design of the system should be aesthetically pleasing to the target audience.</u>
    a. How to make it work?
        i. Visual elements should be used appropriately and carefully.
        ii. All elements in the system should be in harmony.
        iii. Videos and audios should be nicely presented.
    b. Why does it matter?
        i. The beauty of the system adds to the positivity of the experience.
        ii. If the system is beautiful, the user would most likely be more forgiving with respect to usability issues and difficulties.
    c. Exceptions:
        i. Aesthetics is important, but it shouldn't be the main goal of the system.
        ii. It should be noted that aesthetics might make users forgive minor issues but not major ones.
        iii. Aesthetics is subjective to some extent so it might be hard to assess it.
    d. Examples:
        i. In real life, some people would prefer certain restaurants not necessarily because they serve the best food. Instead, because the restaurant has a more beautiful place to sit in than others.
        ii. The same thing in apps and websites, some people would prefer using some sites because they are more aesthetically pleasing than other sites.

2. Organization: system content should be presented in an organized and well-ordered manner.
    a. How to make it work?
        i. All related elements should be organized and grouped together.
        ii. Different groups of elements should be distinct from one another.
    b. Why does it matter?
        i. It makes the system easy to learn.
        ii. It will increase efficiency.
        iii. It will reduce user errors.
    c. Examples:
        i. If there are different menus, then they should be organized in a way that clearly separates them.
3. Simplicity: content presented by the system should be limited to necessary content; extraneous content should be removed.
    a. How to make it work?
        i. Any extraneous content should be removed.
        ii. Any content that would distract the user should be removed.
    b. Why does it matter?
        i. It makes the system easy to learn.
        ii. It will increase efficiency.
        iii. It will reduce user errors.
    c. Examples:
        i. Google homepage is pretty simple, the user is not distracted by many things. The only necessary function is provided, which is the search.

- Help users recognize, diagnose and recover from errors:

There are three concepts under it:

1. Recognizing errors: the user should notice when an error occurs.
    a. How to make it work?
        i. Error indications should be presented in a way the user expects.
    b. Why does it matter?
        i. If users don't know they have made mistakes, they will continue doing what they are doing, leading to a bigger problem.
    c. Examples:

        i. In certain sites, when you make an error, the site makes a sound and shows a different color, just to let you know that something went wrong.

2. Understanding errors: the user should understand exactly what the error is.
    a. How to make it work?
        i. Error information should be readable.
        ii. It should be easy to interpret.
        iii. It should be presented in a way that matches the system's purpose.
        iv. Information shouldn’t be presented in a way that blames the user.
        v. Information shouldn’t be presented in an intimidating way.
    b. Why does it matter?
        i. The first step to rectify something wrong is to know what it is.
        ii. A user might change a correct input, thinking that the problem lies with it. This is going to create multiple problems instead of just the original one.
    c. Examples:
        i. When the users enter an incorrect email, the error message explicitly explains what happened by saying “you entered your email in an incorrect form” so the user knows where and what did happen.
3. Recovering from errors: the user should know how to recover from any specific error.
    a. How to make it work?
        i. Recovery information should be presented in an actionable manner.
    b. Why does it matter?
        i. If the user doesn’t know how to solve the problem, he or she will waste time trying different ways to solve the problem.
        ii. This may lead them to try things that were correct in the first place, doubling the issue.
    c. Exception:
        i. It is always important to provide users with actionable solutions, although for an obvious solution this might not be needed.
    d. Examples:
        i. Even if the user knows that they entered their email incorrectly, they might not know how to enter it

correctly. So, the error message should guide them on how to enter it correctly.

- Help and documentation:

There are two concepts under it:

1. Help: the user should be able to contact a support person if he or she faces difficulties or has questions.
    a. How to make it work?
        i. There should be multiple methods for providing help.
        ii. The expectations for each method should be clearly stated.
    b. Why does it matter?
        i. Documentation can't possibly cover all the problems that users might have on the system.
        ii. People don't always read the documentation.
        iii. Some users are still uncomfortable dealing with technology alone; they feel more comfortable in dealing with humans.
        iv. Even if users trust the system, talking to someone may feel more natural to many users.
    c. Examples:
        i. Contact the support team by chat, phone number, email, etc.
2. Documentation: the whole system or at least its most important aspects should be documented and presented to the user in either written or visual form.
    a. How to make it work?
        i. The material should be complete.
        ii. The material should be easy to understand.
        iii. The material should be accessible.
        iv. The material should be searchable.
        v. The material should be relevant.
        vi. The content of the material should be prioritized.
        vii. The content of the material should be categorized.
        viii. Audio/video documentations shouldn't be too long; there should be video/audio support for each problem or related problems.
        ix. Contextual documentation should be presented
    b. Why does it matter?
        i. It helps in learning the system.

    ii. Without it, some users might be hesitant to try things on the system.
    iii. It could reduce the number of user errors.
    iv. It serves as a way to tell the users what the system offers.

c. Exceptions:
    i. Documentation is not always needed. If the system is small and easy to use, it may be enough to provide help only for the unique cases and documentation would be unnecessary.

d. Examples:
    i. Guide or manual of the system.
    ii. Contextual documentation that users can find in the places where difficulties are expected, such as documentation accessed with a question mark button
    iii. Frequently asked questions (FAQs) and tutorials.

APPENDIX L Reading script

<u>Reading Session: Facilitating Heuristic Evaluation (HE) for Novice Evaluators</u>

This interview is being video recorded for research purposes. If you wish for the recording to stop at any point, please let me know and I will stop the recording. Do you agree to being recorded? Recording starts now

----

My name is Anas Abulfaraj. Thank you for talking with me today.

We are doing a research with the aim of making heuristic evaluation easier to apply by novice evaluators.

We have an outline of information we would like to learn about from you. But we would like for you to think about this more as a conversation than a question-and-answer session and the aim of it is to evaluate the understandability of the usability heuristics not you. We'll start out talking about some demographics. Then we will present you with a list of usability heuristics that we want you to read and try to understand. After that, we'll ask you some questions about the understandability of the heuristics. Does that sound OK to you?

There are no right or wrong answers to my questions; I just want to listen to what you have to say. We will be recording this session for the purposes of the study explained earlier. Your responses will be confidential and only used to help us in our research about facilitating heuristic evaluation for novice evaluators.

This session is voluntary. You have the right not to answer any question, and to stop the session at any time or for any reason. Any questions before I begin?

Screening (asked before, but repeated to make sure):

● HCI/UX, Computer Science, Information Technology, Information Systems, Software Engineering, or Computer Engineering student?

● Have less than 4 years of experience in the field of HCI/UX?

● Never done Heuristic Evaluation before outside of a class settings?

Prep:

● Tell us about yourself.

● Age?

● Gender?

● Years of experience in the field?

● Occupation?

● What do you usually work on?

Presented with the one of the two heuristics:

One group, traditional heuristics (10 Usability Heuristics for User Interface Design (nngroup.com)), the other CoHE.

Upon finishing, presented with post reading questions:

We will ask you some questions about the understandability of the heuristics.

Q1: Were you able to completely understand all the heuristics?

Q2: Were you able to understand all the heuristics easily?

Q3: Were you able to understand all the labels easily?

Q4: Were you able to understand all the descriptions easily?

Q5: Did the examples help in explaining the heuristics?

Q6: Can you briefly explain the heuristics?

Q7: Can you give me examples for each heuristic other than the ones given?

Thank you again for taking the time to speak with me today.

APPENDIX M Operational usability heuristics

List of heuristics

- Visibility of system status

    a. State: Users should know what they are capable of doing in the system at any given moment.
        i. When looking at the different elements on the system (links, buttons, etc.), do you know what you can do with them?
        ii. When any changes happen in the system, are they immediately reflected in the system?
        iii. Do changes happen immediately, or do they take time?
    b. Location: Users should know what system they are in and where they are located within the system.
        i. Do you know in which system you are located, and can you determine this easily?
        ii. Do you know the page you are on? Can you find out easily?
        iii. Do you know where you are in relation to other parts of the system? Can you determine this easily?
    c. Progress: The user should know how far they are from accomplishing their goal.
        i. When you work on an action, does the system tell you how long it will take for completion?
        ii. When you complete a multi-page task, does the system tell you how many steps are left?
        iii. When there is limited capacity in the system (storage, memory, etc.) does the system tell you how much is left?
        iv. Are progress measures accurate? If it says it takes one minute, does it really take one minute?
        v. Are both active and passive progress measures provided?
    d. Closure: Users should explicitly know that their goal was accomplished regardless of the outcome.
        i. When you finish a task, does the system provide you with feedback?
        ii. Does the system give you feedback if you got either a right or a wrong result?
        iii. Does feedback appear immediately, or does it take a long time?
        iv. Can you easily interpret the feedback?
        v. If what you did was wrong, does the system offer an explanation?

- Match between system and the real world:

  a. Understandability: The content of the system should be understandable by the users of the system.
     i. Is every piece of system content (text, icons, images, etc.) understandable, specifically, by the target audience?
  b. Natural and logical order: The connection and the tasks in the system should follow a natural and logical order.
     i. Do the steps required to complete a task follow a natural order, i.e., do they follow the order of how the task would be done in the real world?
     ii. If not, are the steps required to complete the task logical, i.e., can you predict what the next step should be?
  c. Appropriateness: The content of the system should be appropriate to the users of the system,
     i. Is every content element in the system (text, icons, images, etc.) appropriate? Do they match the identity of the system?
     ii. Is there an aspect of the system that might be offensive to the target audience?

- User control and freedom:
  a. Reversibility: Users should be able to undo and redo any action they take on the system.
     i. Can you undo any action (deleted, sent, placed, etc.) you have performed on the system?
     ii. Can you redo any action (deleted, sent, placed, etc.) you have performed on the system?
     iii. Are undo and redo easy to do?
     iv. How many steps back can you undo or redo?
  b. Emergency exit: The user should be able to quit any undesirable situation on the system.
     i. Can you escape from any situation in the system?\
     ii. Is it easy to do?
  c. Informing users: Users should know why the system is asking them to enter certain information, and they should know how the information will be handled.
     i. When the system asks you for personal/sensitive information, does it tell you why it is wanted?
     ii. Does the system tell you how this information is going to be stored and handled?

- Consistency and standards:

  a. <u>Consistency: Elements of meaning, function, organization, effort, and feeling should be consistent throughout the whole system.</u>
     i. Does the system always refer to the same element by the same name across the system?
     ii. Do elements that appear identical across the system always do the same thing?
     iii. Do the interface layout and organization have a similar appearance across the system?
     iv. When you work on a multi- page task, is the same effort needed for each page, i.e., is there consistency in the effort needed for each page?
     v. Do the multiple parts of the system have the same feel, i.e., do you perceive the different parts of the system as belonging to one unit?

  b. <u>Standards: The system should take advantage of common practices in similar systems and follow them.</u>
     i. Do you feel that your past experiences with similar systems helped you in using this system?
     ii. Does this help you in understanding content meaning?
     iii. Does it help in performing actions on the system?
     iv. Does the system organization look similar to that of other similar systems?

- Error prevention:

  a. <u>Instructions: Sufficient instructions should be provided to the user before any given task.</u>
     i. Does the system provide you with instructions on how to complete a certain task that requires a specific thing to be done?
     ii. Are instructions clear and easy to understand?
     iii. Are instructions clearly visible?
     iv. Are instructions too long?

  b. <u>Constraints: The system should not allow the user to take action that will lead to obvious errors.</u>
     i. In situations where the action/input is clearly wrong or will lead to erroneous outcome, does the system prohibit that action/input from taking place?
     ii. When you are prohibited from performing a certain action/input, do you have an idea why this has happened?

  c. <u>Confirmation: The system should ask users to confirm their actions to make sure that the action is intended.</u>

i. When committing to a major action that has a long- lasting impact, does the system ask you to confirm the action?

d. <u>Notification: The system should notify users when changes on the system are happening.</u>
   i. When an important/serious event occurs, does the system notify you?
   ii. Does the system tell you what is going to happen if you do not take a recommended action?
e. <u>Autosaving: The system should auto-save users' input to make sure their effort will not be lost if something goes wrong.</u>
   i. When you are entering/writing input that takes considerable time/effort, does the system automatically save your work?
   ii. Do you know when your input is being auto-saved?
   iii. Is input retrieval easy or automatic?
f. <u>Flexible inputs: The system should give users a choice to enter the inputs in a form with which they feel comfortable.</u>
   i. When the system asks you to enter an input that comes in different forms (date, phone number, weight, etc.), does it allow you to enter it in the form you want?
   ii. When it takes a form you like and converts it, can you see the conversion?
g. <u>Defaults: The system should use the most expected defaults.</u>
   i. When you are in an empty state, are the system defaults the expected ones?
   ii. Is there any indication of exactly what the defaults are?

- Recognition rather than recall:

a. <u>Availability: The content that users need to accomplish certain goals should be clearly presented to them, and they should not need to rely on their memory to remember them.</u>
   i. Is everything you would need to accomplish your goal clearly presented to you?
   ii. When completing a multi- step task, is the information presented in one step also needed in other steps presented to you?
b. <u>Suggestions: The system should provide users with suggestions to make the process of remembering their needs easier.</u>
   i. When you are in the empty state, does the system provide you with suggestions on how to proceed?
   ii. When you search for something within the system and don't exactly recall it, does the system provide you with suggestions?

iii. When you browse the system, does it provide you with suggestions of things similar to your browsing targets?
iv. Are these suggestions accurate?

- Flexibility and efficiency of use:

  a. <u>Flexibility: The system should provide users with different ways to accomplish the same goal to accommodate different users and different situations.</u>
     i. Does the system provide you with different paths toward accomplishing the same goal?
     ii. Will users with a variety of abilities and skills be able to use the system?
  b. <u>Efficiency: The tasks sequence of the system should be in its simplest form.</u>
     i. Could the amount of steps/time/effort required to accomplish a goal be reduced?

- Aesthetic and minimalist design:

  a. <u>Aesthetic: The system should be aesthetically pleasing to users.</u>
     i. Is the system aesthetically pleasing to you?
     ii. Do the interface elements appear to be in harmony?
     iii. Are audio/video well- presented in the system?
  b. <u>Organization: The content of the system should be organized in a way that allows the user to distinguish each element.</u>
     i. When looking at the interface, are related elements organized in a way that shows their relationships?
     ii. When looking at the interface, could you easily distinguish among the different elements (menus, paragraphs, etc.)?
  c. <u>Simplicity: The system should not contain any unnecessary content that would distract the user.</u>
     i. Is there any extraneous content on the system (features, icons, texts, etc.) that could be omitted?
     ii. Is there anything in the interface that distracts you from properly focusing?

- Help users recognize, diagnose, and recover from errors:

  a. Recognizing errors: The user should easily recognize that an error happened.
     i. When an error occurs, do you notice it?
     ii. Is the error indication the expected one, or did it take some time to notice it?
  b. Understanding errors: The user should easily understand what error occurred.
     i. When you notice an error, do you know exactly what it is?
     ii. Can you read it easily?
     iii. Is error information written in an appropriate way?
     iv. Is it written in a way that might intimidate or try to blame you?
  c. Recovering from errors: The system should provide the user with a recommendation on how to resolve the error.
     i. When you know of an error, do you know how to resolve it?
     ii. Is the solution presented to you in an appropriately actionable manner?

- Help and documentation:

  a. Help: The system should provide the user with means to contact the help team.
     i. Does the system provide you with the capability to contact the support team?
     ii. Are there multiple ways to contact the support team or only one?
     iii. Do you receive an explanation of how long it will take to receive a support team response?
  b. Documentation: The system should provide the user with sufficient material to learn the system and to know how to overcome any potential obstacles.
     i. Is there documentation from which you can find solutions or learn more about the system?
     ii. Can you find all possible solutions?
     iii. Is the material easy to understand?
     iv. Can you easily find the material?
     v. Can you easily search within the material?
     vi. Does the material prioritize the action frequency?
     vii. Is the material categorized?
     viii. If there are video/audio tutorials, are they too long?
     ix. Is contextual documentation displayed next to the major tasks?

APPENDIX N Inspecting script

Inspecting Session - Facilitating Heuristic Evaluation (HE) for Novice Evaluators

This interview is being video recorded for research purposes. If you wish for the recording to stop at any point, please let me know and I will stop the recording. Do you agree to being recorded? Recording starts now.

My name is Anas Abulfaraj. Thank you for talking with me today.

We are doing a research with the aim of making heuristic evaluation easier to apply by novice evaluators.

We have an outline of information we would like to learn about from you. But we would like for you to think about this more as a conversation than a question-and-answer session and the aim of it is to evaluate the usability of the system and the method used not you. We'll start out talking about some demographics. Then, you will be given a link to the system along with a number of tasks. You will be asked to go through the tasks and evaluate the system along the way. After that you will be asked some questions regarding your experience with the method. Does that sound OK to you?

There are no right or wrong answers to my questions; I just want to listen to what you have to say. We will be recording this session for the purposes of the study explained earlier. Your responses will be confidential and only used to help us in our research about facilitating heuristic evaluation for novice evaluators.

This session is voluntary. You have the right not to answer any question, and to stop the session at any time or for any reason. Any questions before I begin?

Screening (asked before, but repeated to make sure):

- HCI/UX, Computer Science, Information Technology, Information Systems, Software Engineering, or Computer Engineering student.
- Have less than 4 years of experience in the field of HCI/UX.
- Never done Heuristic Evaluation before outside of a class settings.

Prep:

- Tell us about yourself.
- Age?
- Gender?
- Years of experience in the field?
- Occupation?
- What do you usually work on?

Presented with the Website:

Here is the website that we want you to evaluate:

- …………….

- We will give you the login information.
- In the website, we want you to do the following tasks:
- Transferring credits to another person.
- Deactivating an active service.
- Redeeming points to get free minutes of calls.
- Pay the bill.
- Change the number, and then recharge the second number.
- View the service orders for the last year.
- Activate the roaming service.
- Add international minutes.
- File a complaint about a technical issue in the website.
- Change the current package to a different package.
- Get information on how to activate 5G.

Presented with the one of the two heuristics:

One group, traditional heuristics (10 Usability Heuristics for User Interface Design (nngroup.com)), the other CoHE.

Along the way of completing the tasks, use the heuristics to evaluate the usability of the system.

Upon finishing, presented with post inspection questions:

We will ask you some questions about your level of confidence of the results.

Rate your level of agreement with each statement: (Strongly disagree, Disagree, Neutral, Agree, Strongly Agree).

Q1: I am confident that I found all the critical problems on the website.

Q2: I am confident that I provided useful recommendations that addressed the problems of the website.

Q3: I am confident the overall problems that I found will be one of the most useful lists of problems for the company running the website.

Q4: I am confident that real people who will use the website will encounter the problems that I identified.

Q5: I am confident that I found all the related problems to each heuristic in the list.

Q6: I am confident that I linked all the problems to the correct heuristic.

Q7: I am confident that the problems I found covered all the problems that users will encounter.

We will ask you some questions about your overall experience with the method.

Overall experience:

Q1: How would you describe your overall experience with the method?

Q2: What are the things that you liked about the method?

Q3: What are the things that you disliked about the method?

Q4: What are the easiest parts of the process?

Q5: What are the hardest parts of the process?

Q6: If you can change anything, what would it be?

Q7: How likely are you going to use this method in real life situations?

Thank you again for taking the time to speak with me today.

APPENDIX O Information sheet (Experts)

Information Sheet for Participation in Research Study

Facilitating Heuristic Evaluation (HE) for Novice Evaluators

Principal Investigator: Anas Abulfaraj, PhD Student, College of Computing and Digital Media.

Institution: DePaul University, USA.

Faculty Advisor: Adam Steele, PhD, College of Computing and Digital Media.

We are conducting a research study because we want to learn more about the differences between a new way to conduct Heuristic Evaluation called Coherent Heuristic Evaluation (CoHE) and a traditional Heuristic Evaluation. We want to know how they quantitively and qualitatively differ when used by novices. However, novices tend to find false positive usability issues when evaluating systems. Therefore, we want usability experts to participate by evaluating a system so we can use their results as a benchmark to assess the usability issues found by novices. The ultimate goal of this research effort is to facilitate the use of Heuristic Evaluation and make it more accessible for novice evaluators.

We are asking you to be in the research, because you have at least four years of experience in Human-Computer Interaction (HCI) and have conducted at least three sessions of heuristic evaluation.

If you agree to be in this study, you will be asked to evaluate a system. The session will include giving you a link to the system along with a number of tasks. You will be asked to go through the tasks and evaluate the system along the way. We will also collect some personal information about you such as age, gender, education, and occupation. If there is a question you do not want to answer, you may skip it. The session will be video recorded. We are recording this session in order to make accurate written notes of what you have said and done. Once the session is transcribed, the videos will be deleted. This session will take about 90-120 minutes of your time.

Research data collected from you will be kept confidential. Video recordings are considered identifiable, but we will not put your name on the recording file or use it during the recorded session. The recording file will be labeled with a random number, such as participant 1, participant 2, etc. and then when we transcribe the recording into written notes, there will be no identifiers present on the transcripts.

Your participation is voluntary, which means you can choose not to participate. There will be no negative consequences if you decide not to participate or change your mind later after you begin the study. You can withdraw your participation at any time, by letting me know before the session ends or shortly after the session is completed. Once I transcribe the session and delete the recordings, I will not know which transcript belongs to you.

If you have questions, concerns, or complaints about this study or you want to get additional information or provide input about this research, please contact me at aabulfa2@mail.depaul.edu.

If you have questions about your rights as a research subject, you may contact Jessica Bloom in the Office of Research Services at (312) 362-6168 or via email at jbloom8@depaul.edu. You may also contact DePaul's Office of Research Services if:

1. Your questions, concerns, or complaints are not being answered by the research team.
2. You cannot reach the research team.
3. You want to talk to someone besides the research team.

You may keep [or print] this information for your records.

I have explained the study to you, and by completing the interview session, you are indicating your affirmative agreement to be in the research.

APPENDIX P Inspecting script (Experts)

Inspecting Session - Facilitating Heuristic Evaluation (HE) for Novice Evaluators

This interview is being video recorded for research purposes. If you wish for the recording to stop at any point, please let me know and I will stop the recording. Do you agree to being recorded? Recording starts now.

My name is Anas Abulfaraj. Thank you for talking with me today.

We are doing a research with the aim of making heuristic evaluation easier to apply by novice evaluators.

We have an outline of information we would like to learn about from you. But we would like for you to think about this more as a conversation than a question-and-answer session and the aim of it is to evaluate the usability of the system and the method used not you. We'll start out talking about some demographics. Then, you will be given a link to the system along with a number of tasks. You will be asked to go through the tasks and evaluate the system along the way. Does that sound OK to you?

There are no right or wrong answers to my questions; I just want to listen to what you have to say. We will be recording this session for the purposes of the study explained earlier. Your responses will be confidential and only used to help us in our research about facilitating heuristic evaluation for novice evaluators.

This session is voluntary. You have the right not to answer any question, and to stop the session at any time or for any reason. Any questions before I begin?

Screening (asked before, but repeated to make sure):

● Have at least four years of experience in the field of HCI/UX?

● Have done at least three Heuristic Evaluation sessions before?

Prep:

● Tell us about yourself.

● Age?

● Gender?

● Years of experience in the field?

● Occupation?

● What do you usually work on?

Presented with the Website:

Here is the website that we want you to evaluate:

………………

We will give you the login information.

In the website, we want you to do the following tasks:

- Transferring credits to another person.
- Deactivating an active service.
- Redeeming points to get free minutes of calls.
- Pay the bill.
- Change the number, and then recharge the second number.
- View the service orders for the last year.
- Activate the roaming service.
- Add international minutes.
- File a complaint about a technical issue in the website.
- Change the current package to a different package.
- Get information on how to activate 5G.

Thank you again for taking the time to speak with me today.

APPENDIX Q Raw data

Reading time for condition one

Participant	*Time*
P1	8:07 minutes
P2	10:03 minutes
P3	6:17 minutes
P4	7:16 minutes
P5	11:32 minutes
P6	13:06 minutes
P7	6:23 minutes
P8	9:48 minutes
P9	8:45 minutes
P10	6:19 minutes

Reading time for condition two

Participant	*Time*
P1	24:18 minutes
P2	37:52 minutes
P3	18:22 minutes
P4	24:23 minutes
P5	29:36 minutes
P6	30:45 minutes
P7	32:38 minutes
P8	25:32 minutes
P9	30:01 minutes
P10	30:54 minutes

Missing concepts for condition one

Participant	*Heuristics*
P1	Consistency and standards, Flexibility and efficiency of use, Aesthetic and minimalist design, Help users recognize, diagnose and recover from errors, Help and documentation.
P2	Consistency and standards, Flexibility and efficiency of use, Aesthetic and minimalist design, Help users recognize, diagnose and recover from errors, Help and documentation.
P3	Flexibility and efficiency of use, Aesthetic and minimalist design, Help and documentation.
P4	Flexibility and efficiency of use, Aesthetic and minimalist design, Help and

	documentation.
P5	Consistency and standards, Aesthetic and minimalist design, Help users recognize, diagnose and recover from errors, Help and documentation.
P6	Flexibility and efficiency of use, Aesthetic and minimalist design, Help users recognize, diagnose and recover from errors, Help and documentation.
P7	Consistency and standards, Flexibility and efficiency of use, Aesthetic and minimalist design, Help users recognize, diagnose and recover from errors, Help and documentation.
P8	Flexibility and efficiency of use, Help users recognize, diagnose and recover from errors.
P9	Consistency and standards, Flexibility and efficiency of use, Help users recognize, diagnose and recover from errors, Help and documentation.
P10	Flexibility and efficiency of use, Aesthetic and minimalist design, Help and documentation.

Missing concepts for condition two

Participants	*Heuristics*
P7	Help users recognize, diagnose and recover from errors.

Concepts confused for condition one

Participants	*Heuristics*
P1	Consistency and standards with recognition rather than recall.
P3	Visibility of system status with flexibility and efficiency of use, Error prevention with help users recognize diagnose and recover from errors.
P6	Visibility of system status with match between system and the real world, Error prevention with recognition rather than recall.
P7	User control and freedom with visibility of system status.
P8	Consistency and standards with match between system and the real world.
P10	User control and freedom with error prevention.

Concepts confused for condition two

Participants	*Heuristics*
P2	Consistency and standards with match between system and the real world, Aesthetic and minimalist design with recognition rather than recall.
P4	Aesthetic and minimalist design with flexibility and efficiency of use.
P8	Aesthetic and minimalist design with flexibility and efficiency of use.
P9	User control and freedom with error prevention.

Concepts not understood for condition one

Participants	*Heuristics*
P1	Flexibility and efficiency of use, Help and documentation.
P2	Match between system and the real world, Recognition rather than recall.
P3	Consistency and standards.
P5	Flexibility and efficiency of use.
P7	Recognition rather than recall.
P8	Recognition rather than recall, Help and documentation.
P9	Match between system and the real world, Flexibility and efficiency of use, Aesthetic and minimalist design.

Concepts not understood for condition two

Participant	*Heuristics*
P6	Match between system and the real world, Consistency and standard.
P7	Visibility of system status, Consistency and standards.
P8	Visibility of system status.
P10	Recognition rather than recall.

Comments for condition one

Comment	*No. of Participants*	*Participants*
Lack of examples	7	P1, P2, P3, P6, P7, P9, P10
Unclear words	5	P1, P2, P4, P8, P9
Merging Heuristics	7	P1, P4, P5, P6, P7, P8, P9
Incomplete descriptions	6	P2, P4, P6, P8, P9, P10
Unclear labels	6	P2, P4, P5, P6, P8, P9

Comments for condition two

Comment	*No. of Participants*	*Participants*
Unclear examples	3	P1, P3, P7
Miscategorization	4	P5, P7, P9, P10
Unclear labels	5	P5, P7, P8, P9, P10
Organization	3	P1, P5, P7

Evaluation time for condition one

Participant	*Time*
P1	85:23 minutes
P2	70:12 minutes
P3	80:07 minutes
P4	100:36 minutes
P5	79:20 minutes
P6	99:58 minutes

P7	80:21 minutes
P8	72:23 minutes
P9	83:05 minutes
P10	89:42 minutes

Evaluation time for condition two

Participant	*Time*
P1	93:29 minutes
P2	104:27 minutes
P3	100:22 minutes
P4	99:19 minutes
P5	101:35 minutes
P6	80:51 minutes
P7	118:33 minutes
P8	97:10 minutes
P9	108:53 minutes
P10	115:32 minutes

Number of problems for condition one

Participants	*Number of Problems Found*
P1	22 problems
P2	17 problems
P3	21 problems
P4	26 problems

P5	20 problems
P6	28 problems
P7	20 problems
P8	18 problems
P9	19 problems
P10	29 problems

Number of problems for condition two

Participants	*Number of Problems Found*
P1	26 problems
P2	29 problems
P3	34 problems
P4	24 problems
P5	30 problems
P6	18 problems
P7	37 problems
P8	26 problems
P9	25 problems
P10	29 problems

Number of real problems for condition one (Relaxed)

Participant	*Number of Real Problems Found*
P1	21 problems
P2	17 problems

P3	19 problems
P4	20 problems
P5	18 problems
P6	22 problems
P7	16 problems
P8	18 problems
P9	17 problems
P10	28 problems

Number of real problems for condition two (Relaxed)

Participant	*Number of Real Problems Found*
P1	25 problems
P2	26 problems
P3	26 problems
P4	23 problems
P5	25 problems
P6	17 problems
P7	32 problems
P8	23 problems
P9	24 problems
P10	25 problems

Number of real problems for condition one (Strict)

Participants	*Number of Real Problems Found*
P1	18 problems
P2	13 problems
P3	16 problems
P4	14 problems
P5	15 problems
P6	20 problems
P7	14 problems
P8	17 problems
P9	15 problems
P10	23 problems

Number of real problems for condition two (Strict)

Participants	*Number of Real Problems Found*
P1	18 problems
P2	19 problems
P3	24 problems
P4	20 problems
P5	23 problems
P6	17 problems
P7	27 problems
P8	19 problems
P9	21 problems
P10	21 problems

The severity of problems for condition one

Participant	*Cosmetic*	*Minor*	*Major*	*Catastrophe*	*Total*
P1	0	5	8	5	18
P2	0	1	6	6	13
P3	0	3	6	7	16
P4	0	2	5	7	14
P5	1	6	4	4	15
P6	1	3	10	6	20
P7	0	4	5	5	14
P8	0	4	5	8	17
P9	0	3	3	9	15
P10	0	3	9	11	23

The severity of problems for condition two

Participant	*Cosmetic*	*Minor*	*Major*	*Catastrophe*	*Total*
P1	0	3	7	8	18
P2	0	3	6	10	19
P3	1	6	6	11	24
P4	0	4	5	11	20
P5	1	2	7	13	23
P6	0	1	6	10	17
P7	3	4	10	10	27
P8	1	5	5	8	19
P9	0	4	6	11	21
P10	1	3	8	9	21

The mean and the standard deviation of the strict problems for each condition

	HE	*CoHE*
Mean	16.5	20.9
Standard deviation	3.1002	3.0349