Uncovering Gulf of Execution and Gulf of Evaluation in User-Computer Interactions

A Comparative Study between Manual Analysis and Questionnaire-Based Approach

Bachelor Thesis
Faculty of Science, University of Bern

submitted by

Tunahan Özsoy

from Switzerland, Home Country

Supervision:

PD Dr. Kaspar Riesen
Mathias Fuchs
Institute of Computer Science (INF)
University of Bern, Switzerland

Abstract

This bachelor thesis explores the field of Human-Computer Interaction (HCI) and the role of '7 Stages of Action Model' by Donald Norman in identifying user-system interaction or usability problems, termed "gulfs". This thesis has a textbook application of Norman's model for interaction analysis across nine different products, contributing a valuable addition to the limited examples currently available in the HCI literature. Additionally, for this thesis, a questionnaire is designed based on a simplified Norman's model, aiming to identify in an interaction at least 50% of the gulfs discovered in the expert-based evaluation. These gulfs, referred to as the Gulf of Execution (user's understanding of how the system works) and the Gulf of Evaluation (user's comprehension of the system's reaction after interacting with it), are analyzed through Think Aloud interviews using three different products. Two versions of the questionnaire were presented, and their effectiveness was evaluated by comparing them to the Ground Truth. The Ground Truth can be described as a result of the detailed expert-based evaluation of user interactions.

The results reveal that while the questionnaires achieved more than 50% accuracy in gulf identification, they still missed a significant number, and the performance varied due to users' diverse mental models. Despite certain limitations and the potential for improvement, the questionnaires provide a valuable tool for quick usability checks, and their findings may be of significant interest to HCI researchers and developers.

Contents

Chapter 1

Introduction

This bachelor thesis focuses on Human-Computer Interaction (HCI). HCI is an interdisciplinary field of research with numerous linked disciplines. For example, HCI draws on psychology, computer science, design, and cognitive science, among others. As a result of the rapid development of technology in the early 1980s, HCI gained significant popularity. Apple introduced the Apple Macintosh in 1984. The first commercially viable, reasonably priced personal computers have a graphical user interface. With this, the transition from the usage of computers in large, protected, and cooled laboratories by a few technically skilled people began. Now, non-technically skilled men and also women could purchase and utilize personal computers. As a consequence, the demand for Human-Computer Interaction (HCI) and the importance of it grew significantly.

Human-Machine Interaction (HMI), Man-Machine Interaction (MMI), and Computer-Human Interaction are alternative names for HCI. Although all of these concepts relate to the interaction between humans and computers or other technology, they are not identical. In this thesis, the term HCI will explicitly be used for Human-Computer Interaction.

Additionally, there is the term Human-Computer Interfaces, which is different from HCI (Human-Computer Interaction). Human-Computer Interaction will be exclusively abbreviated with HCI.

HCI has gotten more attention in recent years as a result of rapid technological breakthroughs and the increasing importance of technology in our daily lives. Current research in HCI includes the following topics:

- Systems with artificial intelligence (e.g., partially autonomous automobiles)

- HCI in virtual reality and augmented reality
- Big Data
- HCI in ubiquitous computing and wearable computing
- HCI in social computing and collaborative computing and many others

The Association for Computing Machinery (ACM) defines human–computer interaction as "a discipline that is concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." [1]. The objective is to design and build computer systems and technologies that are user-friendly, efficient, and enjoyable. Poorly planned human-machine interaction can result in numerous unanticipated issues. Three Mile Island is a typical example of a nuclear meltdown, where investigations revealed that the design of the human-machine interface was at least partially to blame for the catastrophe. [2] [3] [4].

HCI academics have developed a number of models and frameworks with the aim of achieving good, efficient, and seamless interactions. The '7 Stages of Action Model' created by Donald Norman has received significant attention in this field. In other literature, it is sometimes referred to as Norman's Interaction Model.

The interaction model developed by Norman is arguably the most significant model in HCI. This model is a valuable foundation for analyzing human-computer interaction. This approach helps in identifying interaction issues so that they can be resolved and enhanced.

When individuals use any product/device, they encounter two gulfs: the Gulf of Execution, in which they attempt to determine how it works, and the Gulf of Evaluation, in which they attempt to determine what occurred. Ed Hutchins, Jim Hollan, and Don Norman came up with these two terms in 1986 when discussing the advantages of 'direct manipulation [1]' in bridging these gulfs for the user [6]. The rule for designers is to assist users in bridging these two gulfs [7]. This model allows for the discovery of these gulfs.

[1]The term 'direct manipulation' is an interaction paradigm, coined by Ben Shneiderman, in which users can directly manipulate objects on a graphical user interface (GUI) through physical actions such as pointing, clicking and dragging. [5]

To really bridge the gulfs, the system should have good interfaces that facilitate the interaction between the users and computers more efficiently. Right now, one of the biggest challenges for the HCI field is to design for multiple devices. As people increasingly use multiple devices, designing interfaces that work seamlessly across multiple devices and platforms is important.

This study aims to find the Gulf of Executions and Gulf of Evaluations by analyzing interactions between users and computer-based systems, including both physical devices and digital products, using Norman's Seven Stages of Action model. This analysis could be beneficial for students studying Don Norman's model and these Gulfs. Sometimes the categorization is difficult and perhaps open to interpretation. This work could be beneficial to students and the field because there are few examples of this analysis using Norman's Interaction model.

In addition, there will be a questionnaire constructed that simplifies Norman's Seven Stages of Action Model to assist users and developers in identifying similar gulfs in their interactions with the products as part of this project. The questionnaire will be assessed through participant interviews. They will interact with a product while simultaneously attempting to answer the questions. The questionnaire will immediately detect the gulfs, which will be compiled at the end to compare them with the 'Ground Truth'. The goal is to find out if this questionnaire can identify at least 50% of the Gulf of Executions and Gulf of Evaluations that were found from the expert-based evaluation using Norman's model.

This thesis consists of eight chapters. Chapter 2 gives the essential theory to comprehend this bachelor thesis in its entirety. There will be provided a comprehensive explanation of Norman's 7 Stages of Action model, the Gulf of Execution, and the Gulf of Evaluation. In Chapter 3, there will be a demonstration of the analysis of three product interactions using Norman's Seven Stages of Action Model. The analysis for the other products is in the appendix for completeness. The questionnaires that were designed for the interviews are shown in Chapter 4. Chapter 5 discusses the methods utilized when conducting the interviews. The findings from the interviews are presented in Chapter 6. Chapter 7 discusses the ramifications of these findings for the field of HCI. In Chapter 8, the thesis gets concluded and makes recommendations for further research.

Chapter 2

Theoretical Basis

2.1 HCI research

At about the same time that the computer began to establish itself as a commercial product in the early 1980s, HCI established itself as an independent field of research. The conference Human Factors in Computing Systems (CHI) of the Special Interest Group on Computer-Human Interaction (SIGCHI) series was the one that kicked off the HCI research. This conference is still considered the most important international conference series in the field of HCI. In addition, there are a number of international conferences on HCI, some with a focus on a specific subarea, such as the ACM MobileHCI conference. Traditional research in HCI is focused on standard measurements such as:

- How long does it take to learn a new Human-Computer Interface?
- How fast can users finish a given task?
- How many errors did the users make in the process?

These three questions can be answered using methods such as usability testing, time and motion studies, or error analysis. In this thesis, the goal is to discover more than 50% Gulfs in an interaction. The approach will include a usability test, which means participants will have to complete tasks on systems guided by the questionnaire while being observed by the researcher. This kind of test will reveal important Gulfs and other usability issues. In this thesis, there will also be an error analysis of some of the important Gulfs based on the answers to the questionnaire.

Current research areas in HCI include:

- HCI with systems that have artificial intelligence (e.g., in semi-autonomous vehicles)

- HCI in virtual or augmented reality
- HCI with Big Data
- HCI in Ubiquitous Computing and Wearable Computing
- HCI in Social Computing and Collaborative Computing
- HCI in Gaming and Learning
- Diversity in HCI (age, gender, culture, religion, impairment, etc.)
- Ethics and privacy in HCI

2.1.1 Developments after 1985

In the 1980s, HCI assumed that a man interacted with a computer to do tasks. Since that time, HCI has evolved along several dimensions; the reasons are many, but three are particularly important:

- Establishment of new types of devices
- Emergence of new goals and tasks
- Networking of users

The result of this evolution is that instead of a 1:1 interaction between a man and his computer to solve a task, HCI now takes place in communities on a wide variety of ubiquitous devices and helps people achieve a wide variety of goals.

Establishment of new types of devices Even before the classic desktop computer, new types of devices were already emerging, some of which allowed new types of interaction and required new paradigms.

- Laptops
- PDA
- Tablets
- Smartphones
- Smartwatches

With the introduction of touch-sensitive devices, the development is not finished. Currently, the ubiquitous inclusion of computers in human living spaces is being observed. This computer penetration in everyday life is called pervasive computing and can be divided into two areas:

- Wearable Computing
- Ubiquitous Computing [1]

New goals and networking among users: Computers were mostly used for office applications. Now, digital products are used for a wider range of tasks and goals. An important aspect of this is social computing. Social software can be any computerized system that supports social interactions between groups of people. The email application, developed in the mid-1980s, can be considered the starting point for social computing. Examples of applications in 'social computing' are:

- Social media
- Social networks
- Wikis
- Blogs, online games, online dating, online auctions, and many more

2.2 Conceptual models

In HCI, the conceptual model is vital to understanding and designing good user interfaces. Don Norman defines the conceptual model as follows: 'A conceptual model is an explanation, usually highly simplified, of how something works. It does not have to be complete or even accurate, as long as it is useful.' [7]
It is an external representation that explains and communicates how a system works. In other words, it is an abstraction to convey complex systems in a more understandable and manageable way. For example, digital products are made up of mechanisms and abstract concepts, such as algorithms and data structures, that help achieve users' goals.

[1] Ubiquitous computing is a concept in computer science and engineering where computing is seamlessly integrated into everyday life and becomes an invisible part of our environment.

A conceptual model is the set of mechanisms and concepts that define how a machine or application actually works. There are two conceptual models:

- Designers' conceptual model
- Users' conceptual model

For Don Norman, the designer's conceptual model is the designer's conception of the product [7]. That means they have their own mental representation or understanding of the product. The designer's goal is to convey the conceptual model to the user as closely as possible. Because in that way, the user will have a smooth experience using the product. But achieving that is not easy, as the designer and the user cannot communicate directly. The communication happens indirectly via the system image.

The system image is the combined information available on the product, usually in the form of instruction manuals, advertisements, experience using similar things in the past, and the product itself with feedforward[2] and feedback. When users interact with the system image or product by reading the manuals or searching online for documentation, their conceptual model can be formed (Figure 2.1). But the user's conceptual model is subjective and can be different from user to user. That is why it is important that designers provide complete and understandable information about their product, as the burden of communication is on the system image.

Users tend to form conceptual models that are simpler than the implementation model actually is. There is nothing wrong with that. People do not need to understand in detail how a complex product actually works in order to use it without error.

But if the system image communicates so poorly that users cannot infer an appropriate mental model, this will inevitably lead to errors in the interaction. When users use a product, they face two gulfs: the Gulf of Execution and the Gulf of Evaluation.

2.3 Gulf of Execution

Don Norman defines the Gulf of Execution in his book 'The Design of Everyday Things' as follows: The Gulf of Execution is the difference between the intentions of

[2]feedforward refers to providing users with information or cues about the potential outcome or result of their actions before they actually take those actions.

Figure 2.1: Designers cannot communicate directly with the user; that is why the burden of communication is on the system image. In this context, the system image can be interpreted as a kind of translation aid that translates the designer's conceptual model into the user's conceptual model. (Image from the book 'Design of Everyday Things' [7])

the users and what the system allows them to do or how well the system supports those actions. [8]
In other words, when people use something and try to figure out how it operates, they face a Gulf of Execution. The goal of the designer is to bridge these gulfs. In terms of the Gulf of Execution, the designer can make things more visible. They can provide feedforward information to make the options readily available (Section 2.7). That means through the use of signifiers, constraints, mappings, and a good conceptual model [9].

2.4 Gulf of Evaluation

The Gulf of Evaluation is the difficulty of assessing the state of the system and how well the artifact supports the discovery and interpretation of that state [8]. The gulf is small when the system provides information about its state in a form that is easy to get, easy to interpret, and matches the way the person thinks of the system. The designer of a product can bridge these gulfs through the use of clear feedback and a good conceptual model.
The next section examines a crucial model by Don Norman to analyze interactions between humans and computers. This model can help discover these Gulfs.

2.5 Norman's Seven Stages of Action Model

To analyze the interaction between a human and a computer, Donald Norman came up with a model that tries to understand the stages a person goes through when interacting with a product. This model is helpful for finding problems in the interaction, like the Gulf of Executions and the Gulf of Evaluations. Donald Norman describes the seven stages of his model as follows:

1. Formulating the Goal
2. Forming the Intention
3. Specifying the Action
4. Executing the Action
5. Perceiving the Systems State
6. Interpreting the Systems State
7. Evaluating the Outcome

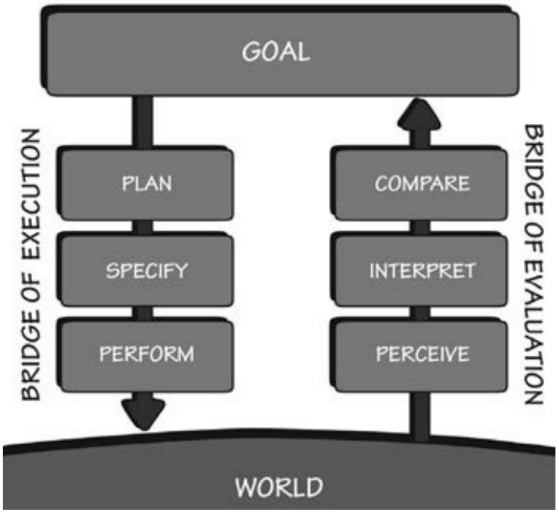


Figure 2.2: The Seven Stages of the Action Cycle (Image from 'The Design of Everyday Things' [7]

Each stage of the action cycle is a user activity. In goal-oriented action, users first define a goal. This goal definition can be vague and imprecise and must be translated into more concrete plans and actions. Often, there are different ways to achieve the goal. In the planning stage, users decide which of the many possible plans they want to pursue. Then, users must specify how they will implement the chosen plan. That is, they define concrete actions that, from the users' point of

view, will help them achieve the goal. Only after that comes the actual execution of the actions. Thus, there are three stages of execution: Plan, Specify and Execute. In Figure 2.2, these three stages are called the 'Bridge of Execution'. If the user has problems in these stages, that could lead to a Gulf of Execution. The same goes for the 'Bridge of Evaluation' and the Gulf of Evaluations.

The evaluation phase also has three stages: Perceiving the state, interpreting it, and finally, comparing it to the goal. If the perceived and interpreted system state matches the users' goal, the product does what the users want, and the interaction is successful. Otherwise, users must formulate a new goal or plan for the existing goal and thus take further actions to restart the cycle.

Most goals cannot be achieved in a single iteration through the seven stages of action. Many cycles are required, and an action may take hours or even days. There are many feedback loops where the results of one action are used to drive further actions. There are also situations where goals become subordinate goals and plans become subordinate plans. In some cases, the original goals are also completely forgotten or completely reformulated.

For most interactions, it is unnecessary to go through all the stages consciously. If it is an everyday activity in which users are experienced and skilled, most actions happen subconsciously. Someone can perform many actions and go through the cycle again and again without really noticing what they are doing. On the other hand, when users are still learning or hit a dead end in an activity they have already learned, setting up the plan, determining exactly what to do, and evaluating the outcome happen very consciously.

2.6 Bridging the gulfs

Unbridged gulfs are the source of serious usability problems in many digital products. How can these Gulfs be bridged? By applying two fundamental principles, around which the next sections revolve:

- Feedforward
  - Affordances
  - Signifiers
  - Constraints

- Mapping

- Feedback

2.7 Feedforward

Feedforward is the collective information on the product that helps users answer the questions of execution so that they can smoothly cross the 'Bridge of Execution' in Figure 2.2.
The questions for the 'Bridge of Execution' are :

- What are the alternative action sequences?
- What action can I do now?
- How do I do it?

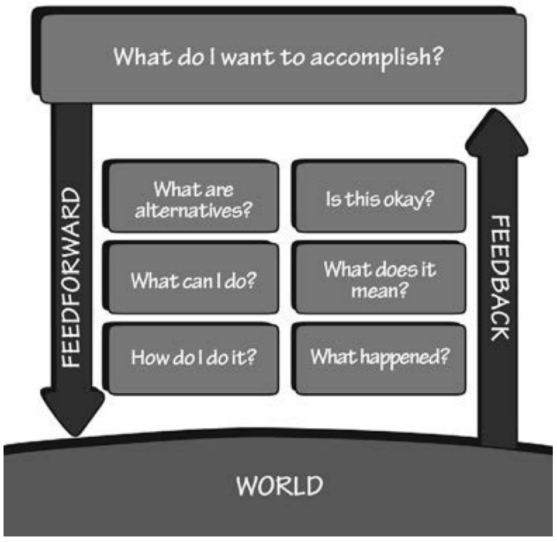


Figure 2.3: Feedforward can help to answer the questions of execution. (Image from the book 'The Design of Everyday Things [7]'

Feedforward is accomplished through the appropriate use of signifiers, constraints, and mappings. The conceptual model also plays an important role [7].
One example of feedforward that helps to answer the question 'What can I do?' would be simply the label on the button. The label tells users what happens if they push the button [10].
Another example would also be Hover Effects. When a user moves their cursor over a clickable element, like a button or a link, the element can change its look to provide feedforward. This change can be shown by things like highlighting, changing

the color, or showing more information. Before clicking, these effects show users what might happen if they do something.

2.7.1 Affordances

The affordance corresponds to a thing's perceived and actual properties, primarily those basic properties that determine how the thing might be used. When objects are clearly shaped to fit the hands or body, it is quickly recognized how they might be used.
Thus, affordance is rather a relation between the object and the person and not necessarily a property of the object. [7]

Imagine someone encountering a physical door with a handle. In this case, the affordance of the door handle is to be grasped and pulled or pushed. The handle's design, such as its shape, size, and placement, provides visual and physical cues that indicate how it should be operated. This is an example of an affordance in the physical world.

In the digital realm, one example of an affordance would be a button on a website. The button has a three-dimensional appearance with shadows and gradients and is placed on a colorful background. These visual cues create an affordance that suggests that the button can be pressed or clicked. The button's design provides a clear indication of its interactive nature, implying its functionality for the user.

2.7.2 Signifier

In the real world, an object does what it does based on its physical form and its connections with other physical objects[3]. In the digital world, however, an object does what it does because developers have given it the power to do so.
If a digital object had no visible cues on what the object does, the user would not know what the functionality of the object is.

Signifiers play a crucial role in bridging the gap between the invisible functionality of a digital element and the user's understanding of what the element does. By using signifiers effectively, developers can create intuitive interfaces that are understandable and easy to use.
Interface elements in digital products almost always need to be accompanied by text or symbols so that users can understand their function. To do this, signifiers

[3]For example, a doorbell that is next to a door

are used, which refer to any markings or labels that tell the user the correct way to act with an object. Signifiers are thus signals that indicate how to proceed or what an action on an element will do.

The affordance of a product determines what actions are possible with it (print, tap, drag, slide, etc.). For example, on a touchscreen, the affordance exists to move the finger up, down, or sideways or to tap on it. Signifiers in the other hand, help show where to press and what actions they will cause.

2.7.3 Mapping

A mapping in HCI refers to the relationship between the control elements and the resulting outcomes. Mapping is an important concept in the design of User Interfaces. Let us imagine a non-digital scenario: Suppose there are many lamps on the ceiling of a room and a row of light switches on the wall. The mapping defines which control element (in this case, switches) controls which device (in this case, lamps). In the best case, the control elements are located directly where the object to be controlled is (or at least very close). This is because it is very easy for users to find out which control element is connected to which object. However, this is not always possible (e.g., with ceiling lamps) or is sometimes associated with safety problems (e.g., with stove tops).

Even if the control elements are not directly located at the object to be operated, there are mappings that support the users: So-called 'natural mappings'. In natural mapping, the layout of the controls and the devices to be controlled match spatially. With natural mapping, users can usually immediately understand which control performs which action.

In addition to natural mappings based on spatial correspondences, there are also mappings that are culturally determined. For example, the universal gesture that raising a hand (or turning it to the right) signals 'more' and lowering it (or turning it to the left) signals 'less'. This is the reason why it is appropriate to display intensities or quantities vertically and for knobs to have the intensity increase when turned to the right.

2.7.4 Constraints

Constraints are useful hints about what can be done with a device or in a situation by specifically limiting the possible actions. Constraints thus reduce the number of possible actions and can thus make the desired interactions more obvious. Con-

straints are more effective and useful if they restrict the possible actions even before anything has been done. Otherwise, constraints prevent wrong actions after they have been tried.
A distinction is made between physical constraints and virtual constraints.

Physical constraints

Physical constraints reduce the number of existing or possible operating options in the real world. USB connectors, for example, are physically constrained, but depending on the type, these constraints are so inconspicuous that it is sometimes difficult to find the correct orientation (Type C is designed so that orientation no longer matters).

The traditional cylindrical AAA battery has insufficient physical constraints. It can be inserted in two different positions in a battery slot: one is correct, and the other can damage the device. For correct handling, signifiers are needed. Devices with batteries often solve the problem better, e.g., by preventing the battery compartment from closing if the battery is placed the wrong way (but this is a suboptimal constraint that constrains the user only after they have tried the action).

Virtual constraints

One of the most obvious constraints in the virtual world is that the mouse pointer cannot be moved beyond the desktop. Another prominent example is the graying out of options: In Figure 2.4, the paste action is grayed out to indicate that this action is not accessible to the user at this time because there is nothing that got copied.

2.7.5 Forcing functions

Forcing functions are strong constraints (applicable to physical and virtual worlds). Interlocks and lock-ins are both forcing functions.An interlock enforces that sequences of actions take place in a certain order. For example, microwave ovens usually use interlocks to prevent people from opening the doors before the heating process is complete.

A lock-in keeps an action active and thus prevents it from being stopped prematurely. Many computer programs have lock-ins that ask the user when they close the program if they really want to exit (without saving the data, for example).

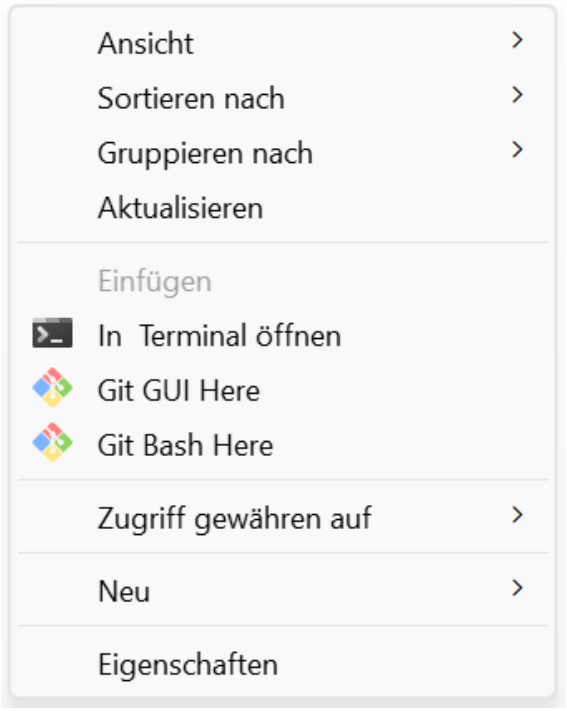


Figure 2.4: Virtual Constraint

2.8 Feedback

Feedback should be immediate. If feedback appears only slightly delayed, this can be unnerving for users. Many people give up and move on to other things if the delay is too long. In principle, all actions users take should be acknowledged, but this must be done subtly. There are many situations where feedback is poorly implemented. Feedback should not stop the current flow of action by reporting something obvious or ordinary. Using a constraint, for instance, can often prevent feedback on incorrect data entry.

If there is too much feedback, people tend to ignore it or, as far as possible, turn it off, leading to missing important messages. Until recently, most designers have used the same tool, the dialog box, to convey feedback to the user. Unfortunately, this means that subtle status information is never communicated to users because most designers do not want dialog boxes to always appear (thus stopping user flow). But constant feedback, especially positive feedback, is exactly what users need.

Feedback with detailed information about the status or attributes of a process or object in the current application is also possible without dialog boxes. Alan Cooper refers to this as modeless feedback. [11]

Feedback is modeless if the information for users is built into the structures of the interface without interrupting the normal flow of activities. Modeless feedback is characterized, in particular, by the fact that this information is subtly displayed all the time, and no special action is required on the part of the user to obtain or close the feedback.

2.9 Errors

Estimates suggest that up to 90% of all industrial accidents are the result of human error [12]. Why are so many people so incompetent? The answer is: They are not. Those accidents are caused by bad design [7].

Definition of Human Error: Human Error is defined in Norman's book as: "deviance from the generally accepted correct or appropriate behavior" [7]

2.9.1 Classification of Human Error

Don Norman and the British psychologist James Reason divided human error into two categories: slips and mistakes [7].

Slips

A slip occurs when a person intends to do one action that was learned but ends up doing something else. With a slip, the action performed is not the same as the action that was intended [7]. Slips happen more than mistakes because people often perform learned procedures.

There are two major classes of slips: action-based and memory-lapse. In action-based slips, the wrong action is performed. In lapses, memory fails, so the intended action is not done, or its results are not evaluated. Action-based slips and memory lapses can be further classified according to their causes. [7]

Mistakes

A mistake occurs when the wrong goal is established, or the wrong plan is formed. From that point on, even if the actions are executed properly, they are part of the error because they are inappropriate and part of the wrong plan. [7]
Mistakes have three major classes: *rule-based*, *knowledge-based*, and *memory-lapse*. A rule-based mistake occurs when a person accurately identifies the situation but

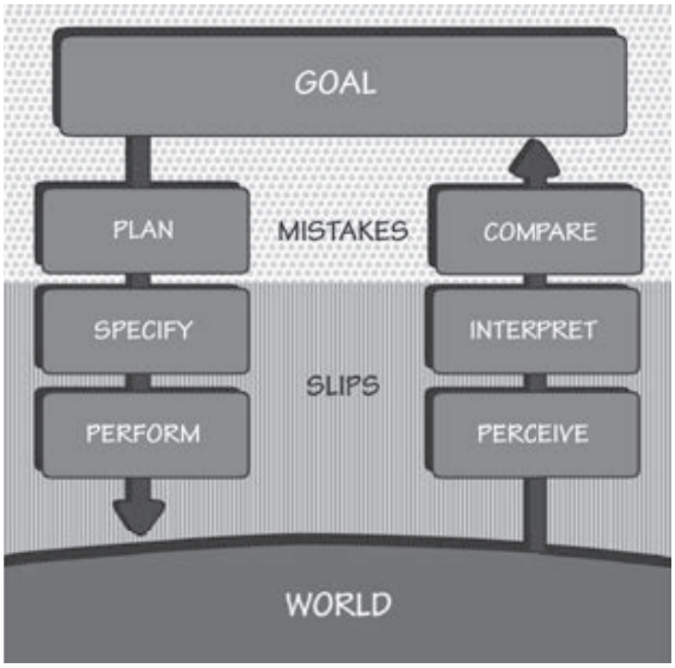


Figure 2.5: Action-based slips come from the bottom four stages of the action cycle, and mistakes come from the top three stages. (Image from the book 'The Design of Everyday Things' [7])

then selects and follows an incorrect rule or course of action. In a knowledge-based mistake, the problem is misdiagnosed because of incomplete knowledge. The memory-lapse mistake involves a lapse in memory that hinders the user's ability to execute their intended actions or accurately assess their performance effectively.

Error and the Seven Stages of Action

In Figure 2.5, it can be seen that mistakes occur at the higher levels of cognition and involve errors in goal-setting, planning, and comparing results with expectations. On the other hand, slips arise at the lower stages and include errors in executing a plan or perceiving and interpreting outcomes. Memory lapses can happen at any stage of transition, leading to the interruption of the action cycle and incomplete desired actions.

Slips happen when actions that are done unconsciously get off track, while mistakes happen when people decide to do something wrong. Sometimes, the same cognitive processes that make users creative and insightful can also lead to mistakes. For example, users may generalize too quickly and classify a new situation as similar to an old one, even when there are significant differences. These mistakes can be hard to spot and fix [7].

Chapter 3

Interaction Analysis

Here will be presented three out of nine examples (the rest in Appendix B, C, D, E, F, G) of the manual analysis of the products using Norman's Seven Stages of action model. This collection of examples help to showcase how a text-book analysis of Norman's model works. But first, here are a few things to consider:

- The analysis depends heavily on the user. For example:
  - The user's actions
  - The user's characteristics
  - The user's technical know-how
  - And other small factors, like the user's preferences and so on
- Sometimes, the worst-case scenario is chosen to show more Gulf of Executions and Gulf of Evaluations.
- When this analysis encounters Gulfs, they will be identified as (A, B): A represents the number of Gulfs of Execution, and B represents the number of Gulfs of Evaluation. This style helps to distinguish and refer to these Gulfs clearly.

3.1 Example 1: Casio

First of all, it is important to describe the product and the user before continuing the examples. This product is called a Casio[1] and is a digital watch (Figure 3.1). It has three buttons that can do certain functions: 'Activate Back-light,' 'Setting the Alarm', 'Stopwatch', 'Changing the Time-Format', and 'Changing the Time'. The user that tries to accomplish the goal does not have a good conceptual model

[1]Specifically, it is the CASIO Vintage A700WEM-7AEF

of the Casio (digital watch). That means that he or she does not own a digital watch and never had to use one.

Goal: The time must be moved forward by one hour.

Plan: The user has some options to reach their goal:

- The user could look at the instruction manual and thus try to change the time that way
- The user could give the digital watch to his uncle, who has experience with digital watches that would change the time for the user
- The user could also give the watch to a watch dealer and ask how to change the time or let him do it for the user
- Or the user could try to change the time by himself, without reading the manual.

The user chooses the last option.

Specify:

1. Find out which button leads to the settings.
2. Press the correct button until 'Time-setting' mode appears, where one can change the time.
3. If the hour indicator is blinking, use another button to increase hours (the user does not know which button increases the hours and does not know if the hours will blink first after getting to 'Time-setting' mode).

Execute:

1. Under the display, the clock has provided a mapping. The mapping is not perfect because the proximity to the buttons is not guaranteed. But still, one knows which description belongs to which button. For the next step, the user excludes the 'Light' button. And must now decide which of the 2 buttons ('Mode', 'Start-Stop / 12-24H') is the right one. The user believes the Mode Button will change between different modes. The user thinks of a stopwatch mode and an alarm mode because 'Alarm' is written on the top-left of the case, above the screen. It could also be that the time setting is also a 'mode'.

Figure 3.1: Timekeeping Mode of the digital watch. The mode button is the under-left button. The light button is the upper-left button. And the right button is for changing the time format.

When the user then looks at the button on the right side, he thinks that this button will start and stop the stopwatch. Also, because of the 'slash' (/), the user believes that this button has two functions. Namely, start-stop the stopwatch and the 12-24H format function. But the user is not sure what is meant by the latter function. It could be that one can change the time with this function, or it could be that it changes the time format. But this is not obvious, and therefore this is a small **Gulf of Execution(1,0)**. Because the user is not sure which button to press and therefore has to guess. The user listens to his gut feeling and chooses the Mode button.

2. After pressing the Mode button once, a sound is heard, and the Alarm menu (Figure 3.2) appears. The time has changed to 0:00, and the date and day are not displayed. At the top, one can see 'AL'. And further up on the top left of the case, 'Alarm' can be seen as a functional description of the clock. The user interprets this display as the alarm menu. This can be a bit difficult to interpret correctly **(Gulf of Evaluation (1,1))**. The user is also sure that the watch has multiple modes because he has the four descriptions above (Alarm, SIG, SPL, Chrono). However, he does not know what SIG, SPL, and Chrono mean.

   - The user presses the Mode button and sees the stopwatch display (Figure 3.3) with 'ST' instead of 'AL', and the seconds are at 0.
   - The user might interpret the 'ST' mode as a stopwatch or time setting menu, which is a small **Gulf of Evaluation(1,2)**.

Figure 3.2: Alarm Mode

Figure 3.3: Stopwatch Mode

- The user presses the Mode button again and enters the Time Setting mode, where the seconds are blinking, indicating that they can be changed.
- However, it is not indicated how to change the seconds. The user assumes that he can change to minutes by pressing the Mode button once more, but the seconds stop blinking, and the clock returns to the initial state, which is a **Gulf of Execution(2,2)** because the user guessed to press the Mode button.

Perceive: Normal time display. The seconds are no longer blinking. Seems like the watch is in 'Timekeeping Mode'. The last step in 'Specify' (change hours) is not reached.

Interpret: The clock is back to the initial state and displays the time. The last press on the Mode button leads to going out of the menu.

Compare: Time still the same. The time change did not work. The mode button does not change to hours/minutes. Press another button next time.

Next Iteration:

Goal: Adjust the time. The time needs to be set one hour ahead.

Plan: Increase the time by 1 hour in the menus. Without a user manual. Through trial and error.

Specify: The user has learned from the previous interaction how many button clicks it takes to enter the time-setting mode and now knows which button not to press when the seconds are blinking.

- Press the Mode button 3 times.
- When the seconds are blinking, press the right-side Button to switch to minutes.
- Press the right-side Button again until the hours start blinking (guessed).
- Use the Light Button to adjust the hours (guessed).

Execute:

- After the first Mode Button click, the alarm menu is displayed. With the second click, the user enters the stopwatch menu. And with the third click, the user finally enters the time-setting mode because the current time is displayed, and the seconds start blinking.

- Since the seconds are blinking, it means that the seconds can be changed. However, the user wants to change the hours, so he presses the right-side Button to switch to minutes/hours. But after pressing the right-side Button, the seconds are reset to 0. Once again, the user has guessed and pressed the wrong button. Therefore, there is a significant **Gulf of Execution(3,2)** here because it was simply not apparent to the user which button to press to switch from setting the seconds to set the minutes/hours. Since the user did not know that, they had to guess which button was the right one. This is the second time guessing in the entire process. However, the seconds continue to blink. This means that the clock is still in 'time-setting mode.'

Perceive: Seconds have been reset to 0. Seconds are still blinking.

Interpret: Seconds have only been reset. Still in time-setting mode.

Compare: Right-side Button has reset the seconds. Goal not achieved. Try using the Light Button next time. The user could also set the seconds correctly in the next iteration but decides against it.

Next iteration:

Goal: Same goal. The time must be moved forward by one hour.

Plan: User tries to change the time by himself, without reading the manual. By trial and error

Specify: The user found out that the clicks he guessed did not do what he wanted. He has already done all the combinations, and now he is missing the last one. Namely, press the Light Button and then press the right-side Button.

- The seconds are still blinking, so the user can directly try pressing the Light Button to make the minutes blink.
- Press the Light Button again to make the hours blink.
- Press the right-side Button to increase the hours, with each press increasing the hours by one.
- Exit the mode by pressing the 'Mode' button.

Execute:

- The user presses the Light Button while the seconds are still flashing, and the clock changes from seconds to hours and then to minutes. This is a small **Gulf of Execution(4,2)** because the user did not expect the hours to flash directly.
- The user can skip the step where the hours start flashing and directly increase the hours by pressing the right-side Button.
- The hours are still blinking after increasing the hours.
- The user exits the mode by pressing the Mode button and returns to the time display. The hours stop flashing, and the clock goes back to its initial state.
- In this iteration, the desired actions from the user were almost the same as the actions allowed by the system, resulting in no Gulf of Execution. The user understood the state of the system through trial and error and did not have to guess which key to press.

Perceiving: The user only had to press the Light Button once for hours to change. By clicking the right-side Button, the hours increased by 1. Hours are still blinking. After pressing Mode Button, hours stop flashing.

Interpret: Light Button has changed the selection to the hours. And the right-side Button has increased hours by 1. By pressing the Mode Button, the clock returns to the initial state.

Compare: Hours increased by 1. Goal reached.

During the entire interaction with the digital clock, a total of **4 Gulf of Executions** occurred and a total of **2 Gulf of Evaluations**.

3.2 Example 2: Bluetooth Speaker

This product is a Bluetooth speaker from Bang&Olufsen. The exact model is called BeoPlay P2. It is a small and portable Bluetooth speaker with a minimalist design. At first sight, it seems that this device has no buttons. It has one Power Button, which is hidden in the backside of the speaker.
The user never used this concrete Bluetooth speaker. He owns a Bluetooth speaker, but it is not the same model. Additionally, the user does not have a manual available for him.

Goal: The user wants to set up the Bluetooth speaker, that is, connect it to his smartphone and play and pause music.

Plan:

1. Ask for guidance from a friend
2. Look on Youtube for videos on how to use the speaker
3. The user tries it himself
4. Look on the manufacturer's website for instructions or a video

The user chooses option 3.

Specify:

- Look for the on/off button
- Switch on the device
- Locate the Bluetooth pairing mode button
- Activate Bluetooth pairing mode
- Take the smartphone and go to Bluetooth settings
- Scan for Bluetooth devices
- Select the speaker and press connect
- Play music on the smartphone
- Find the button to pause music
- Press the button to pause music

- Press the button again to play music

Execute:

- The user looks for prominent buttons on this speaker but finds no buttons. There are no obvious elements to click on. The user does not know what can be done with this device. This is a big **Gulf of Execution(1,0)**. After a long time of fumbling around with the device, he realizes that he can click on the logo. Under the logo is the button. (Figure 3.4)

Figure 3.4: Power Button

- By mistake, the user clicked on the button briefly. There was a short, high-pitched signal tone from the speakers afterward. When he turned the device over and looked at the front side, he saw that one LED started blinking white. This is a small **Gulf of Evaluation(1,1)** because the feedback does not happen on the same side where the user clicked the button, and it is not clear how to interpret the LED. He had to perform another action to perceive the LED feedback. (Figure 3.5) After less than 2 seconds, the LED stops blinking and just lights up. The user interpreted that the device started

Figure 3.5: White LED

booting when it started blinking, and when it only lit white, he interpreted that the device had now finished booting.

- For some reason, the user is pretty sure that it has no other buttons. So he wonders how to get into pairing mode. The user thinks he has 3 options:
  - Long click on the power button for 2-4 seconds
  - Double-click on the power button
  - Click on the power button for a very long time, 5-10 seconds

The user guesses and tries to execute the first option first. This is a **Gulf of Execution(2,1)** because he is not sure which option will lead him to the goal. It is also not described anywhere.

- The user presses the power button for a long time. After pressing it for three seconds, he hears a short beep, and then he turns the device over. There was no vibration. On the front, the user sees that the LED is now blinking blue. (Figure 3.6) The user interprets this blue flashing that the device is in Bluetooth pairing mode. The blue color comes from Bluetooth, and the blinking means to him that the device is available for other devices to pair with.
- The user takes out his smartphone. He goes to the settings and activates Bluetooth on his smartphone.
- After the user has activated his Bluetooth, it starts to search for new devices. This means that the user does not have to click on 'Scan' at all. After a

Figure 3.6: Blue LED indicating the Bluetooth Pairing Mode

few seconds, he sees a device called 'Beoplay P2'. (in Figure 3.7) The user

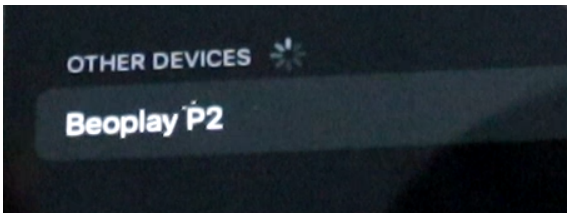


Figure 3.7: Device name in the settings

is not sure if this is his device or another device. His friend did not tell him what the device was called. That is a **Gulf of Evaluation(2,2)**, because the user can not figure out what kind of device 'Beoplay P2' is or if it is a device from another person. Maybe if it was indicated on the settings that this was a Bluetooth speaker, the gulf would be smaller. The user enters the name 'Beoplay P2' in the search bar of his browser. After clicking 'Search', he saw a picture of the same speaker he had in the results.

- Now, the user is sure that Beoplay P2 is his speaker. He selects these speakers on the smartphone by clicking on the name. There will be a hint if one wants to pair with the device. The user clicks on the name of the device and shortly after sees on the smartphone that the device is paired. Then he hears two short tones from the speakers. The LED lights up white, there is no more blue flashing. The user interprets that the device is now connected to the smartphone.

- The user takes out his smartphone and opens the Spotify app, a music streaming service. There he selects a song and plays it. The music is now played on the Bluetooth speaker.

- To pause the music on the speaker, the user does not find any other buttons or keys on the speaker. There is only the power button. This is a **Gulf of Execution(3,2)** because the user now has to deviate from his plan and figure out how to pause the music. He can not see on the device what he can do. Maybe all actions are based on the power button.

- The user decides to press shortly on the power button to stop the music. In fact, the music is stopped, but the LED is extinguished. The user interprets that the device is now turned off. That means the power button is not the pause button. This is a **Gulf of Execution(4,2)** because his actions did not lead to what he wanted. Maybe he has to press the button twice for a short time. When he tried that too, what he wanted did not happen again. Nothing happened, and the music kept playing. This is a **Gulf of Execution(5,2)** because the user's actions did not bring him closer to the goal. The user does not know what to do, so he looks for the instructions for the device. This is another **Gulf of Execution(6,2)** because the user has to take additional steps than he thought. So he searches the internet for a manual of this device. He enters 'pause Beoplay P2 music' in the search bar and goes to the manufacturer's page[2]. He then sees a manual, and when he opens it, he finds out that he has to click two times in the middle of the 'grid' on the case of the device to pause the music (Figure 3.8). The user does just that, and the

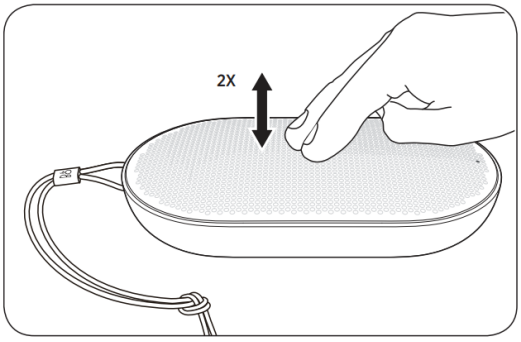


Figure 3.8: Picture from the official manual demonstrating how to start and stop the music.

music stops. That means the center of the 'grid' is sensitive to button presses.

[2]https://www.bang-olufsen.com/en/no/speakers/beoplay-p2

- To play the music again, he double presses the center of the device again. The user now hears the music again. It worked.

Perceive: The user does not perceive where the buttons are placed at first glance and is not even sure if it has buttons. After finding the button, he can start setting it up, and he sees and hears that the device turns on after the first click. The LED in front starts flashing briefly and then lights up white. And later, after that, there is a high-pitched beep (non-speech sound) to tell the user that the device has been turned on. This sound is considered a confirmation that the user has turned on the device. The device responds to Touch on the grid. After the double-click, the music will be paused, and after a double-click again, the music will be played again.

Interpret: The user interprets that the device understands these touch commands, and there may be several such touch gestures for controlling the device.

Compare: The setup has worked. The main controls, such as pause and play, also work, but after looking up the instructions. This touch control was not obvious to the user at first glance.

During the entire interaction with the device, a total of **6 Gulf of Executions** occurred and a total of **2 Gulf of Evaluations**.

3.3 Example 3: PayPal

PayPal is an online platform where users can send and receive money electronically. The user has never used PayPal to send money to someone.

Goal: The user wants to send money to his friend abroad.

Plan: Some options are available:

- Send money via PayPal
- Send money via Western Union at the SBB counter
- Make a bank transfer
- Send Cryptocurrencies

The user chooses the first option. (Note: The example with PayPal has a huge Gulf of Execution, and therefore this example will be taken to show this Gulf).

Specify:

- Open the browser
- Type PayPal in the search bar and search
- Click on the correct PayPal link
- Find the login button and click on it
- Enter the credentials and log in
- Find and click the Send button
- Select the recipient
- Enter the amount
- Click on send
- Ask the friend if the money has arrived

Execute:

- Open the browser

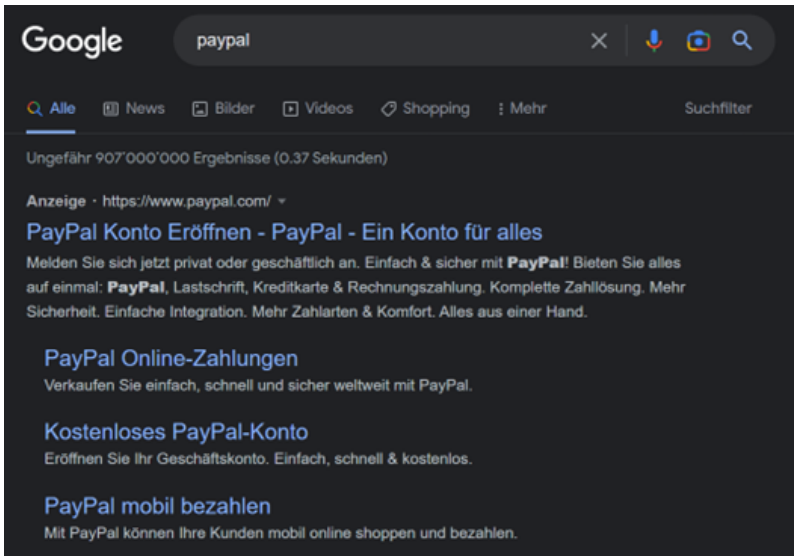


Figure 3.9: Search results

- Type PayPal in the search bar and search
- The user sees that the first link is the right one. Therefore he clicks on it. (Figure 3.9)
- The user now sees the homepage of PayPal (Figure 3.10). To find the login button, he looks to the top right because such buttons are usually located there. He sees the login button, and to the right of this button, there is a 'Sign Up' button with a higher affordance because the background is blue, which is a contrast to the background color of the navigation bar (white). But the user presses the login button because this action leads closer to the goal. (User could click on it because of the high affordance. Instead of 'Login', he could also click on 'Sign Up'. This would have been a Gulf of Execution if the user pressed on it. Because the higher affordance of the "Sign-Up" button might have led to confusion or misled the user into believing it was the right action to take. This would mean there would have been a mismatch between the user's intention and the available options provided by the system).
- The user now sees the login field and enters his credentials (Figure 3.11). Here there is no constraint if one did not enter because the login button is and will not be grayed out if one did not enter anything. After the user has entered his data, he presses Log In.
- Now, the user sees a kind of welcome page with an advertisement in the middle (Figure 3.12). The user does not see a Send button (**Gulf of Execution(1,0)**, because this page/view does not meet his expectations). He expected an

Figure 3.10: Sign Up button has higher affordance than Log In Button

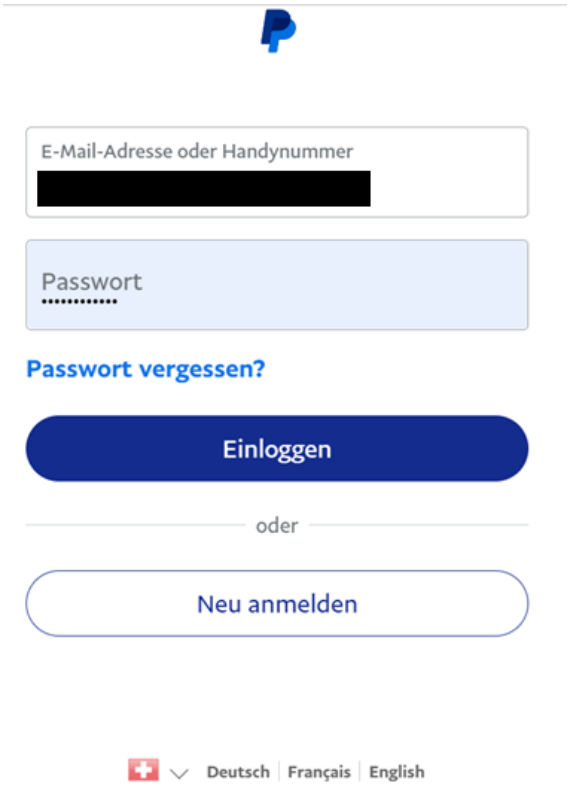


Figure 3.11: Login field

overview page). At the top, next to the logout button, he sees a 'My PayPal' button. He interprets this button as his account, he is logged in, and thinks that the 'Send' button is there. Therefore he clicks on it.

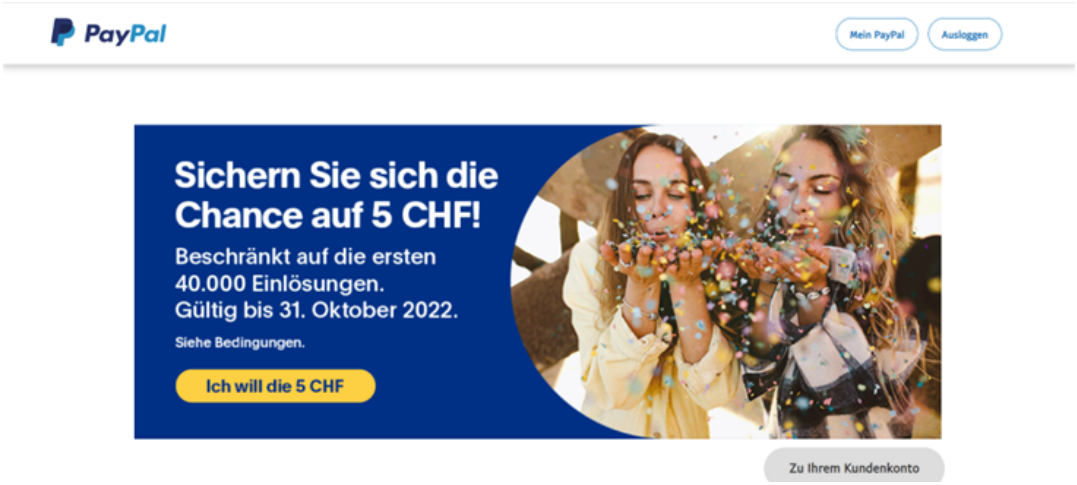


Figure 3.12: User clicks on 'My PayPal'

- The user is now on his account and has a good overview of his account balance and some important functions (Figure 3.13). In this shortcuts area, he finds the 'Send' button. He clicks on it.

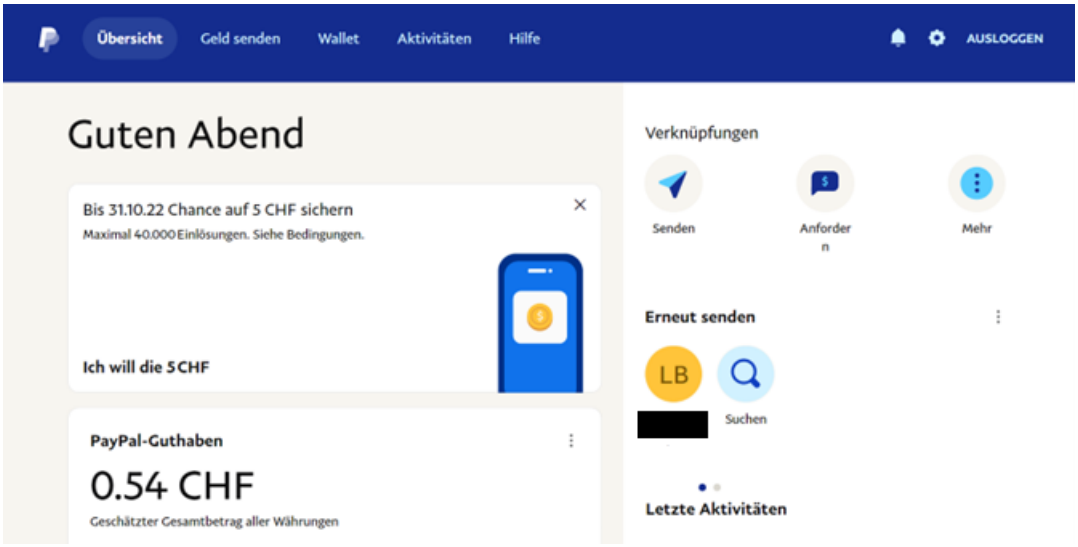


Figure 3.13: User clicks on the 'Send' shortcut

- The user is now in the send function and sees a search bar (Figure 3.14). This search bar has a signifier because, in this search bar, it says 'name, @username, email address or ...'. This helps the user because he now knows what he can enter. The user enters the email address of his friend (Figure

3.15). After that, the 'Next' button lights up, and then clicks on it. (The user needs PayPal for the first time and therefore, he does not know exactly how to interpret this, because he does not understand how to send money via email. (user's mental model not sufficient))

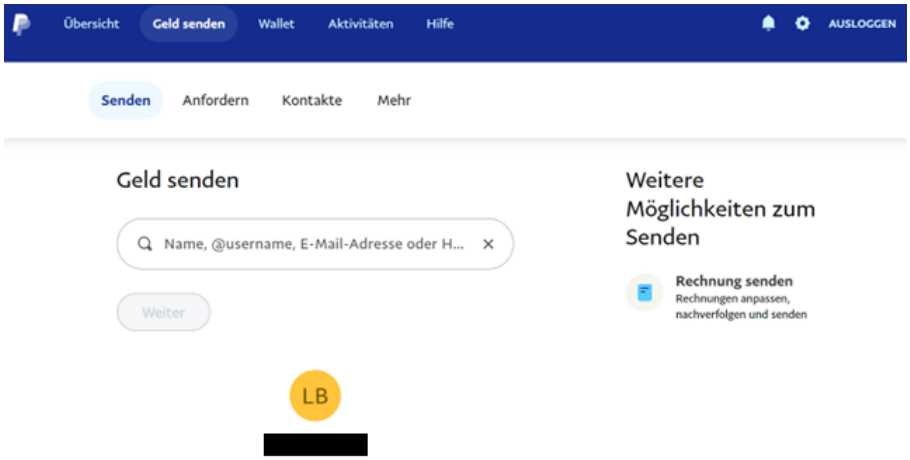


Figure 3.14: Field to enter the information about the receiver

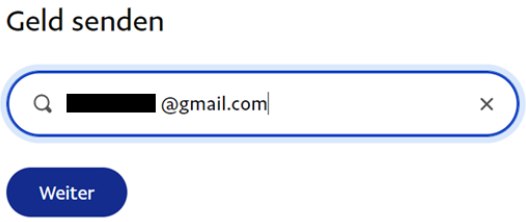


Figure 3.15: After entering the email address, the user clicks on 'Continue'

- The user must now enter the amount (Figure 3.16). The user must be able to send his Euro because he only has Euro on his account.

- At first sight, it is not so obvious where exactly he has to click to change the currency (small **Gulf of Execution(2,0)** because what the user wants is insufficiently communicated by the 'system'. So it is not clear how the user can change the currency). When he moves the mouse over the currencies, he notices how the mouse pointer changes. Namely, the mouse cursor changes to a 'Hand' mouse cursor in Figure 3.18. This is a signifier to let the user know that he can click on this element. The user clicks on the 'CHF'. Three options/selections come up that can be selected (Figure 3.17). The user chooses EUR. After the selection, there is only a text field to enter the amount (Figure 3.19).

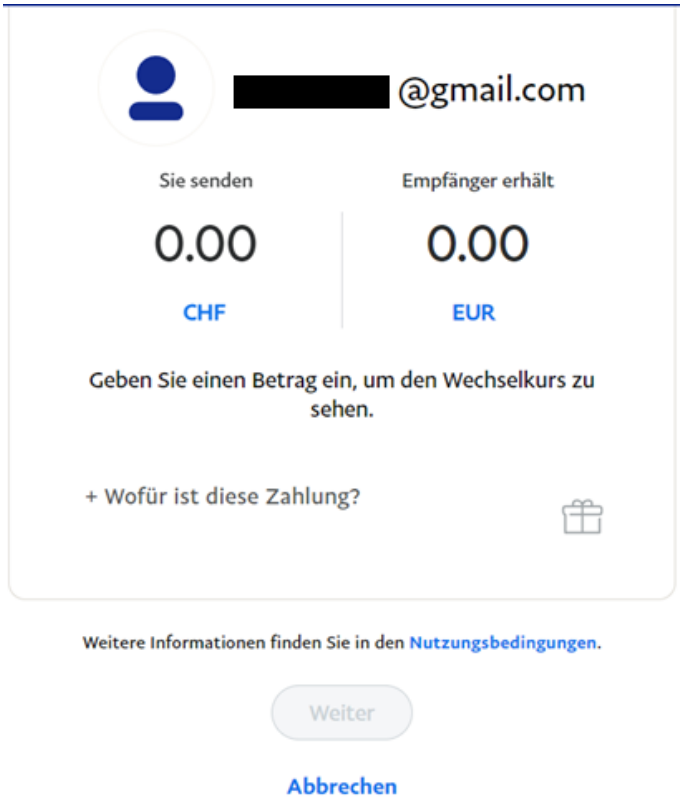


Figure 3.16: User can enter the amount here

Figure 3.17: User can change the currency after clicking on 'CHF'

Figure 3.18: The mouse cursor changes to a Hand cursor signaling that the user can click on it

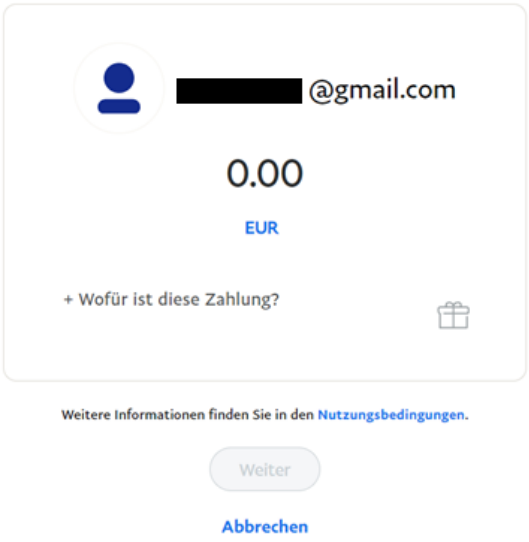


Figure 3.19: User changed to EUR

- The user wants to send all his money. However, he does not find an option to select the maximum amount, which the user expected (**Gulf of Execution(3,0)**), and additionally, nowhere does it show how much money the user has. Since he can not remember his amount, he goes back to the Overview by clicking on the Overview tab. He notes down how much Euros he has and starts the whole process again from the beginning (**Gulf of Evaluation(3,1)**, because the user's balance is not displayed on the page, he/she had to interrupt his actions and thus perform more steps than he/she expected just to figure out how much money he/she has). He enters the amount in EUR now and then clicks on 'Continue', which now lights up (Figure 3.20).

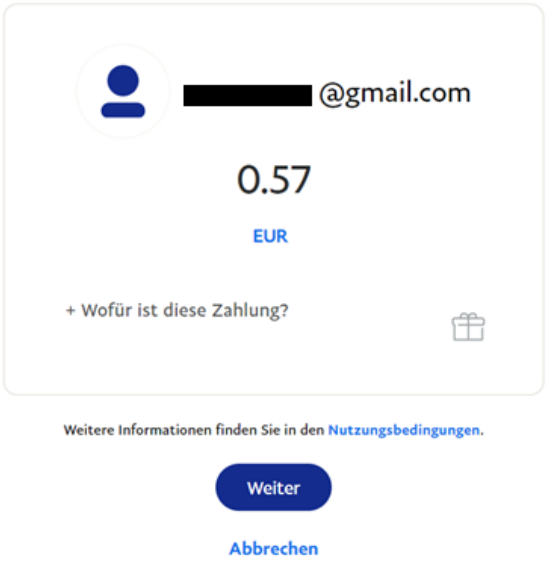
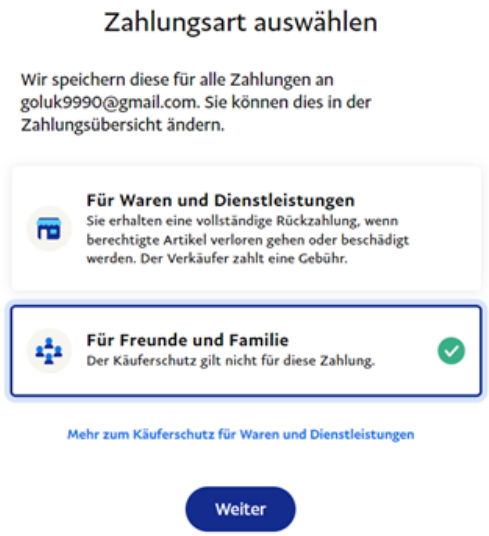


Figure 3.20: User enters maximum balance after going back to the Overview page

- After comes a pop-up where one can choose the payment method. As he sends the money to his friend, he chooses 'For friends and family' (Figure 3.21a). This option is selected, and the selection is also confirmed with a green check mark (Figure 3.21b). The user then clicks 'Next'.
- Now the transaction is summarized (Figure 3.22).
- At the bottom is the 'Send payment now' button. However, this is grayed out and cannot be clicked. This is a constraint that prevents the user from sending the money before a requirement is filled. The missing requirement is not communicated to the user. The mouse cursor changes to the 'Not Available' cursor, when hovering over the 'Send payment' button, which is a signifier that the user cannot perform this action. This is a big **Gulf of Execution(4,1)**. The user does not know why he can not make this payment.

(a) Two Payment Options



(b) User chooses Friends and Family

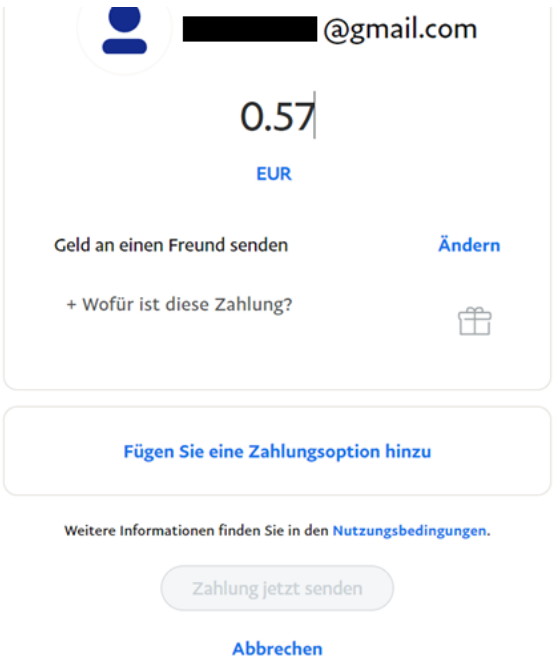


Figure 3.22: User can not continue with the payment

He tries to find out why it did not work (**Gulf of Evaluation(4,2)** because there is big room for interpretation). He thinks that this transaction did not work because he did not write anything in the text field 'What is this payment for'. He writes on this text field: 'Gift'. The button still has not changed (**Gulf of Execution(5,2)** and **Gulf of Evaluation(5,3)**). The last option for him would be to click on 'Add a payment option'. When he clicked on it, a pop-up opens again (Figure 3.23).

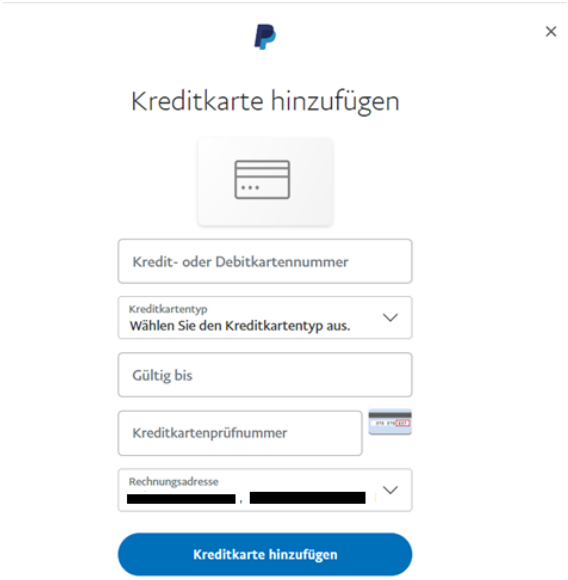


Figure 3.23: Pop-Up for entering credit-card information

- The user is asked to add a credit card. The user does not want to do this, so he clicks on the cross in the upper right corner to close this pop-up (**Gulf of Evaluation(5,4)**, because the user still does not know why he can not make the payment since there is no sufficient feedback). The user then sees his transaction summary again. Since the button is still grayed out, he has to come up with something else or he gives up. The user decides to give up (**big Gulf of Execution(6,4)**).

Perceive: In the last step, the user tries to send the amount. But the button can not be clicked and is grayed out. The system does not give any feedback on why the button does not click.

Interpret: One can interpret many things according to this condition. It could be interpreted that perhaps more money is needed for a transaction, i.e. that there is a minimum amount of money that must be sent for a transaction. Or it could also

be that the recipient does not exist or has entered incorrectly or even does not have a PayPal account. Maybe there are hidden fees that prevent the user from making the payment since the user is sending all of their funds. Or maybe the user had to enable a security option so that he can send all his money.

Compare: Goal not reached. The user could not make the payment for unknown reasons. The user gives up.

During the entire interaction with the website, a total of **6 Gulf of Executions** occurred and a total of **4 Gulf of Evaluations**.

Chapter 4

Questionnaire

The questionnaires that are shown in this chapter are made to identify as many instances of Gulf of Executions and Gulf of Evaluations as possible and to see how it performs against an expert-based evaluation. The goal is to discover at least 50% of the Gulfs that were discovered in the expert-based evaluation. The manual analysis in Chapter 3 will not be in the comparison because this analysis was done without the questionnaire and without participants. More detail about the Methodology is found in Chapter 5. The main idea of this questionnaire is when participants evaluate this questionnaire with products, the found Gulf of Executions and Gulf of Evaluations will be counted, and finally, it will be compared with the Ground Truth.

The questionnaire is a simplified version of Norman's Seven Stages of Action model. The reason for that is that the questionnaire is meant for a quick and easy evaluation of the interaction with a product. The manual analysis with Norman's Seven Stages of Action model is a tedious task compared to the questionnaires. The questionnaires have two phases that make up a cycle: *Execution* and *Evaluation*.
An interaction is a cyclic process where the user first performs actions on the system (Execution phase) and then observes and interprets the system's response (Evaluation phase). During the execution phase, users turn their goals into actions to interact with the system. This means deciding what steps to make and then using the system's interface to carry out those steps. After users have done something, the Evaluation phase starts. Here, users evaluate how the system responded to what they did, seeking feedback from the system. This feedback lets the person know if what they did moved them closer to their goals. They interpret the system's current state and compare it to what they expected. If it turns out that the user did not reach their goal in the evaluation phase, they go back to the Execution phase, change what they did, and try again. Based on their evaluation of the system's feedback, users continue this cyclical process until they feel they have reached their

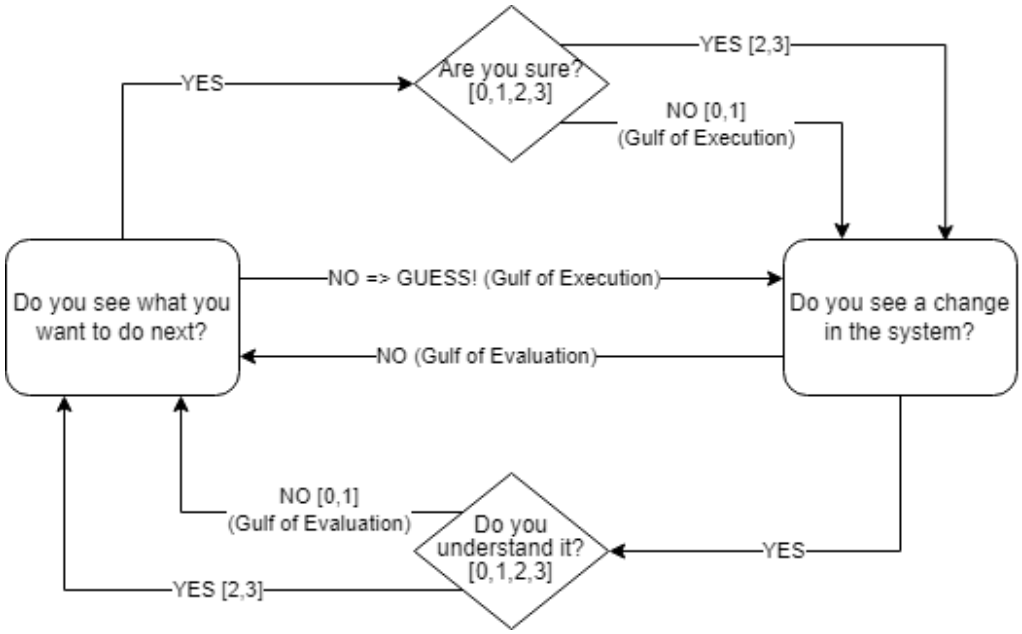


Figure 4.1: First Questionnaire

goals.

For every phase, there are two questions to determine if something is a Gulf of Execution or Gulf of Evaluation. The quality of these questions will determine how well the questionnaires will identify these Gulfs. If one question is too vague or unclear, that could lead to false positives, meaning a Gulf got discovered that should not be a Gulf at all.

The questionnaires were conducted as a think-aloud evaluation, making it easier to see what users think when doing an action. If the user is not sure about an action and has to guess what that action does or what kind of action the user can do, then that would be considered a Gulf of Execution.

The first questionnaire in Figure 4.1 starts from left to right, namely with the question: 'Do you see what you want to do next?'
This question is asked first because the participant has to achieve a goal. Therefore, the user has to think about how to approach this problem/goal and sometimes subconsciously make the planning and specification of the actions. If the user answers this question with a NO, that means that there is a Gulf of Execution because the system does not allow those actions that the user had in mind.

If the answer is YES, the user has to answer the question: 'Are you sure?' This question was designed because the idea is to slow down the user more to make the questionnaire more effective. If the questionnaire does not slow the user down, the user would then rush the interview, and the questionnaire would not bring valuable information as this is a think-aloud evaluation. The user can then choose between 0, 1, 2, and 3 for how sure he/she is about the planned action. If the participants answer this question with a YES or [2,3], then the user can take action. If NO or [0,1], then that would be a Gulf of Execution because the user's intended actions did not clearly match the action possibilities of the system [8].

In both cases, the user must do an action. If the user is not sure, then he/she has to guess a possible action. After the execution of an action, the execution phase of the questionnaire is over, and the user perceives the state of the system and subconsciously interprets the state of the system and even compares if the goal is reached.
So the next question the user has to answer is: 'Do you see changes in the system?' This question aims to find out if the user understands if there was feedback from the system after the action. The user answers with YES if the state of the system changes (that can be a new window, a notification, a sound, and so on).
If the user answers with NO, then according to the questionnaire, that is a Gulf of Evaluation. The idea behind this question is to make it very easy for the user to answer questions and have enough cognition for the task. The assumption here is that no feedback means that the user has a hard time understanding the state of the system, thus – Gulf of Evaluation. But there are exceptions. For example, when a user tries to click on something, not knowing whether it is clickable or not, and after clicking, nothing happens, the user automatically assumes that it is not clickable and thus understands the state of the system even though the system or the state did not change. That is why this question is not optimal for discovering real Gulf of Evaluations, as this would result in false positives. A few examples are presented in Chapter 6.
If the user answers with YES, the next question is, "Do you understand it?". This really lets the user slow down and think about what the state of the system is and if he understands it. There is also a scale from [0,1,2,3]. If the user thinks he understood it, then it is not a Gulf of Evaluation, even though it could be one. That is also a flaw in this questionnaire which will be discussed in Chapter 6. If the user answers it with under a two, which would be a NO, then that would be a Gulf of Evaluation. With that, the Evaluation phase ends, and the cycle continues again from the beginning.

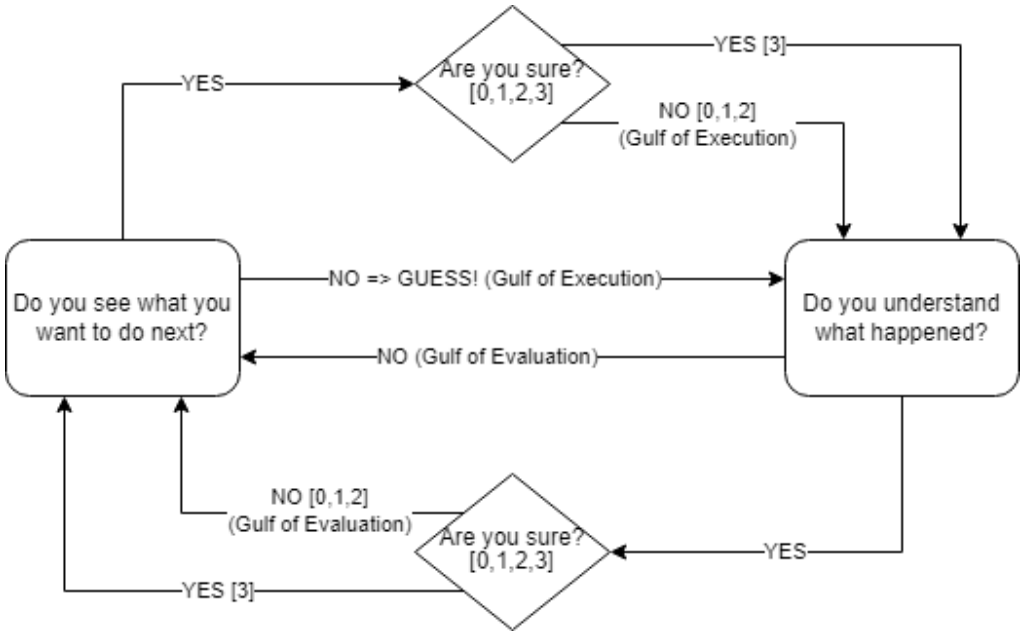


Figure 4.2: Second Questionnaire

The second iteration of the questionnaire in Figure 4.2 is similar to the first one based on the structure. It is still based on the execution and evaluation cycle. The only things that changed are the questions, and the scale was tweaked. The first question, "Do you see what you want to do?" is still the same. The outcomes are still the same. The next question, "Are you sure?" is also the same, but what changed here is the scale of the answer. When the user is even in the slightest unsure, then that would be a Gulf of Execution. So if he answers under a three, that would mean a NO, which would result in a Gulf of Execution. The parameters are changed that way because even the slightest insecurity of the users' actions counts as guessing and, thus – Gulf of Execution. But that thought is not a good indicator for Gulf of Execution. This will be discussed in more detail in Chapter 6 and 7.

After answering the question, the user has to take an action. After taking the action, he will be asked immediately: "Do you understand what happened?". This question differs from the questionnaire's first iteration because seeing if something happened is not a good indicator for the Gulf of Evaluation. The user then has to answer it with a YES or a NO. If it is a NO, then that is a Gulf of Evaluation because the system provided insufficient feedback to show clearly in what state the system is. If the user answers with a YES, the next question will be, "Are you sure?". This question is also different from the first iteration, and the scale has been changed. This question will be for slowing down the user and thinking about

the state of the system more if they really understand it. And here, the scales are 0,1,2 for a NO and 3 for a YES. In that way, there will be a Gulf of Evaluation of the insecurity of the users understanding of the state. And after that, the cycle starts again.

Chapter 5

Methodology

In this chapter, there is a detailed description of how the interviews were conducted and what the process looked like after the interviews.
The goal is to compare the two questionnaires in terms of how many Gulfs they discover. The questionnaire that comes close to the Ground Truth regarding the Gulfs will be the better questionnaire. In this chapter, there is a detailed description of how the Ground Truth is created.

First, it is important to mention again that the participants will be doing a Think Aloud interview using both iterations of the questionnaire. In total, there are nine examples of products that were manually analyzed, and only three of them will be used for the interviews. The three examples are the same as in Chapter 3. Namely 'Casio', 'Bluetooth Speaker', and 'PayPal'. Five participants are getting interviewed with the first iteration of the questionnaire. And another five participants with the second iteration of the questionnaire. Every participant has to evaluate the same three products (Bluetooth Speaker, Casio, PayPal). After the interviews are finished, every evaluation will be analyzed to define a 'Ground Truth'.

The 'Ground Truth' is not the same as the manual analysis with Norman's Seven Stages of Action model in Chapter 3. The Ground Truth is the result of the expert-based evaluation, and that does not mean that the evaluation will only use Norman's Seven Stages of Action model. And also, the manual analysis in Chapter 3 will not be compared to the evaluations with the participants because each participant has different actions, experiences, and preferences. Therefore, comparing the two would not provide significant insights into the quality of the questionnaire.
That is why every evaluation will be analyzed separately. This analysis will result in the **Ground Truth**. The main idea for the Ground Truth is that the interaction will be looked through the eyes of the user. It is possible to create a Ground Truth

as long it is clear why one user does one action. The expert will decide if a situation is a Gulf or not, strictly following the definitions of the two Gulfs. In other words, the Ground Truth will be a result of the expert-based evaluation. For that, the Think Aloud interview was a good fit for this approach because it is a good way to understand their thoughts for every phase of the interaction. That way, it is easier to decide whether something was a Gulf. It is also important to mention that the questionnaire in itself does not influence the Ground Truth. In this analysis, only the users' thoughts and actions will be looked at.

In short, some of the **advantages** of the Think Aloud method are:

- **Qualitative Data:** In-depth understanding of the user's rationale behind an action.
- **Natural Behavior:** Actions and responses of the user closely resemble their usual environment or way of interacting with a system. Realistic capture of how the user would interact with the system in their daily lives.
- **Real-Time Issue Identification:** Instant identification of confusion, frustration and satisfaction. If the user encounters problems, the researchers can immediately ask follow-up questions to gather more details about the problem, sometimes finding the cause of it.

And some of the **disadvantages** are:

- **Cognitive Load:** Putting thoughts into words takes a certain amount of cognitive work. When participants are asked to interact with a system and say what they are thinking at the same time, it might take their attention away from their main task. This could lead to errors, mistakes, unnatural behavior, and user flow interruption.
- **Hawthorne Effect:** The Hawthorne effect is when a person's behavior changes because they know they are being observed. This could cause people to act differently from how they normally do, which could lead to biased results. When people know they are being watched and their actions are being recorded, they might work harder, avoid doing certain actions, or overthink their decisions.

After conducting the Ground Truth analysis for the interactions, all the Gulf of Execution and Gulf of Evaluation from the participants are compiled for comparison. If the participants' evaluation yields similar results to the Ground Truth analysis,

in terms of the Gulfs discovered, it suggests that the questionnaire is effective. An effective questionnaire means that it has a good mapping of the 'expert'-based analysis and the evaluation obtained through the questionnaire. In other words, if the Gulfs obtained from the participants' evaluation are the same as the Ground Truth, then that means the questionnaire is effective. Due to the simplicity of the questionnaire, users can also use it to independently analyze interactions and accurately identify instances of Gulf of Execution and Gulf of Evaluation.

To summarize, the main methodology for these interviews is the Think Aloud method. With that method, a better understanding of the users' thought processes while interacting with the products is gained. However, it is not the perfect method to truly understand what goes on in the heads of the users because "the human is a psychological being engaged in a psychological interaction, which cannot be reduced to that which is concurrently verbalized. Thinking is much more than what can be explicitly expressed in words."[13]. The Think Aloud method is still an adequate method for a Ground Truth analysis. In fact, for detecting more problems, the Think Aloud method is suited more than letting participants work silently [14].

So the interview process looks like this:

- Every participant will be audio recorded. That will be made clear from the beginning. Also, assure them every recording of them will be deleted after the study to reduce anxiety.
- Every audio recording will be transcribed.
- The products Bluetooth and Casio will also be filmed while the users interact with them. Visual cues help to determine if users are confident with their actions.
- The same goes for the PayPal example. That example will be screen recorded.
- Start with Questionnaire 1 (Chapter 4) interview with five participants, each evaluating three products (Bluetooth, Casio, PayPal). The participants had the questionnaire in front of them, and they had to ask themselves the questions on it loudly. This guaranteed that the users were slowed down.
- Count every Gulf of Execution and Gulf of Evaluation for every evaluation with the questionnaire.
- Find the Ground Truth for each evaluation. Compile the results.

- Interview five different participants with the same three products but now on the second iteration of the questionnaire (Chapter 4). The process is like in Questionnaire 1: Let the users ask the questions loudly themselves.
- Count every Gulf of Execution and Gulf of Evaluation.
- Find the Ground Truth. Compile the results.

When the interviews and the Ground Truth analysis are done, the results of the two questionnaires are then compared. The results are shown in Chapter 6.

5.1 Problems in finding the Ground Truth

First of all, finding the Ground Truth is not an exact science. Sometimes assumptions had to be made because either the user did not talk too much or questions were forgotten to be asked. For example, in an interview, the question 'Are you sure?' was forgotten to be asked, and that had a negative effect on a potential Gulf that had to be reflected on. Later on, it was assumed he was not sure because he was not decisive with his decision which was concluded thanks to the video recordings.

In some cases, it is unclear to gather a Ground Truth in the analysis. For example, maybe the users did not talk much and did not describe what they were thinking, and maybe they were not asked enough questions to understand them. To counter that, sometimes the interviewer had to intervene and ask additional questions to get that important information from the user. Sometimes users do not talk too much on purpose because there is a fear to mess up and even embarrassing themselves.

To minimize these problems, the interviewer should make the users comfortable before starting the evaluation, in which they are assured that there are no false actions and that the recordings will be deleted. Additionally, the user should be reminded that they have to say out loud everything they think. Something also important is to slow the user down. Sometimes in the evaluations, this would be a problem where the user would perform multiple actions and perceive multiple things at once. That leads to incomplete information and data to create the Ground Truth. A measure against that would be to intervene after every additional step they make and force the user to tell what they thought in the previous actions.

Chapter 6

Results

In this chapter, the results gathered from the evaluations are examined, and the factors that affected them are discussed. To repeat, the main goal of the thesis is to find out if having a simplified Norman model is a good option to quickly discover at least 50% of the Gulfs found in the manual analysis. The simplified Norman model should be quick and easy to use, even for users that are not knowledgeable in this area.

Another goal is to compare the two questionnaires (Questionnaire 1 in Figure 4.1 and Questionnaire 2 in Figure 4.2) to determine whether the second iteration identifies more Gulfs and fewer false positives. Additionally, the objective is to measure the difference between them.

To give the readers a broad overview, in this chapter, there are three different types of charts that show how many Gulfs the two questionnaires discovered and how they performed comparing it to their Ground Truth in these three example scenarios:

- Bluetooth: Connect the smartphone with the Bluetooth speakers.
- Casio: Advancing the time on the Casio digital watch by one hour.
- PayPal: Sending the maximum amount of money to a friend abroad.

The first types of charts in Section 6.1 show a broad overview of the discovered Gulf of Execution and Gulf of Evaluation from the three scenarios above. These charts indicate the True Positives, the False Positives, and the Ground Truth for a broad comparison between the two questionnaires.

The second type of chart in Section 6.2 is the Bubble Chart. This chart shows every Gulf that got discovered in a questionnaire. But the important point here is that every problem got categorized by its severity (from 1 to 3) and by the number of users that discovered one particular problem (from 1 to 5). The best case would be that all of the 5 users would discover all problems with severity 3. Unfortunately, that is not always the case, and for some important Gulfs, there is an explanation provided why not every user discovered it.

Furthermore, the plot has been assigned a 'crucial area' where all the sub-optimal cases are located. In this context, 'sub-optimal' refers to cases that have both received a high severity rating and have been discovered by only a few users.
In addition, the graph of the Gulfs is divided into two separate graphs: one displaying only the Gulf of Executions and another showing only the Gulf of Evaluations. The idea here is to identify differences or common grounds between the two gulfs.

The last type of graph in Section 6.3 shows what kind of problems were discovered the most. For each questionnaire, only the five most discovered problems are chosen to find out if the same problems arrive in both questionnaires and to find out which questionnaire has the most problems discovered by everyone.

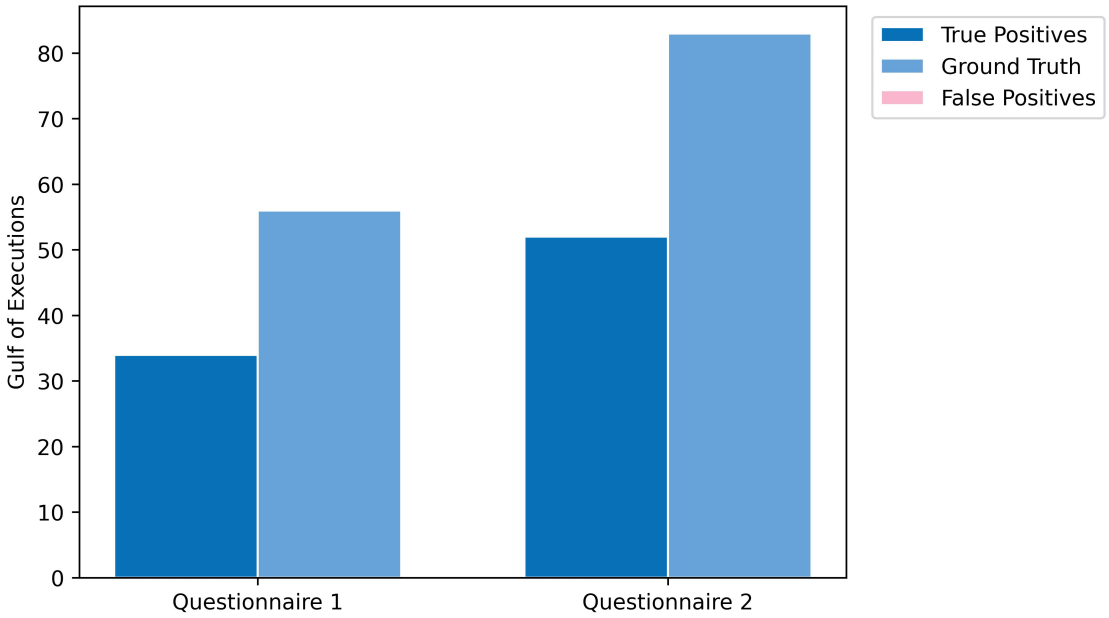


Figure 6.1: Unique Gulf of Executions with Ground Truths from both questionnaires. The Ground Truth of Questionnaire 2 seems to have more Gulf of Executions than the Ground Truth from Questionnaire 1

6.1 Comparison of Unique Gulfs

In Figure 6.1, it looks like Questionnaire 2 (Q2) performs better than Questionnaire 1 (Q1) because it discovers more Gulf of Executions (True Positives). The first questionnaire discovers only 34 Gulf of Executions, and the second one discovers 52 Gulf of Executions. But comparing the True Positives from each questionnaire to their respective Ground Truths reveals that both questionnaires also miss a lot of Gulf of Executions.

To conclude which questionnaire is better, the accuracy in Table 6.1 has to be looked at. It is worth noting that there are more Gulf of Executions in the Ground Truth of Q2 compared to Q1. This can be attributed to the fact that users take longer to achieve their goals in Q2. Users spending more time naturally leads to more actions being performed. These additional actions can result in a Gulf of Execution.

One reason could be that users in the Questionnaire 2 group were more adventurous and more likely to try more actions to reach the goal. The increased number of Gulfs of Execution in Q2's Ground Truth could be a result of users' willingness to explore alternative paths and engage in more actions during the task.
Referring to Table 6.1, Questionnaire 2 is slightly better with its 63% accuracy than

Questionnaire	True Positives	Ground Truth	Accuracy
Q1	34	56	61%
Q2	52	83	63%

Table 6.1: Discovery of Gulf of Executions and their accuracy

Questionnaire	True Positives	Ground Truth	Accuracy
Q1	23	40	57.5%
Q2	25	41	61%

Table 6.2: Discovery of Gulf of Evaluations and their accuracy

Questionnaire 1 with 61%.

It seems as if both of these questionnaires are a reliable way to discover real Gulfs, but it is difficult to draw definite conclusions about the superiority of one questionnaire over the other. It is also worth noting that the questionnaires miss a lot of the Gulfs when compared to the Ground Truth (Light-blue bars in Figure 6.1 and Figure 6.2). The question here should be why there is this big difference between the Gulf of Execution from the questionnaires and the Ground Truths. One possibility is that users in the Questionnaire 2 group were more adventurous and more likely to try more actions to reach the goal. Because if someone does more actions, they will eventually encounter more problems and, therefore, more Gulfs.

In Figure 6.2, it is evident that the number of True Positives in Gulf of Evaluations from Questionnaires 1 and 2 are close. The only difference here is that in Questionnaire 1, there are two Gulf of Evaluations that are classified as False Positives. Questionnaire 1 falsely discovers 2 problems as Gulf of Evaluations. That means it classifies problems as a Gulf of Evaluation even though they should not be. This is most likely a flaw in the questionnaire because the questions were not clear enough. There are some examples that showcase the unclear questions in Section 6.2.1 and 6.2.2.

Furthermore, Table 6.2 illustrates that Questionnaire 2 got slightly better accuracy. The difference of 3.5 percentage points may not be statistically significant in determining which one is better, but it is still a good sign that Questionnaire 2 has no false positives.

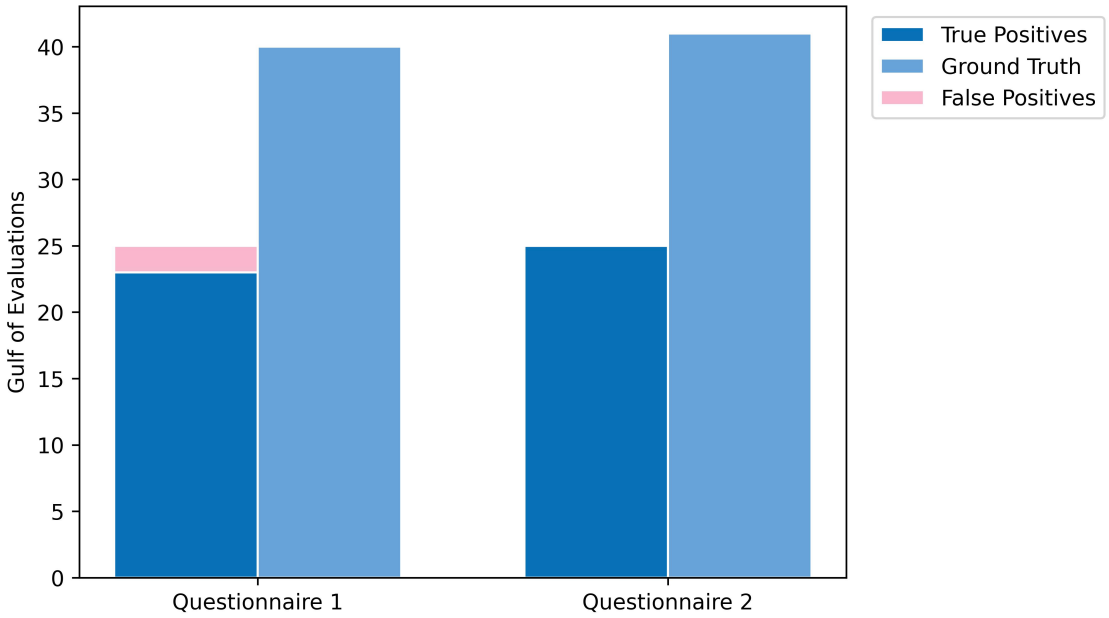


Figure 6.2: Unique Gulf of Evaluations with Ground Truths from both questionnaires. There are two false positive Gulf of Evaluations from Questionnaire 1

6.1.1 Investigating the False Positives

To understand the two false positives in Q1, it is important to analyze what happened in those two cases. In the Casio scenario, a user encounters a problem while trying to change the time. After clicking the 'Mode' Button several times, the user lands in the 'Time Setting' Mode. The seconds were blinking, which means the user could modify the seconds. But because he has to change the hours, he has to switch to a mode where he can change the hours. That is why the user pushed the 'Mode' Button again to switch the mode from seconds to minutes and then hours.

However, after clicking the 'Mode' Button, the blinking stops and puts the watch in the 'Time-Telling' Mode. When asked if the user understood what happened, the user said "NO". According to Questionnaire 1, this is classified as a Gulf of Evaluation, but technically it is not because the user understands the state of the system (that the watch is in Time-Telling Mode). However, the user does not understand *why* the blinking stopped, which is not directly a Gulf of Evaluation.

In the Bluetooth scenario, where the goal is to connect the Bluetooth device with the smartphone, a user is unsure if their device is connected and tries playing a song to test it. After hearing the sound from their smartphone, he realizes the device is still not connected. However, when asked if he sees a change in the system, he responds with "NO" on the questionnaire, which is classified as a Gulf of Evaluation.

According to Don Norman's definition, though, a Gulf of Evaluation refers to the difficulty of assessing the state of the system and how well the artifact supports the discovery and interpretation of that state [7]. In this case, the state of the system was instantly known after hearing the song from the phone, so it is not a Gulf of Evaluation and therefore is considered a False Positive.

Unclear questions were responsible for the False Positives observed in the two examples discussed. Specifically, the questions "Do you understand it?" and "Do you see a change in the system?" were partially responsible. The second question is particularly problematic because it assumes that a lack of change will result in a Gulf of Evaluation, which may not always be the case. In the Bluetooth example, the question is difficult to interpret, leading to different responses. Additionally, users tend to respond "NO" to the question "Did you understand it?" when something did not happen exactly as expected, possibly due to a different understanding of the question. These examples highlight the need for clear and unambiguous questions in order to classify Gulfs correctly.

6.2 Categorizing and Visualizing Severity of Problems

While maintaining the list of all the problems in the interaction, an effort was made to categorize every problem into three grades of severity:

- Low (1): Relatively minor issues that should not impact the user's ability to complete tasks.

- Medium (2): These are problems that have a noticeable impact on the user's ability to complete tasks. These also have a negative impact on the user's experience. It can sometimes cause low to medium frustration.

- High (3): Problems with high severity are connected with negative emotions and even great frustration. These have a significant impact on the user's experience and the ability to complete a task. Sometimes this can lead to users giving up on tasks.

The approach to categorizing the users' problems is a subjective task, as definitions are vague, and different people may classify them differently. However, two plots were created to show how many of the five users discovered a problem with a particular severity level. If all five users discover the problem, the bubble is on the rightmost side. If there are many cases where a problem with a certain severity level got discovered by a certain amount of users, the corresponding bubble in the plot would be larger.

For example, the large bubble on the left in Figure 6.3a indicates that there were many cases where only one user experienced a low-severity problem. On the other hand, the small bubble in the middle represents only a few cases where a problem with severity level two was experienced by three users with the help of Questionnaire 1.

This plot is useful in demonstrating how many problems with high severity are being discovered, as not every user encounters all the problems being addressed. Some users may not experience these problems due to taking different actions.
The goal is to find out the number of cases within a questionnaire that is in this 'crucial area', represented by the hatched region. Afterward, the two questionnaires will be compared based on the number of cases in this crucial area. Although a direct comparison would not be fair as both questionnaires have different numbers

of gulfs, it would be beneficial to look at the ratio of cases in the crucial area to the total number of gulfs in the questionnaire.

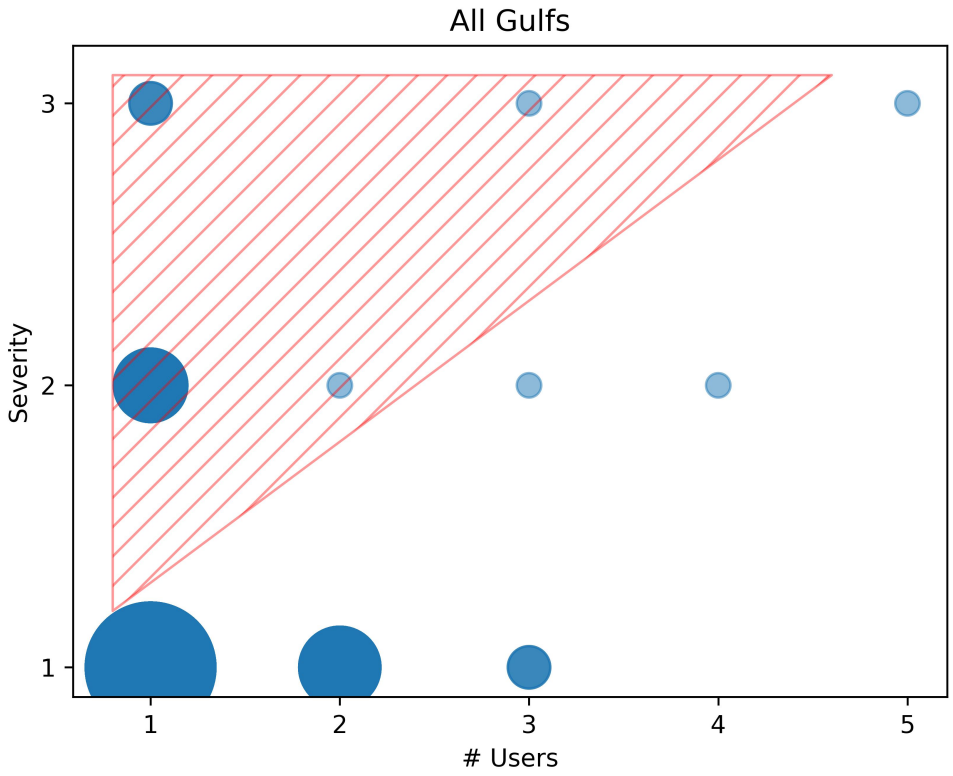
6.2.1 Questionnaire 1

The bubble graph displaying all the problems in Questionnaire 1 can be seen in Figure 6.3a. The biggest bubble is in the under-left corner. This would mean that there are a lot of cases where a low-severity problem gets discovered by only one user. This is not a big deal because of the nature of the interviews, where users have a lot of freedom; it can happen that they do not take the same action and therefore do not encounter the same problems. Note that there are no entries of zero users in this chart because the cases included here are only the ones that the participants discovered. Therefore, there are no gulfs in this chart from the Ground Truth.

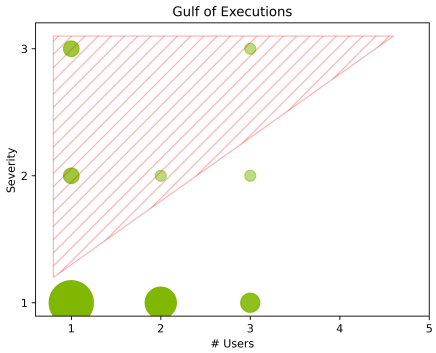
Regardless of that, the focus is now on the problems in the crucial area of the plot. The crucial area is the upper triangular region marked by red hatched lines within the chart. Since in the crucial area, the cases are severe, and not every user discovered it. Hence it is necessary to talk about the crucial problems and to analyze why everyone did not discover it. And to see if they would actually encounter the problem if they performed different actions.

The chart in Figure 6.3a got split by the type of Gulf as it can be seen in Figure 6.3b for the Gulf of Executions and in Figure 6.3c for the Gulf of Evaluations. The idea is to potentially see important insights about the differences in their frequency and to see which Gulf has more cases in this crucial area.

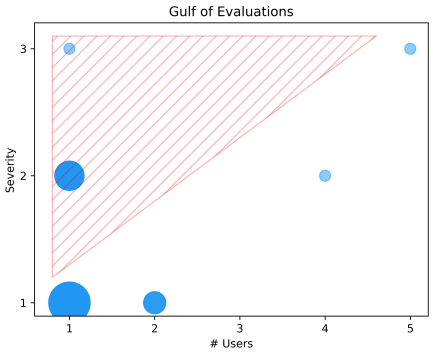
Both Gulfs have many cases where only one user discovered a problem with the severity 1. The only big difference here is that there were Gulf of Evaluation cases, where four and five users discovered them. Whereas in the Gulf of Execution, there was no such case. It can be concluded that the questionnaire made more users discover particular Gulf of Evaluation cases. If the crucial area is looked at and the cases there are counted, then the Gulf of Evaluation cases (eight) are higher than the Gulf of Execution cases (six) even though there are more Gulf of Executions than Gulf of Evaluations (Table 6.1 and Table 6.2). This can be interpreted that Questionnaire 1 failed to let many users discover crucial Gulf of Evaluations.



(a) Bubble Plot for Questionnaire 1 which shows how many cases there are where a specific amount of users discovered a problem with a specific severity. If the bubble is big, that means the frequency of these cases is higher



(b) Bubble Plot of Gulf of Executions



(c) Bubble Plot of Gulf of Evaluations

Figure 6.3: Bubble Plot for Questionnaire 1

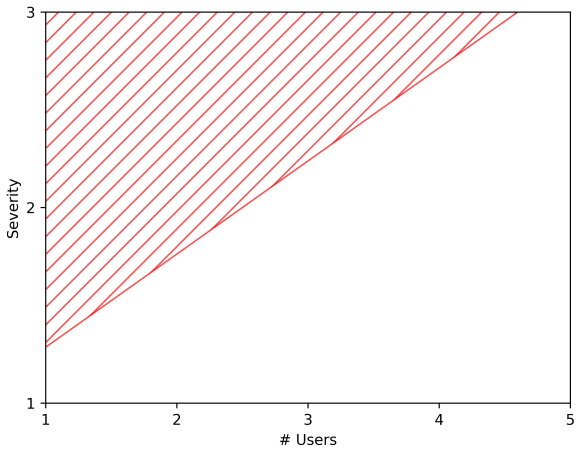


Figure 6.4: All the cases in this area (hatched in red) are going to be examined in detail

Firstly let us look at problems that lead to Gulf of Executions (#Users, Severity):

1. Bluetooth (1,3): The user gives up.

    - Only one user discovered this crucial 'problem'. The fact that only one user discovered this crucial problem suggests that the questionnaire may have performed poorly. But it is not the case here. Of the five users, only one could not manage to connect the Bluetooth device. That is why he gave up.

2. PayPal (1,3): User wanted to google and was not sure what to do next

    - This problem has a high severity because seeking help from Google was his last resort as he did not know how to proceed next. In a way, it is like giving up. This problem could be summarized and combined with the next problem. Again this is not a problem with the questionnaire because only this particular person wanted to get help from Google, and the others just gave up.

3. PayPal (3,3): Does not see what to do next and gives up

    - If the question 'Why did not everyone discover this problem' is asked, it should also have to be looked at the problem above. In its essence, those two questions are almost the same in a way. It is not the questionnaire's fault that this problem did not get discovered by everyone. The fifth user is an exception because he made it work but with the option 'Waren und Dienstleistungen', and that is why that did not count as giving up.

4. Casio (1,2): The user had to guess between MODE or Right Button. (Guessed MODE)

    - This problem could also be summarized with 'User unsure if MODE Button will help'. Then four users would have encountered the same problem. The fifth user did not have this problem because he/she was really sure what to click, and that is why the user did not have to guess.

5. Casio (1,2): Unsure if Right Button will help

    - Users guessing the Right Button is a common thing, but the intention and the order of the actions is an important context for this problem that seemingly got discovered by only one person. This problem was formulated a bit too general. Here the user's first action was to click on the Right Button, thinking this action would bring him closer to the goal. He had three buttons to choose from and guessed the Right Button. There are also other similar problems like 'Unsure if clicking Right Button will add hours' and 'Unsure if clicking the Right Button will change to minutes/hours'. If this problem would be summarized with the others to one general problem like 'User guessed Right Button', then every user would have encountered the problem, and this would be (5,2)

6. PayPal (2,2): Does not know what to do next

    - In the PayPal example, no one got it right, and everyone gave up. So it is surprising why this kind of problem only got discovered once. This problem should be distinguished by the problems where the users give up. This is not the case here because, in this problem, the user got frustrated after something did not work out.

Now the problems that lead to the Gulf of Evaluations:

1. Bluetooth (1,2): Does not see the Power Button

    - If someone would look at the device, no one would see instantly where the Power Button is. That is why this is interesting that only one user discovered it. In a closer look, the user's next action is to look at the device to spot the Power Button. When the user did not see one, he answered the question "Do you see a change in the system?" with "NO". That is why according to Questionnaire 1, this problem is a Gulf of

Evaluation. The user just could not understand why there was no Power Button. This problem is also in the Ground Truth, so it is a real Gulf of Evaluation and, therefore, a True Positive. For other users, that problem was a Gulf of Execution. That could mean that some questions could be understood differently by other users. Considering the other users, it is clear that they have problems like 'Guesses where to press the Power Button', which means just looking at the device to see where to press. In that case, the user knew what to click, and therefore, in this case, that was not a Gulf of Evaluation.

2. Bluetooth (1,2): No change in systems state after pairing failed

    - After the pairing failed, the phone showed a notification that the pairing failed. That was feedback that the user received and understood. The user did not understand why the pairing did not work. And because the pairing did not work, he thought that the state was the same since the device still was not connected. So for the user, that meant that the state of the device did not change, which is not fully true. So in a way, the state changed, but his view was a bit broader than expected from the question.

3. Bluetooth (1,2): Does not understand why it still needs an app after seeing Pop-Up

    - Although the user downloaded the app thinking that is enough to connect, he went to the Bluetooth settings. He clicked on the device. But still, the same Pop-Up appeared. The understanding of the state was not aligned with the system state itself. The user had to accept Geo-Locations on the app and tried to connect from there. But the user thought that installing the app was enough to evade that Pop-Up. The only reason that only one user discovered this problem is that only this person tried to connect it via the app. All the other users tried to do it 'manually'.

4. Bluetooth (1,2): No feedback after clicking on accept Geo-Location

    - The user could not understand the state of the device after clicking on 'Accept' because, after the click, nothing happened. There was no way for the user to know what happened and why it did not work. Only this user did discover this problem because he was the only one that tried to connect it via the app.

5. Bluetooth (1,2): Does not understand why the app could not pair (App could not find device)

    - After going to the app to connect the device from there, the app could not find the device, although the device was on. Therefore the user does not understand why the app could not find the device, and hence the user does not understand the state of the system. Only one user discovered this problem because he was the only one that tried to connect it via the app.

6. PayPal (1,2): Does not understand the situation with a bank account and PayPal credits

    - The user does not understand if the money was sent and if it was deducted from the PayPal credits or from the bank account. There is a lot of confusion because she does not understand why she still can not send the money. For her, there is no way to know the state of the system. This was a specific issue that stemmed from the lack of understanding of the system, which only got 'discovered' by one user. Other users sometimes also did not understand the system, but they never had this specific confusion about bank accounts and PayPal credits.

7. PayPal (1,2): Still not clear what state the system is in

    - After sending the money did not work with 'Friends and Family', he tried to do it with 'Waren und Dienstleistungen'. That change somehow worked, but it was not the goal of this task. So the user is still stuck on why it did not work with 'Friends and Family'. Therefore the user did not fully understand the state of the system. Like above, there are many cases where users did not understand the system, but in this specific problem, only one user 'discovered' it.

8. PayPal (1,3): Does not know why 'Code anfordern' did not work. Does not understand state.

    - After clicking on 'Code anfordern' and writing the code down, and pressing Enter, PayPal asks again for requesting the code again. No feedback on the previous action. For the user, it is difficult to understand the system. This specific problem only happened to one user, surprisingly, which is why naturally, only one user discovered it.

As observable from this examination, it does not give the readers a good indication of the quality of the questionnaire. It really depends on the individual problems. If the problems were defined too detailed, that would result in too many problems that get discovered by only one person. And that would seem like it is a problem of the questionnaire. But it is possible that the questionnaire might not be sufficient enough in handling different user mental models and the varying interpretations of the system's feedback. For instance, it might not be enough to ask if users noticed a change in the system. Maybe it would be better to ask more specific questions about the user's understanding of what exactly changed and why.

6.2.2 Questionnaire 2

Referring to Figure 6.5a, it becomes apparent that the plot for Questionnaire 2 (Q2) has similarities to the first one (Figure 6.3a). There is a prominent huge bubble in (1,1), which means that there are still many cases where a single user discovers a problem with severity 1. However, in this graph, it is more prominent than the one from Q1. One key difference is that there are some cases with severity 3 that got discovered by everyone in Q2. This is a good indicator that this questionnaire managed to make the users discover these critical problems. However, it is important to mention that there are other participants involved in evaluating these products (Bluetooth speaker, Casio digital watch, PayPal) with the second questionnaire. Therefore, different users with different mental models and experiences skew the results. That is why a direct comparison of these charts of both questionnaires is not insightful. The evaluation is influenced by the user's experience of the product, which is a limitation that must be taken into account.

The chart in Figure 6.5a got split into two different charts, which show the Gulf of Execution cases in Figure 6.5b and the Gulf of Evaluation cases in Figure 6.5c. Considering the cases in the crucial area, it is evident that there are fewer Gulf of Executions in the crucial area than Gulf of Evaluations. These cases have to be examined closely before jumping to conclusions. But if we compare them both, it is clear that at first glance, the questionnaire performs better in detecting crucial Gulf of Executions. There are many cases that got discovered by multiple users. On the other hand, for the Gulf of Evaluation, it can not be said the same. There is not a single case that got detected by all five of the users. Examining the cases in the 'crucial area' of the chart[1] and counting them can give insightful information about a questionnaire's quality and ability to detect crucial problems. That examination

[1]See 6.4 for a visualization of the 'crucial area'

will be shown later in this chapter in the Section 'Summary'.

However, if these bubble plots of the two questionnaires are compared with the bare eye, it can not be determined which one was better, mainly because these plots do not give the readers a sense of the scale of the bubbles. For example, it is not clear how many cases there are in the case (1,1). And later on in this chapter, it will be shown that not every case outside of the hatched upper triangular region does mean that the questionnaire is faulty. The cases in these areas will be examined: (1,3), (1,2), (2,2), (2,3). It will be seen why these problems did not get discovered as much as they should be. One key point in this examination is if the other users had the chance to discover it with different actions.

First, let us look at the cases that were Gulf of Executions:

1. Bluetooth (1,2): The user had to guess where to press for Power Button.

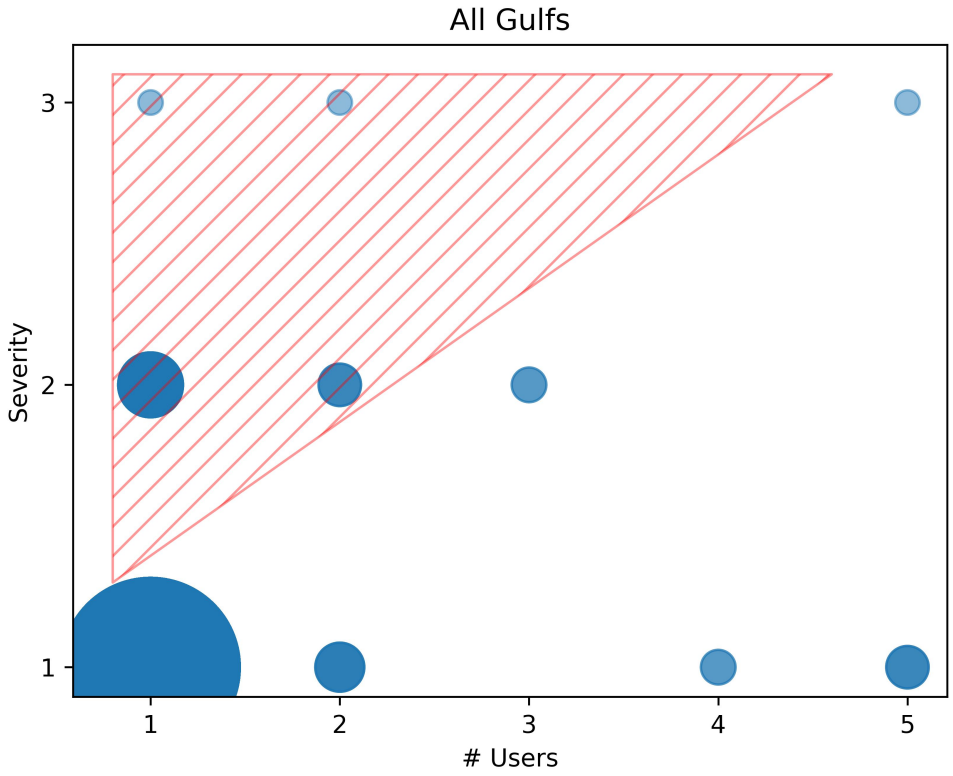
    - The four other users had the chance to get this problem, but they did not answer the questionnaire the same way because they were sure where to press it. That is why it did not count as a Gulf of Execution for these users. Some problems are similar, though. For example, 'User is not sure if the Button is the Power Button' is really similar to this problem, but the key difference is that in this problem, the user is not sure what action to take specifically. And for the other problem, the user is not sure if the button is the right one, which is associated with lower uncertainty.

2. Bluetooth (1,2): The user is not sure what to do next.

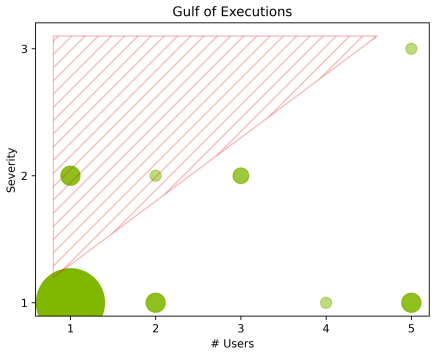
    - Almost every user had uncertainty about a specific action, but they would quickly guess an action so they would not encounter this problem. That is why only one user would encounter this problem. The user that discovered this problem was asked if she would know what to do next. She would answer 'NO' because, for her next action, she was unsure.

3. Bluetooth (3,2): User is not sure if the Button is the Power Button

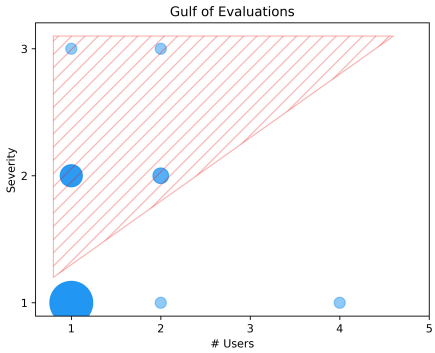
    - Because the button is not sufficiently labeled, there are some users that had trouble finding out if that button is really the Power Button. Two of the users were sure that that one was the Power Button from earlier experience or through logical conclusions.



(a) Bubble Plot for Questionnaire 2, which shows how many cases there are where a specific amount of users discovered a problem with a specific severity. If the bubble is big, that means the frequency of these cases is higher. For example, the most frequent case is at (1,1). Meaning there are many cases of only one person discovering a problem with severity 1.



(b) Bubble Plot of Gulf of Executions



(c) Bubble Plot of Gulf of Evaluations

Figure 6.5: Bubble Plots for Questionnaire 2

Now let us look at the cases that were Gulf of Evaluations:

1. Bluetooth (2,2): Not sure if the device is On

    - One user, after managing to turn the device on and then off, was not sure if the device was On (It was OFF). That is because she did not realize the LED of the device. She did not realize that the LED turned on and then turned off. The other user also managed to turn the device on and off because after clicking on the button, the device did not instantly open, so he pressed it the second time to turn it on, but then the device turned off. The culprit of this problem was that the LED Light was not in the same area as the Power Button and the impatience of the user to click the Button multiple times and then not knowing in what state the device was in. The remaining users did not discover this problem because after clicking on the Power Button, they realized the LED, so they did not press it again, and they would know that the device was on.

2. Bluetooth (1,2): User does not understand the state after not finding the device on the Bluetooth settings

    - Our user that discovered this problem was not sure about the state of the device, so that is why there is a Gulf of Evaluation there. But she was not the only person that did not find the device in the Bluetooth settings. The other users directly understood that the device was not On or not in Pairing Mode. That is why this questionnaire did not discover this problem in the other users.

3. Bluetooth (1,2): Does not understand in which state the device is in

    - This problem happens often, but in this questionnaire and in this example, only one user discovered it. The remaining users did not discover that particular problem, but they did discover similar problems.

4. Bluetooth (2,2): Not sure if the device is the real one

    - Here, this user chose to go to the Bluetooth settings and scanned for devices. The feedback was a list of available devices. He chose a name from the list. But the user is not sure if it is the real device. That can also be classified as a Gulf of Execution since the user has to guess whether it is real.

5. Casio (1,2): Not sure what the state is

- There are other similar problems as this one, but this one is the most generalized one. That is why only one user 'found' this problem because the other similar problems were more specific than this one. Hence these problems were spread out, and it looks like only a few users discovered it. But these problems can also be the result of a bad or insufficient mental model coupled with a bad product design.

6. Casio (1,2): The user does not understand what happened. Thinks she broke something

    - That problem happened after the user clicked on the wrong button, so she went back to the previous state. She thought she knew the steps she had taken to go there where she was last time. She was so sure what actions she had to take, but those actions were wrong, and now she thinks something is broken or it is not the same even though nothing has changed. This only happened to one user.

7. PayPal (2,3): Does not understand why the 'Pay' Button is grayed out

    - This is a crucial problem to reach the goal. So the frustration is really high because it does not work, and there is no feedback on why it does not and how to solve it. Almost nobody could send the money. That is why it is interesting that only two users could discover it.

8. PayPal (1,3): Does not understand why it does not work after trying to send with 'Gift'

    - This user was the only user in this questionnaire that tried to solve the grayed-out Pay Button problem with the Gift-Symbol. Unsurprisingly it did not work out.

Summary of the Severe Cases Comparison

The number of Gulf of Execution cases in the 'crucial area' (in Figure 6.4) differs between Questionnaire 1 (Q1) and Questionnaire 2 (Q2). Specifically, Q1 has six cases, while Q2 has only three. For Gulf of Evaluation, both have eight cases each in the 'crucial area'. If all these cases are counted, then Q1 has the most cases in the 'crucial area' with 14 cases. To get a better sense of the quality of the questionnaires, it is beneficial to compare these numbers with their total number of gulfs. Table 6.3 illustrates the ratios for each questionnaire. The ratio for Q1 is 24.56%, which means that almost one-quarter of all the gulfs lies in the

Questionnaire	Cases in 'Crucial Area'	All Gulfs	Ratio
Questionnaire 1	14	57	24.56%
Questionnaire 2	11	77	14.29%

Table 6.3: For Questionnaire 1, 24.56% of the problems are in the 'crucial area'. On the other hand, Questionnaire 2 has a better ratio of 14.29%

crucial area. Unfortunately, this is not a favorable number for the questionnaire's quality. However, Questionnaire 2 got a better result. With only 14.29%, it beats Questionnaire 1, and that means, in general, that there were fewer cases of high severity where only a few users discovered them.

Problem	Description	Gulf	Users	Questionnaire
P1	Not sure what the problem is with Gray Button	Evaluation	5	Q1
P2	Button is still gray. No change in the system	Evaluation	4	Q1
P3	Does not see what to do next, gives up	Execution	3	Q1
P4	Guesses where to click the Power Button	Execution	3	Q1
P5	Unsure if MODE Button will help	Execution	3	Q1
P3	Not sure what to do next - Gives up	Execution	5	Q2
P5	Not sure if clicking on MODE Button will help	Execution	5	Q2
P6	Not sure if long click on Power Button will work	Execution	5	Q2
P7	Not sure if clicking on LIGHT Button will help	Execution	5	Q2
P8	Not sure how to interpret feedback from value entering	Evaluation	4	Q2

Table 6.4: Table of Top 5 problems for each questionnaire (sorted by 'Questionnaire')

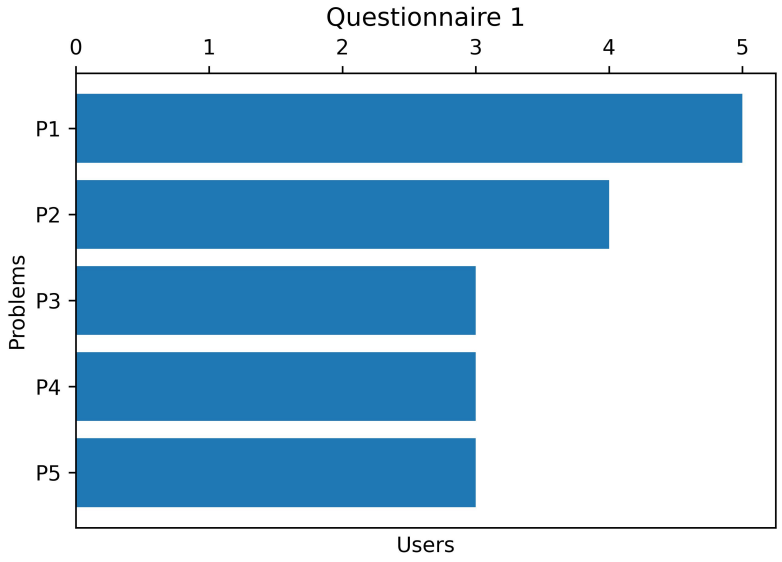
6.3 Top Five Problems

In this section, the focus is going to be on the five most discovered problems for each questionnaire to see if there are interesting differences. There is a table (Table 6.4) compiled where every problem is described, which Gulf it is, how many users discovered it, and from which questionnaire it comes.

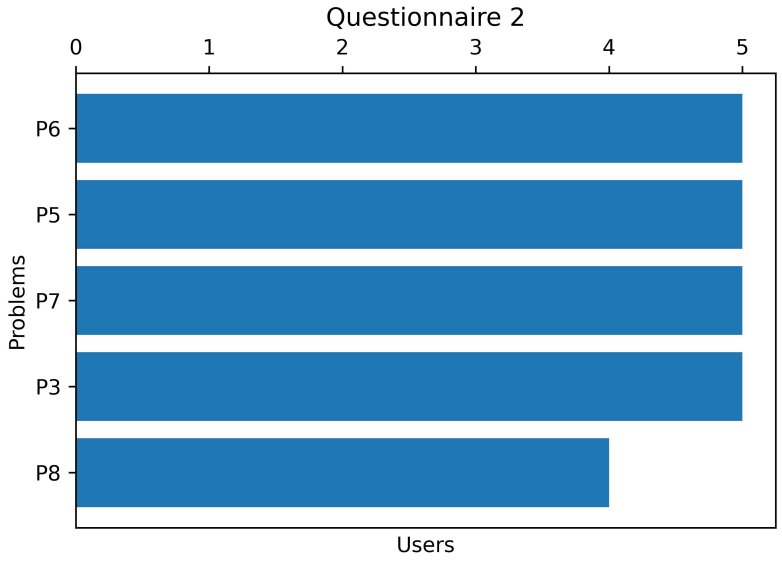
With those five problems for each questionnaire, the goal is to find out if there are differences or common grounds between them. There are two plots generated for each questionnaire with their most discovered five problems. One problem can be either a Gulf of Execution or a Gulf of Evaluation.

In Figure 6.6a, it is observable that only one problem got discovered by all of the five users in Questionnaire 1. Compared to Figure 6.6b, there are more problems

discovered by all five users. Problems P3 and P5 also appear in Questionnaire 1, but it only got discovered by three users compared to the five users in Questionnaire 2.



(a) Questionnaire 1: Only one problem that got discovered by all five users



(b) Questionnaire 2: Four problems got discovered by all five users

Figure 6.6: Comparison of the discovered problems in Questionnaire 1 and Questionnaire 2

It can be concluded that Questionnaire 2 has a slight advantage in identifying these problems. Let us look at P3 and P5 again in more detail and find out if Questionnaire 1 did miss these problems or if there was a chance for the user, with

the help of the questionnaire, to discover it.

- P3: Does not see what to do next, gives up
  - This problem appears in the PayPal example; two users did not experience this problem. Let us say there are User A and User B.

    User A: This user did make it work to send the money, so the questionnaire ended there. Technically it was not really finished because he chose the 'Waren und Dienstleistungen', which was not the right way. If 'Waren und Dienstleistungen' did not work, there would be a high probability this user would not see what to do next and would give up.

    User B: After not reaching the goal, the user wanted to search on Google why it does not work. The interview did finish there because that was similar to giving up and seeking external help, which is not a bad thing. But technically, this could have also been a problem where the user 'gives up'.

- P5: Unsure if MODE Button will help
  - There are two users that did not discover the problem.

    User A: Casio 'Had to guess between MODE or Right Button. Chooses MODE' is the same as 'Unsure if MODE Button will help'

    User B: The user was confident about what the button did. He knew to change the time. He had to click on the MODE Button. He also knew that by clicking on MODE, the watch would cycle between Modes. He even knew not to click on MODE to increase the hours. His understanding of the watch is really good, and his critical thinking helped him to manage to reach the goal without any big problems. He even managed to quit the time setting mode by clicking the MODE Button.

Chapter 7

Discussions

In this chapter, the gathered results are discussed and interpreted related to the thesis research question.

Both of the questionnaires did a good job of discovering the Gulf of Executions and the Gulf of Evaluations. The expected goal was that the questionnaires should at least discover 50% of the Gulfs of the Ground Truth analysis. In Figure 6.1 and Figure 6.2, we can see that both these questionnaires had an accuracy of over 50%. Unfortunately, it is still not quite the same as a manual analysis with the Norman Seven Stages of Action Model, and the first questionnaire even has a few false positives.

However, the time efficiency of the questionnaire method, which typically takes between three to eleven minutes per evaluation, is an interesting observation considering the manual analysis of the same product using Norman's Seven Stage of Action model often takes approximately one hour. Also interesting, in some cases, the shorter questionnaire-based evaluation revealed more Gulfs than the longer manual analysis. However, this is not an entirely fair comparison because these evaluations depend on the individual participant and their unique mental model of the product. By comparing the number of identified Gulfs in the appendix (Appendix B to G) with those from the manual analysis in Chapter 3, it appears that the manual approach often discovers fewer Gulfs. This raises questions: why does Norman's Seven Stages of Action Model sometimes identify fewer Gulfs than participant evaluations, and does this difference matter? First, let us look at some examples:

- User A discovered in the Casio example with the Questionnaire 1 (Q1): **3 Gulf of Executions** and **1 Gulf of Evaluation**

  - In the Ground Truth analysis, there were also 3 Gulf of Executions and 1 Gulf of Evaluation

- User B discovered in the same example and with the same questionnaire (Q1): 4 Gulf of Executions and 1 Gulf of Evaluation

  - The Ground Truth analysis concluded: **5 Gulf of Executions** and **3 Gulf of Evaluations**

If we look at the Casio example in Chapter 3 it can be seen that the amount of Gulfs (4 Gulf of Execution and 2 Gulf of Evaluation) is almost the same as user A. The important thing to note here is that for User B, even though he got more Gulfs than the manual analysis, the questionnaire still missed some of these Gulfs if looked at the Ground Truth.
One important factor is the user itself. With their different mental models, some of them took different actions than others, and that could also lead to different Gulfs. Whereas in the manual analysis in Chapter 3, an imaginary user was used that took different action strings than other users.
So should product designers and developers consider the manual analysis with Norman's Seven Stages of Action model to usability-test their products?

It could be argued that doing an evaluation with the questionnaire, more precisely Questionnaire 2, as it is slightly more accurate than Questionnaire 1, would be more beneficial. There are multiple reasons:

- With Normans Seven Stages of Action Model, it is not possible to discover every possible Gulf in an interaction

- Designers would have to spend hours testing every possible action sequence.

- Their evaluation would even be biased if they did the analysis themselves. As they programmed or designed their product, they have the same mental model as the conceptual model of the designers (Chapter 2.

- Instead, with an evaluation of Questionnaire 2 and some users, the most important Gulfs would be discovered

- If the designers wish to have a more comprehensive analysis, they could do the Ground Truth analysis in some or all of the users' evaluations. As is known that Questionnaire 2 leaves approximately 40% of the Gulfs undiscovered.

In summary, the designers would discover more unique Gulfs in a short amount of time when using Questionnaire 2 instead of doing the manual analysis with Norman's Model.

Comparing Questionnaire 1 and Questionnaire 2

Questionnaire 2 did not bring the improvements that were expected. It detected the same amount of Gulf of Evaluations as in Questionnaire 1. But to the credit of Q2, it had **no False Positives**. Still, both had the same weakness of having significantly fewer Gulf of Executions than the Ground Truth. One possible cause of that would be that some users were overconfident. In the execution phase, they are sure that their actions are correct, which in reality, sometimes, it is not. When they found out that their intentions did not line up with what the system allowed them to do, then according to the questionnaires, it was not a Gulf of Execution, even though it should be one. Because every time they answered that they were sure about one action, then that would mean that a Gulf of Execution would be missed and the execution phase of the questionnaire would be over. Because of this design choice or choice of these questions, the two questionnaires missed many Gulf of Executions.

Also, some of these users never used some of these products. Therefore they had to explore to find out what the product could do.
In a flow of exploring, it is hard to know if something is a Gulf of Execution or a Gulf of Evaluation because the definitions are not that clear-cut. The definition, in other words, is: Trying to figure out how to use the product. If users struggle, then that is a Gulf of Execution. But there is no threshold that says if the user struggles this hard, then it is a Gulf of Execution. If he struggles less than that, then it is not a Gulf of Execution. That is the difficulty.

Implications of the Results

These results imply that the questionnaires are a good way to save time on the analysis of interaction and, at the same time, also cover a good amount of Gulfs. But there is still room for improvement: The questions for the Gulf of Executions can be optimized to discover more Gulf of Executions. The same thing goes for the Gulf of Evaluation. But there is a limit to that. That is because the users or participants do not have all the necessary knowledge of the product. And the Ground Truth sometimes assumes a Gulf of Execution when the user did choose the

wrong action. And there is no way that the user can know if the action is wrong, meaning the intentions of the users and what the systems allow them to do, do not match.
There was also this idea to create a questionnaire for developers in the scope of this work to test their product in a simple and fast manner. But it turns out that these questionnaires would be rendered useless because the developers that build their own products are too confident. Meaning they have the same or similar mental model as the conceptual model of the developers. They already know which actions take where and do not have to guess, or at least rarely. Thus, reducing the Gulf of Executions. The same way goes for having fewer Gulf of Evaluations because they do not really need feedback because they know what will happen. And they always know in what state the system is in.

Evaluating the Think-Aloud Method

The Think-Aloud method for this type of evaluation is a good fit. The goal of these evaluations with the questionnaires was to discover Gulfs ultimately. To do that, the users had to talk about every thought while interacting to better see the discrepancies between the mental models of the users and the conceptual model of the developers. In that way, it is more clear if something is a Gulf or not. The downside of the Think-Aloud method is that it slows users down while interacting. It can have an additional cognitive load on the user, and that can limit the user in achieving his goal. Technically this could lead to more Gulfs, which is not investigated.
Another problem also is when users want to use the questionnaire for themselves, they would not strictly follow the questionnaire as they are sometimes in a form of flow that is not easy to break. There are a lot of examples in these interviews where they just were doing multiple action sequences at once without explaining why.
It had to be made sure that the users should be slowed down. There were some measures that were taken to slow them down. For example:

- Let them ask the questions themselves. That way, they subconsciously have in mind that there will be questions and, therefore, would not skip them.

- Have the questionnaire always in front of them. That helps to train the user on the structure of the questionnaire.

- Intervening sometimes when users take multiple actions at once. For example, questioning the users' previous actions.

All these measures helped to slow the user down. Still, there were still some exceptions where the user would not slow down and even do multiple steps at once.
One possible cause for that would be that the user feels embarrassed to explain their thoughts and artificially tries to feel smart when doing multiple steps.
Nevertheless, the evaluations still discovered many important Gulfs and interesting insights into how different users behave differently when interacting with a product. These evaluations, combined with the Ground Truth analysis, result in a really helpful tool to discover usability problems in a fast manner.

Chapter 8

Conclusions and Future Work

In this final chapter, all of the pieces of the study are put together to make a detailed summary of the findings. The central research question of 'Can the questionnaire discover 50% of the Gulfs in an interaction' has been addressed, contributing to the existing body of knowledge in HCI. This chapter serves as a reflection on the study. It highlights key findings, evaluates methods, and discusses possible future research directions.

The answer to the research questions is 'Yes'. It discovers more than 50% of the Gulfs, which can be seen in Table 6.1 and 6.2. If the time benefits are also included, then the questionnaire is a viable option to discover crucial problems in a product quickly. That is achieved through simplifying the Normans Seven Stages of Action model.

As discussed in Chapter 7 there are still drawbacks to using the questionnaire as it also misses many Gulfs. This could be fixed by: slowing the users more, finding better questions, expanding the questionnaire, or intervening more. Some of these potential solutions also have their downsides.
For example, if the questionnaire were expanded with more questions, then the user would be potentially fatigued during the interaction as it is repetitive. The Think Aloud method would hit its limits as the user would talk less, and the given information would be less insightful. But it would be interesting to see how much different the results would be in future research.
Slowing users more and intervening more would distract the user, and the natural flow of interaction would be interrupted. So having a good questionnaire is a matter of finding the right mix.

Developers that want to quickly see if the usability of their product is good can use the questionnaire and discover the most important gulfs. If they want an in-depth analysis, it is also possible to do the Ground Truth analysis, to cover almost all Gulfs in combination with the evaluations of the participants. Compared to only manually doing the Normans Model analysis and only discovering a few Gulfs, it is more beneficial to evaluate the product with the questionnaire and the Ground Truth analysis.

Additionally, in Chapter 3 and in Appendix (B, D, C, F, E, G) there are nine examples provided of the Interaction Analysis using Normans Seven Stages of Action. In the literature, there are not many examples that use this model to analyze the interaction. And the analysis that is out there does not have complex examples. So for HCI researchers or students, that is a great resource to learn how to apply Norman's Seven Stages of Action model to analyze an interaction between human and computer.
As the readers can see in these examples, even big tech companies like PayPal can have some crucial usability problems that still did not get discovered (as of writing this thesis). With the use of the questionnaire, this could have been easily detected. But of course, it still depends on the user and on the goal, what Gulfs, and how many it can discover.

Some of the limitations in the evaluations of the interaction with the questionnaires are:

- Users' past experience with the product, leading to a varying number of Gulfs. The number of actions taken varies.
- Users' mental model of the product
- The goal of the interaction. Clear and specific goals lead to fewer Gulfs.
- Users' willingness to explore and experiment.
- Overconfidence, leading to a lot of missed Gulfs. (Chapter 7)
- Think Aloud method can put a cognitive load on the users and lead to unwanted interaction errors.
- In the Ground Truth analysis, some Gulfs are subjective since the Think Aloud method was not really effective on some users. They would not talk too much; therefore, it was hard to tell sometimes if something was a Gulf or not. That is why some Gulfs are a result of reasonable assumptions.

- Full freedom on the interaction leads to many different action paths and unique gulfs. Direct comparisons were not always possible. Every Gulf was slightly different, so to assign similar Gulfs was subjective, and the plots in Chapter 7 could look slightly different when someone else evaluated these results.
- All the participants were students from diverse fields of study.

Unfortunately, there are some limitations that can not be changed or evaded. There is a huge gap between the Gulfs that got discovered by the questionnaire and the Gulfs that are in the Ground Truth. It is the advantage and disadvantages of this open way of the questionnaire. It is open in a way that users have total freedom how to reach the goal. That is why there are so many unique Gulfs (Figure 6.1 and Figure 6.2).

Some suggestions for future research:

- Try to come up with better questions for the questionnaire that discovers more Gulfs and does not interrupt the user's action flow.
- Expand the questionnaire to detect more Gulfs as Questionnaire 2 (Chapter 4) without overwhelming the participants in terms of cognitive load and time.
- Find a new way to incorporate developers into the analysis without needing participants to discover Gulfs.
- It would also be interesting to do a full in-depth analysis of a product with interactions of every possible action sequence to discover all Gulfs possible.
- Incorporating different user groups in terms of age groups and backgrounds.
- Try to come up with a different simplification of the Normans Seven Stages of Action Model or even a new model.

In closing, this study has provided valuable insights into the effectiveness of the questionnaire approach while also providing detailed examples of the Normans Seven Stages of Action Model.

Appendix A

Appendix: Short Overview

The remaining examples of the manual analysis will be provided in the appendix. Three of them are in English and can be found in Chapter 3.

- Note all the examples are in German. This is because they were analyzed and documented before the decision was made to write the thesis in English.
- Gulfs are labeled as (A, B). A represents the number of Gulf of Executions, and B represents the number of Gulf of Evaluations.
- **Appendix B:** Setting up an analog camera and taking a picture
- **Appendix C:** Downloading a folder from the Ilias website, a learning management system (LMS)
- **Appendix D:** Transferring pictures from a Canon camera via the 'Camera Connect' app from Canon
- **Appendix E:** Creating a Crypto Wallet with Metamask
- **Appendix F:** Swapping from one Crypto Token to another
- **Appendix G:** Checking the payment status from a Zalando order

Appendix B

Analog Camera

Analogkamera einrichten und erstes Bild schiessen

Der User hat noch nie eine Analogkamera gebraucht und weiss nicht, wie es genau funktioniert.

1. Ziel: Der User will seine neue Analogkamera einrichten und ein erstes Bild schiessen
2. Planen: Der User hat mehrere Optionen:
    a. Auf Youtube ein Tutorial nachschauen, wie man die Kamera einrichtet
    b. Auf der Webseite des Herstellers nachschauen, ob es eine Anleitung gibt
    c. Die Gebrauchsanleitung lesen, die in der Verpackung liegt
    d. Seinen Vater geben zum Einrichten, denn er hatte Erfahrungen mit Analogkameras
    e. Selbst ausprobieren und wenn er nicht weiterkommt, die Gebrauchsanleitung lesen.

    Der User wählt die letzte Option

3. Spezifizieren:
    a. Verpackung öffnen
    b. Kamera herausnehmen
    c. Batteriesteckplätze finden
    d. Batterien einstecken
    e. Herausfinden wie man Gehäuse öffnet
    f. Gehäuse öffnen
    g. Herausfinden wie man Film hineinsteckt
    h. Film einstecken
    i. Gehäuse zumachen
    j. Dann auf den Shutter-Button klicken, um ein Bild zu schiessen

4. Ausführen:
    a. Verpackung öffnen
    b. Den «Behälter» nehmen mit den Anleitungen und dann die Kamera herausnehmen
    c. Ein bisschen anschauen wo die Batteriesteckplätze sind. Der User findet sie unterhalb der Kamera, die sehr gut beschriftet/angedeutet wird, dass es sich um Batteriesteckplätze handelt.

    Abbildung 1

    Es wird sogar auch die Polen der Batterien richtig angezeigt, was ein Signifier ist, bevor man den Steckplatz aufmacht. (Abbildung 1)
    d. Der ‘Handle’ der Steckplätze ist ein guter Affordance dass sich in vielen Produkten eingebaut ist. Man muss mit einem Finger nur diesen ‘Handle’ hineindrücken, um diesen Steckplatz zu öffnen. Der User macht die Steckplätze auf und steckt eine AA-

Batterie rein mit der richtigen Position. Der User wusste aber nicht, dass die Batterie nicht notwendig ist, um Bilder zu machen, sondern die Batterie ist eigentlich für den Blitz benötigt. (Mentales Modell entspricht nicht dem System-Image)

e. Das Gehäuse für den Film befindet sich hinter den Linsen und der User findet links vom Gehäuse einen Button. Dieser Button hat eine hohe Affordance das man den nach unten ziehen muss. Ausserdem hat es einen Signifier, denn der Button hat auch einen Pfeil, der nach unten zeigt. (Abbildung 2)

Abbildung 2

f. Der User zieht den Button nach unten und erwartet das die Klappe sich öffnet. Es tut sich nichts. Das ist ein kleiner **Gulf of Execution(1,0),** weil nichts passiert ist und ausserdem ist es ein kleiner **Gulf of Evaluation(1,1),** weil seine Erwartungen nicht erfüllt ist und er nicht weiss wieso. Es gab ungenügend Feedback. Entweder hat er zu schwach am Button gezogen oder er muss die Klappe öffnen, während er am Button zieht. Der User probiert ein bisschen stärker nach unten zu ziehen. Als dann auch nichts passiert, was wieder ein **Gulf of Execution(2,1)** ist (und **Gulf of Evaluation(2,2),** denn User weiss immer noch nicht, wo das Problem liegt. Es gibt kein gutes Feedback.), entscheidet sich der User die Klappe dann zu öffnen, während man den Button nach unten zieht. Der User struggeled am Anfang, denn es fühlt sich nicht natürlich an (ungewohnt) gleichzeitig mit einer Hand ein Button nach unten zu ziehen und gleichzeitig mit der anderen Hand die Klappe irgendwie zu öffnen, denn es hat keine Markierungen oder, wo man mit den Fingern einfacher festhalten und die Klappe öffnen kann. Dennoch lässt sich die Klappe dann aber öffnen. Der User sieht dann, dass es doch Markierungen hat an der Klappe, die wie «Haken» aussehen (Abbildung 3). Diese 2 Markierungen an der Klappe sind auch Signifier aber ein bisschen diskreter, zum Zeigen, dass sich dort Halterungen befinden, die die Klappe festhält. Beim ersten Mal schauen wäre der User nicht sicher gewesen, was es bedeutet.

Abbildung 3

g. Der User sieht links eine grosse Öffnung und von oben hat es wie eine Art Halterung oder Hindernis. In der Mitte sieht er die 3 Linsen und ein bisschen oben dran hat es eine Art Rad. Und rechts sieht er eine Öffnung mit einem schmalen Zylinder, der unten auch mit diesen «Zähnen» bestückt ist. Der User glaubt, dass der Film links hineingesteckt werden muss, da es dort am meisten Platz hat.

h. Der User nimmt ein Film hervor. Der Film ist der Kodak Gold 200 35mm. Der User merkt, dass der Film nicht Platz hat rechts und deshalb schliesst daraus, dass der Film links hineinmuss. Der Film hat oben einen 'Bump', der dann verhindert, dass es oben im Gerät nicht Platz hat. Der User probiert dann mit 'roher Gewalt' den Film hineinzustecken, aber es passt nicht rein **(Gulf of Execution(3,2),** denn seine Absicht wurde vom System nicht erfüllt. Die Rolle passte in diesem Zustand nicht rein). Der User merkt aber, dass die Halterung nach oben gesetzt werden kann, so dass es mehr Platz hat in der Öffnung. Auch wenn die Halterung oben ist, hat es immer noch kein Platz (**Gulf of Execution(4,2)**). Der User merkt, dass der Bump vom Film immer noch kein Platz hat (**Gulf of Evaluation(4,3)**). Der User merkt dann als letztes, dass unten eine kleine Öffnung für den Bump hat. Das ist ein Constraint, damit der Film in der richtigen Richtung hineingesteckt wird. Deshalb dreht er den Film um und der Film hat nun Platz und geht rein. Der User hatte die ganze Zeit den Film falsch herumgehabt. Die Halterung ist noch oben und der User weiss nicht, ob er es so lassen soll. Er ist so unsicher, dass er direkt in der Anleitung nachschaut, ob die Halterung heruntergeklappt werden soll (Abbildung 4). (**Gulf of Evaluation(4,4),** denn in diesem Moment ist das 'System' für den User nicht verständlich und deshalb will der User diese Kluft mit einer Anleitung überbrücken)

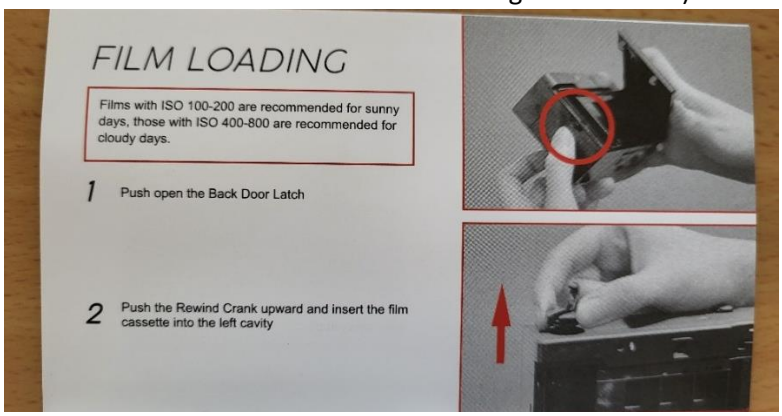


Abbildung 4

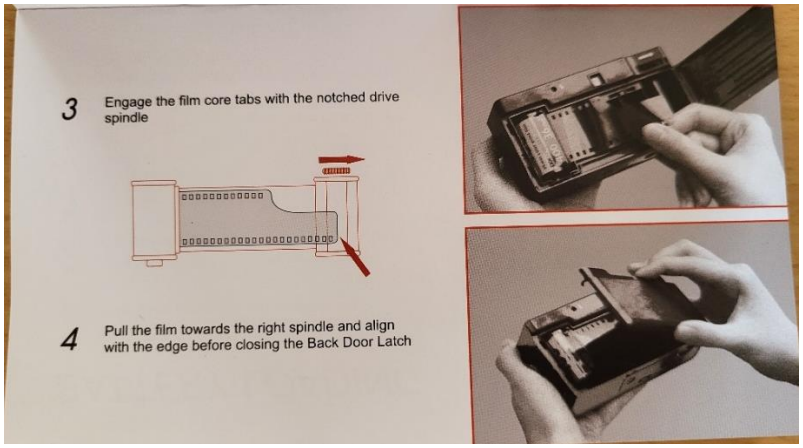


Abbildung 5

Da sieht er, dass bei Schritt 2 der User den Rewind Crank aufklappen sollte, aber bei weiteren Schritten wird nicht erwähnt, ob sie dann wieder heruntergeklappt werden sollte (Abbildung 4). Das ist ein **Gulf of Execution(5,4),** denn der User weiss nicht genau wie er weiter vorangehen soll und diese Brücke ist nicht genügend überbrückt. Er muss raten und entscheidet den Rewind Crank trotzdem herunterzuklappen, denn im letzten Bild ist der Rewind Crank unten (**Gulf of Execution(6,4),** denn der User war sich nicht sicher und musste raten). Von da an weiss er nicht was er als nächstes machen muss und deshalb schaut er in der Anleitung wieder nach wie man weiter vorangeht (**Gulf of Execution(7,4)**). Bei Schritt 3 wird gesagt, dass man den Film zum rechten Zylinder ziehen soll und dann die 'Löcher' im Film zu den 'Notches' einführen muss (Abbildung 5). Im Schritt 4 wird gesagt, dass man den Film weiterziehen und am Rand ausrichten soll. Doch es wird nicht gesagt, wie man am Film zieht. Der User schaut dann zu Schritt 3 und kann sehen, dass mit einem Pfeil signalisiert wird am Rad zu drehen. Trotzdem wird nicht gesagt, wie lange man am Rad drehen soll. (Abbildung 6)

Abbildung 6

Das ist ein **Gulf of Execution(8,4),** denn es steht nicht genau wie lange der User am Rad drehen sollte. Deshalb muss der User raten und am Rad drehen bis etwas passiert. Das ist auch ein kleiner **Gulf of Evaluation(8,5),** denn beim Drehen merkt der User nicht wie lange er noch drehen muss. Es gibt kein Progress-Bar oder sonstiges was den Progress zeigen würde, oder haptisch gibt es auch kein Feedback während dem Drehen, so dass es zum Beispiel schwieriger wird, je länger man dreht.

Der User dreht also lange am Rad und sieht das auch der Rewind Crank sich mit dreht. Er interpretiert das so, dass das Filmband beim Drehen des Rads immer mehr gezogen wird. Der User gelangt aber an einem Punkt, wo er nicht mehr am Rad drehen kann. Da weiss er, dass der Film bereit ist und er ein Bild schiessen kann.

i. Der User macht denn Deckel zu, und er weiss jetzt das die Kamera bereit ist Bilder zu schiessen
j. Der User nimmt ein Objekt vor sich und 'zielt' mit dem Viewfinder auf das Objekt und drückt auf den Shutter-Button. (Abbildung 7)

Abbildung 7

Es ertönt ein leises Geräusch, was man eigentlich fast nicht hören kann (kleiner **Gulf of Evaluation(8,6))**. Wenn man ein Bild geschossen hat, kann man am Rad drehen, damit man weitere Bilder schiessen kann. Wenn man am Rad dreht, sieht man auch dass gleichzeitig auch der Rewind Crank mit dreht. Der User interpretiert, dass wirklich neuer Film verwendet wird für den nächsten Bild. Er schaut auf die kleine Anzeige, um zu kontrollieren, ob die Kamera richtig gezählt hat. Die Zahl war vor dem Bild bei 12 und nach dem Bild ist sie jetzt 11. Das heisst die Kamera hat richtig gezählt und der User hat noch Kapazität für 11 weitere Bilder. Es könnte sein, dass der User vergessen hat, was die Zahl vorher war, und deshalb kann es passieren, dass der User nicht weiss, was die Zahl bedeutet. Es könnte bedeuten, dass der User 11 Bilder geschossen hat oder es könnte auch bedeuten wie viele Bilder verbleiben. Wenn man die Anleitung nicht anschaut, weiss man nicht, was die Zahl bedeutet. Deshalb ist dies ein kleiner **Gulf of Evaluation(8,7).** Der User hat dann das Feedback zu interpretieren.

5. Wahrnehmen: Beim Drücken auf den Shutter Button ertönt ein leises Geräusch. Der User probiert das Rad zu drehen und merkt das nun das Rad sich drehen lässt. Er merkt auch während dem Drehen, dass der linke Latch mit dreht.
6. Interpretieren: Das Bild wurde geschossen und befindet sich im Film. Der User muss am Rad drehen damit neuer Film kommt. Der linke Latch dreht mit und der User weiss, dass wirklich neuer Film kommt.
7. Vergleichen: Die Kamera wurde eingerichtet und der User konnte ein Bild schiessen

Insgesamt **8 Gulf of Executions** und **7 Gulf of Evaluations** in dieser Interaktion.

Appendix C

Ilias File Download

Ilias Ordner herunterladen

Der User hat ein gutes Konzeptmodell wie Ilias funktioniert. Es ist nicht sein erstes Semester an der Uni.

1. **Ziel**: Den Ordner 'Projektbeschreibungen' von der Vorlesung PSE herunterladen
2. **Planen**:
    a. Auf die Website von Ilias gehen und in der Vorlesungsseite den Ordner herunterladen
    b. Einen Freund fragen, ob er den Ordner schicken kann.
    c. Ein Email an den Professor schicken und fragen ob er den Ordner schicken kann (Footnote: 'Kultureller Unterschied')

    Der User entscheidet sich für die erste Option.

3. **Spezifizieren**:
    a. Browser öffnen
    b. In Google 'Ilias' eingeben und auf den ersten Link klicken
    c. Auf Ilias dann anmelden
    d. Zum Kurs navigieren, wo man den Ordner herunterladen will
    e. Im Vorlesungsverzeichnis / In der Vorlesungsseite den Ordner finden, den man herunterladen will.
    f. Rechts vom Ordner auf den Button klicken. Dann im Dropdown auf 'Herunterladen' klicken.
4. **Ausführen**:
    a. Browser öffnen
    b. 'Ilias' in die Google-Suchleiste eingeben und auf 'Enter' drücken. Der erste Link ist von der Ilias Seite von Uni Bern. Dann auf diesen ersten Link klicken (ilias.unibe.ch)
    c. Der User landet direkt in der Login Seite. Bevor man sich einloggt, will Ilias, dass der User eingibt über welche Universität er sich einloggen will (Abbildung 1). In der Mitte kontrollieren, ob die richtige Universität angewählt ist. Der User erwartete nicht, dass dies noch gefragt wird, denn im Link selbst steht am Schluss 'unibe.ch'. Das ist mühsam einen zusätzlichen Schritt zu machen, deshalb wäre das ein kleiner **Gulf of Execution(1,0).** Obwohl das dieser Schritt nicht seinen Vorstellungen entspricht, muss der User nicht lange überlegen was als nächstes zu tun ist. (Vielleicht auch ein kleiner **Gulf of Evaluation(1,1),** weil nicht sicher ob richtigen Link geklickt hat. User verwirrt wenn im Link unibe steht.)

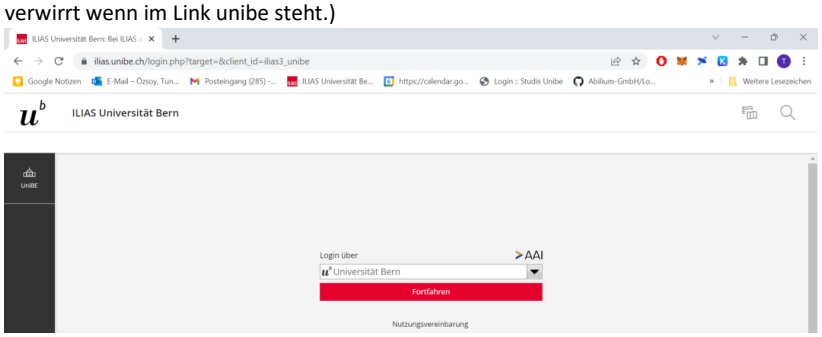


Abbildung 1

Und dann auf ‘Fortfahren’ klicken. Der User wird weitergeleitet zu SWITCH Edu-ID (Abbildung 2). Anschliessend dort die Anmeldedaten eingeben (email, password) und dann auf ‚Login‘ klicken.

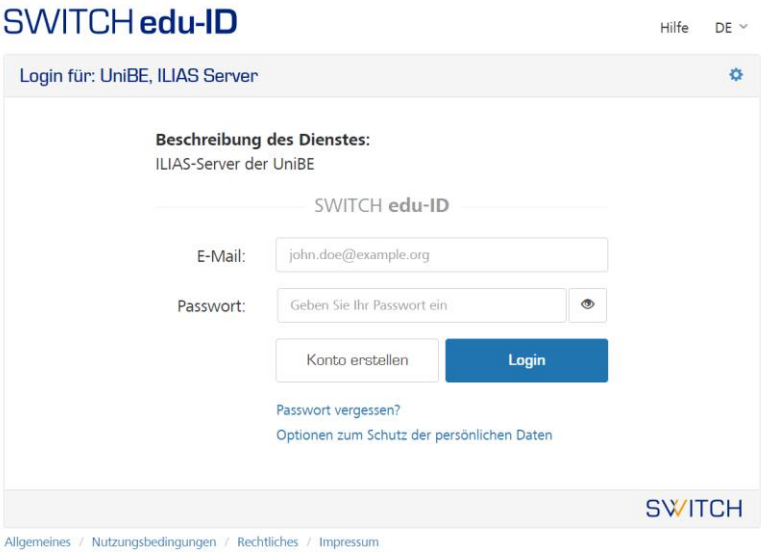


Abbildung 2

d. Nachdem man sich eingeloggt hat, kommt als erstes die Kalendersicht (Abbildung 3) Der User muss sich jetzt zum Ordner navigieren. Da der Ordner nicht in dieser Kalendersicht ist, klickt der User auf den Menüpunkt ‘Arbeitsraum’ (Abbildung 4). (Für neuere User ist das vielleicht nicht ganz selbstverständlich, denn der Menüpunkt ‘Unibe’ wäre ein Kandidat oder sogar auf dem Profil zu klicken. Deswegen vielleicht **Gulf of Execution(2,1)).** Dort werden ein paar Optionen angezeigt. Der User muss sich zwischen ‘Aktuelles Semester’ und ‘Meine Kurse und Gruppen’ entscheiden, denn diese zwei Optionen sind die einzigen die zum Ziel näherführen als Sicht des Benutzers. Der User entscheidet sich für ‘Aktuelles Semester’, denn die andere Option ist für den User aus Erfahrung zu unübersichtlich. Als der User auf ‘Aktuelles Semester’ klickt sieht er, dass er keine Vorlesungen hat in diesem Semester (Abbildung 5). Das wäre ein kleiner **Gulf of Evaluation(2,2)** für einen neuen User. Es könnte natürlich sein, dass der User vergessen hat, dass das neue Semester angefangen hat. Der User wäre deshalb überrascht, wenn er keine Vorlesungen gesehen hätte. Aber der User weiss, die Vorlesung, die er sucht, befindet sich im Frühjahrsemester 2022. Also navigiert er zum FS2022 Tab (Abbildung 6). Die ‘Vorlesung’, die er sucht, ist die Praktikum in Software Engineering (Abbildung 7). (Die Seite ist übersichtlich geordnet. Nebenfächer und Hauptfächer werden alphabetisch sortiert. In diesem Fall erscheint Betriebswirtschaftslehre als erstes und Informatik als nächstes Fach. Und in diesen Fächern werden die Vorlesungen auch alphabetisch sortiert. Die Mensch Maschine Schnittstelle Vorlesung befindet sich in der Informatik Kategorie weiter unten.) Nach dem er es gefunden hat klickt er darauf.

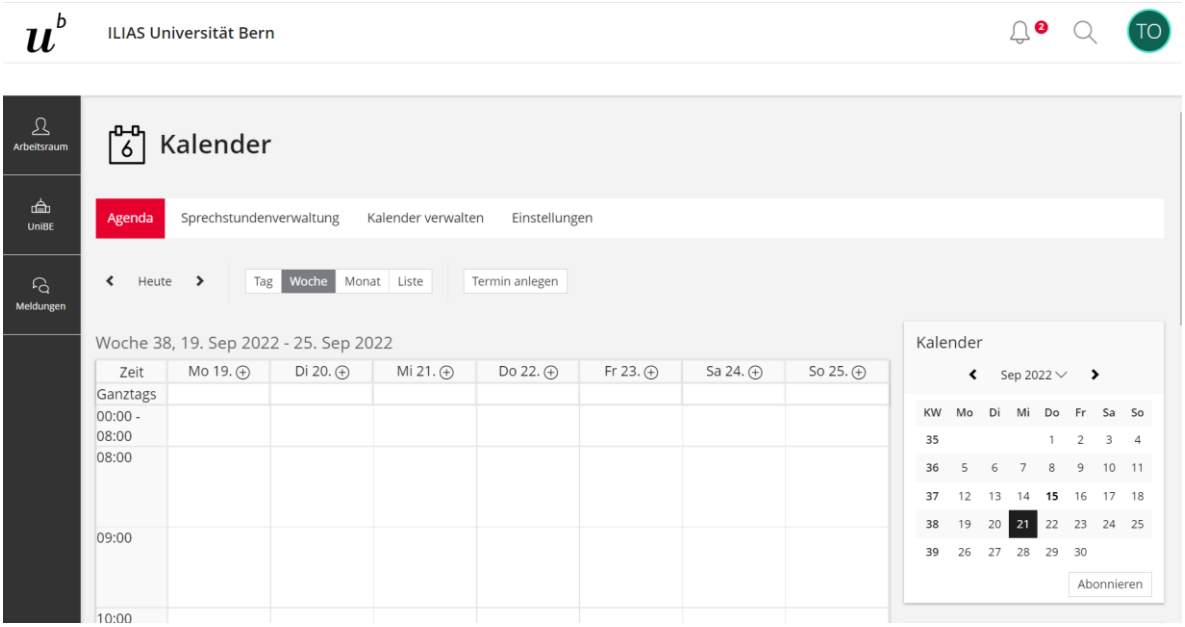


Abbildung 3

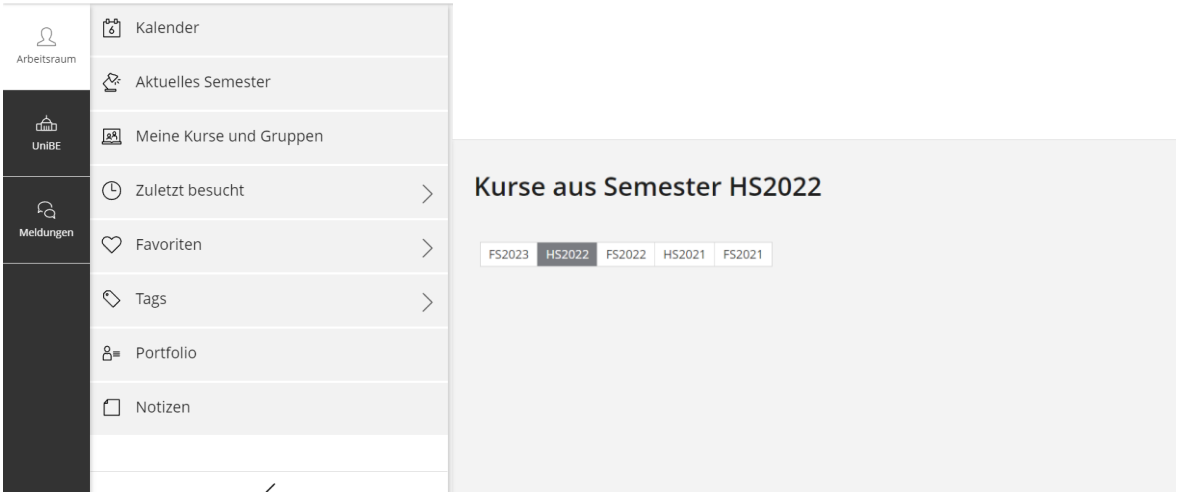


Abbildung 4 & 5

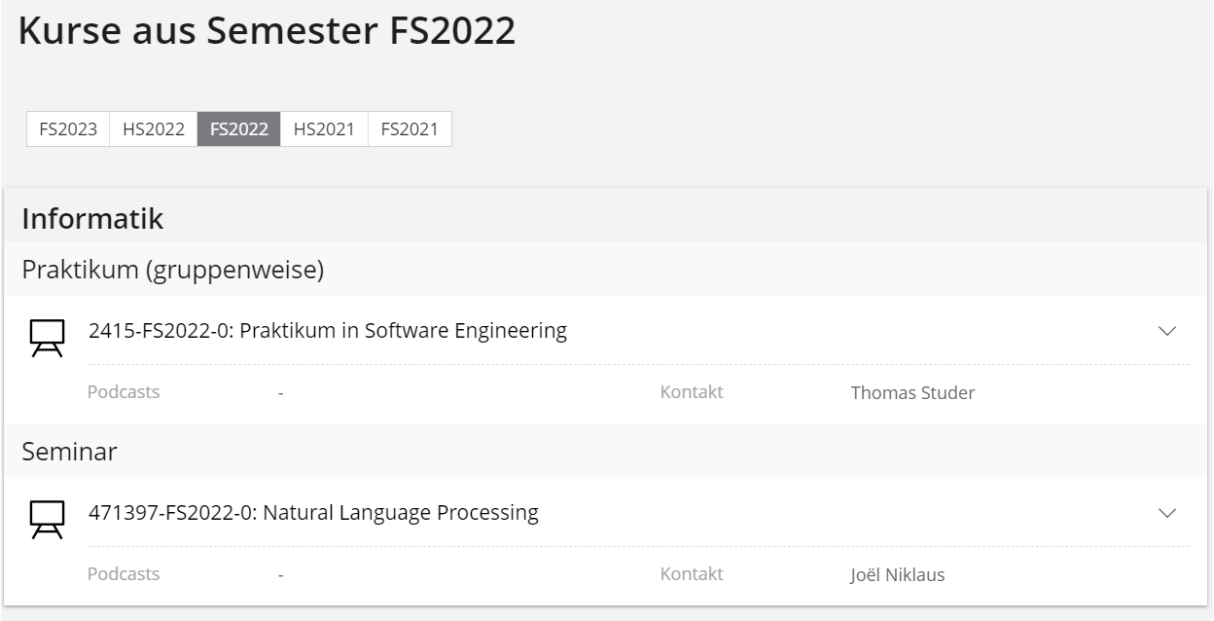


Abbildung 6

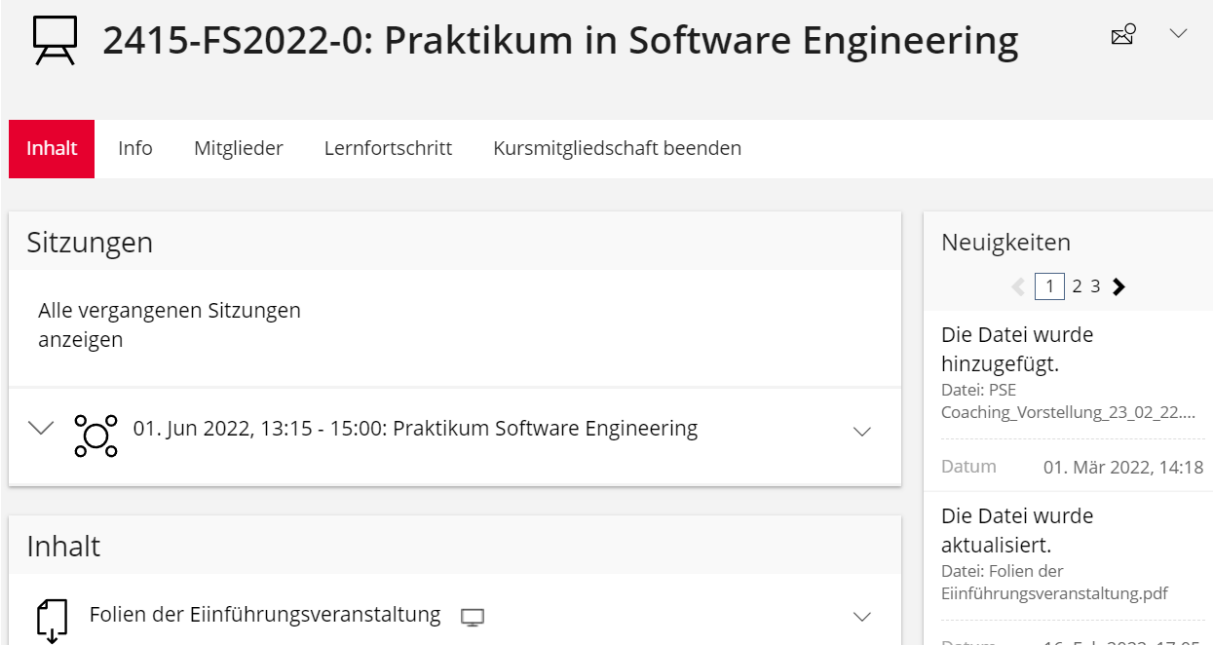


Abbildung 7

e. Der User sucht und findet den Ordner ‘Projektbeschreibungen’. Der User sieht sofort, dass es ein Ordner ist, weil hier der Akten-Icon für den Ordner verwendet wird. (Abbildung 8)

f. Der User sieht rechts vom gewünschten Ordner einen ‘Pfeil gegen unten’ -Button. Dieser Pfeil ist ein Signifier, dass man ein Dropdown Menü öffnet, wenn man auf den Button klickt. Der User klickt auf diesen Button und der Dropdown Menü öffnet sich (Abbildung 9). Als erste Option wird ‘Download’ angezeigt. Die Anordnung des Menüs ist wahrscheinlich nach Wichtigkeit geordnet, denn der Download Button ist ganz oben als erstes. Der User klickt dann auf den Download Button.

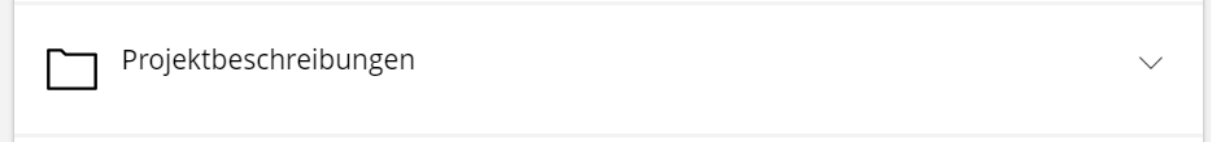


Abbildung 8

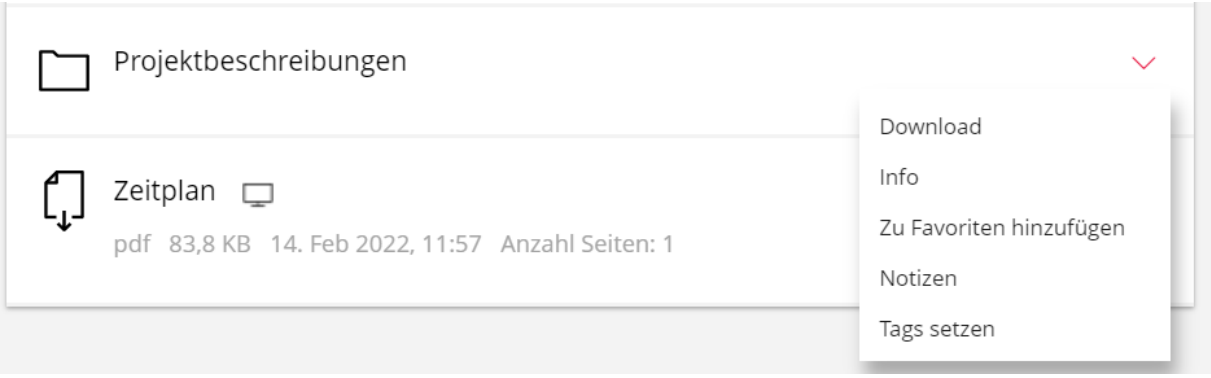


Abbildung 9

Nachdem der User auf 'Download' geklickt hat, wechselt man auf den Ordnerinhalt und diese Benachrichtigung wird angezeigt. (Abbildung 10)

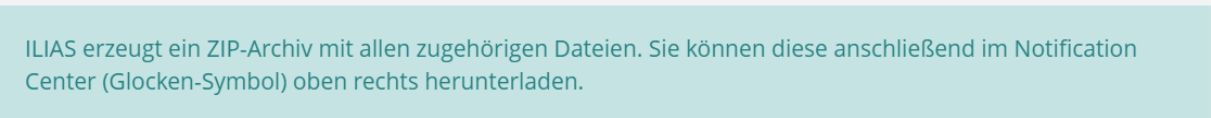


Abbildung 10

Das ist ein grosser **Gulf of Execution(3,2)**. Der User ist sich bei anderen Seiten anders gewohnt. Normalerweise sollte die Seite die Datei oder den Ordner direkt herunterladen und keine kleine Zwischenschritte einführen für solch eine simple Aufgabe.

Es gibt hier auch eine kleine (**Gulf of Evaluation(3,3)**). Nachdem man auf Download geklickt hat, kommt die Benachrichtigung das Ilias ein ZIP-Archiv erzeugt. Obwohl der Ordner nicht heruntergeladen wurde, bekommt man eine Benachrichtigung zu sehen (Abbildung 11). Diese Benachrichtigung gibt dem User Bescheid, was genau aus diesem «Download» passierte und wo man genau suchen muss für den Download. Da in der Benachrichtigung direkt zum Glockensymbol hingewiesen wird und beschrieben wird, wo sich das Symbol befindet, ist es ein sehr kleiner Gulf of Evaluation, da der User nicht viel Mühe hatte den Zustand des Systems zu sehen und zu interpretieren. Es ist trotzdem ein Gulf of Evaluation, denn der User hat so ein Feedback nicht erwartet, das weitere zusätzliche Schritte einführt.

Abbildung 11

5. **Wahrnehmen**: Benachrichtigung wird angezeigt. Und die Seite wechselt zum Ordnerinhalt. Ordner wurde nicht heruntergeladen.
6. **Interpretieren**: Ilias hat den Ordner in ein ZIP-Archiv gepackt. Ordner noch in den Servern. Muss deshalb nur noch heruntergeladen werden.
7. **Vergleichen**: Ordner nicht heruntergeladen. Man muss Ordner in Notifications-Menü herunterladen.

Nächste Iteration:

1. **Ziel**: Den Ordner 'Abbildungen zum Skript' von der Vorlesung MMS herunterladen
2. **Planen**:
    a. Auf die Website von Ilias gehen und in der Vorlesungsseite den Ordner herunterladen
    b. Einen Freund fragen, ob er den Ordner schicken kann.
    c. Ein Email an den Professor schicken und fragen ob er den Ordner schicken kann (Footnote: 'Kultureller Unterschied')

    Der User entscheidet sich für die erste Option.

3. **Spezifizieren**:
    a. Auf Glockensymbol klicken
    b. Auf den Ordner klicken, um die ZIP-Datei herunterzuladen
4. **Ausführen**:
    a. Der User klickt auf den Glockensymbol. Da sieht er 2 Optionen, Background Tasks und Posteingang (Abbildung 12). Da herrscht ein **Gulf of Execution(4,3).** Der User weiss nicht genau, wo er klicken muss, denn es wird nirgendwo Download oder den Ordner erwähnt. Auch in der Benachrichtigung wird nicht erwähnt, wo sich genau

der Ordner befindet. Trotzdem klickt der User dann auf Background Tasks, denn Posteingang kann es nicht sein, denn der Download handelt sich um keine Email. Der User sieht jetzt seine Ordner. Das war deshalb ein Gulf of Execution, denn die Vorstellungen des Users stimmten nicht überein mit dem, was das System erlaubt. Der User hat gedacht, dass die Ordner nach dem Klick auf den Glockensymbol direkt ersichtlich sind.

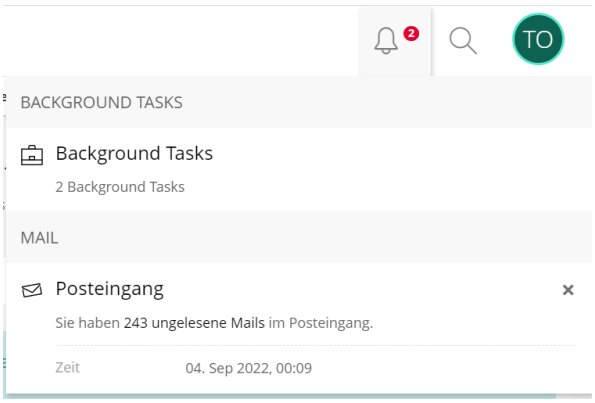


Abbildung 12

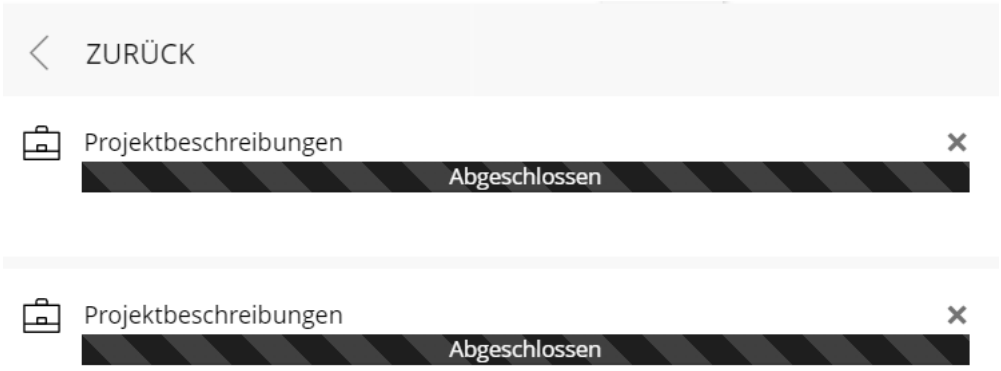


Abbildung 13

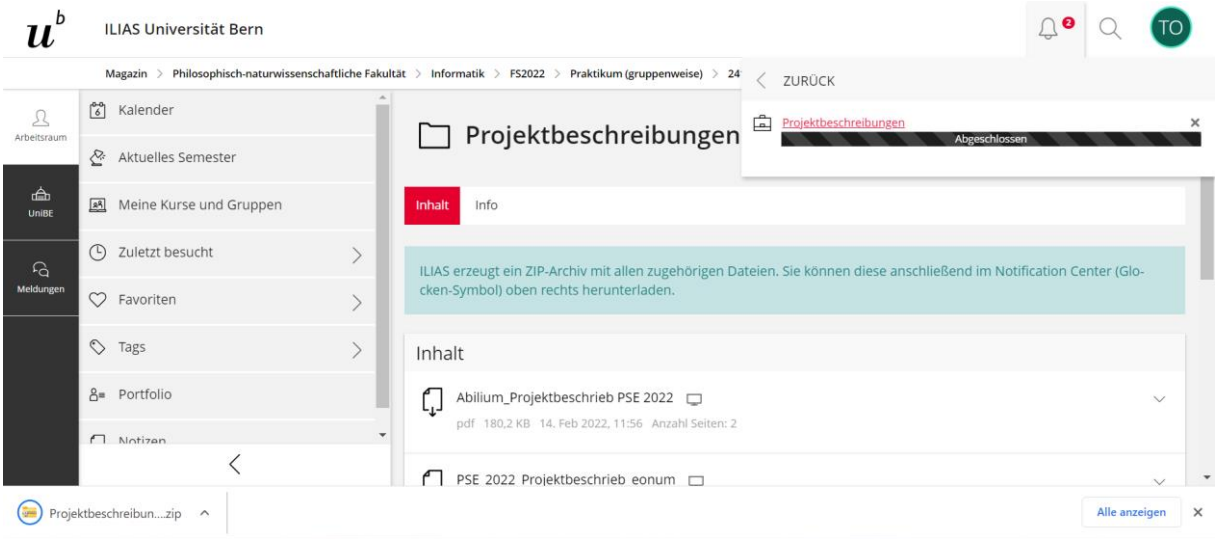


Abbildung 14

b. Er sieht jetzt seine Ordner, die der User heruntergeladen hat (Abbildung 13). Er sieht die 2 gleichen Ordner. Vielleicht hat der User früher versucht den Ordner herunterzuladen. Hier gibt es auch einen kleinen **Gulf of Execution(5,3)**, denn der User weiss nicht genau auf welchen Ordner er klicken muss/ downloaden muss. Das ist auch ein **Gulf of Evaluation(5,4)**, denn der User weiss nicht welcher Ordner der aktuelle ist. Es gibt keine zusätzlichen Informationen über diese Ordner. Es wäre hilfreich gewesen, wenn in der Anzeige bei den Ordnern stehen würde, wann genau diese 'Downloads' abgeschlossen wurde. Man sieht auch nicht wie diese Background Tasks angeordnet wurden. Der User rät und entscheidet sich für den oberen Ordner. Zum Herunterladen ist für den User auch nicht klar, wo er klicken muss. Das ist auch ein **Gulf of Execution(6,4)**. Dem User wird nicht leicht gemacht, wo man drücken muss zum Herunterladen. Es hat keine Download Icons oder sonstige Signifier, die man direkt erkennt. Der User bewegt mit dem Cursor zum Ordner und sieht, dass der Name vom Ordner sich anklicken lässt. Sonst lässt sich nichts anklicken, ausser dem 'Abbrechen' Button (X) rechts vom Ordner. Der User klickt dann auf den Namen. Der Ordner wird dann heruntergeladen. Kein Gulf of Evaluation. Der Download wird direkt ersichtlich im Browser.

5. **Wahrnehmen**: Zip-Datei /-Ordner wird heruntergeladen und man sieht es unten links am Browser (Abbildung 14)
6. **Interpretieren**: Der Ordner wird heruntergeladen und befindet sich dann auf meinem PC
7. **Vergleichen**: Ordner wurde heruntergeladen. Ziel erreicht.

Insgesamt **6 Gulf of Executions** und **4 Gulf of Evaluations** in dieser Interaktion.

Appendix D

Canon Image Transfer

Bilder von der Kamera zum Smartphone übertragen

Der User braucht seine Systemkamera zum ersten Mal, hat aber schon die Einrichtung abgeschlossen. Er weiss, dass man auf den Bildschirm drauftippen kann. Nun hat er sein erstes Bild geschossen. Jetzt will der User das Bild an seinem Smartphone schicken.

1. Ziel: Der User will das Bild von seiner Kamera zum Smartphone schicken
2. Planen:
    a. Kamera mit Smartphone verbinden. Vielleicht mit Bluetooth oder WLAN
    b. Die Speicherkarte herausnehmen und in das Smartphone hineintun
    c. Die Speicherkarte herausnehmen und in den Computer anschliessen. Danach das Bild auf den Desktop kopieren. Anschliessend das Smartphone an den PC anschliessen und die Datei vom Desktop in das Smartphone kopieren.
    d. Wie c.) aber Smartphone nicht an PC anschliessen, sondern das Bild zu einem Cloud Speicherlösung hineinkopieren. Anschliessend das Bild auf dem Smartphone ansehen.

    Der User wählt die erste Option aus.

3. Spezifizieren:
    a. Der User geht in die Einstellungen der Kamera.
    b. Er sucht die Konnektivitäts-Einstellungen und sucht eine Option die 'Mit Smartphone verbinden' oder so ähnliches heisst.
    c. Befolge Anweisungen
    d. Nachdem es verbunden wurde, zu den Bildern gehen in der Kamera und ein Bild auswählen die man zum Smartphone senden möchte.
    e. In die Galerie navigieren und kontrollieren, ob das Bild angekommen ist.

4. Ausführen:
    a. Der User will zuerst herausfinden, wie man die Einstellungen aufmacht. Als erstes schaut er an der Kamera herum, ob es einen Knopf gibt mit einem Werkzeugsymbol. Er sieht kein solches Symbol, aber dafür findet er einen Knopf, der mit 'Menu' beschriftet ist (Abbildung 1). Der User ist nicht sicher, dass dieser Knopf seinem Ziel die Einstellungen zu öffnen näherbringt (**Gulf of Execution(1,0)**). Er klickt darauf und es erscheint eine Menu-Sicht mit 4 verschiedenen Tabs und für jedes Tab hat es verschiedene Seiten. Er interpretiert diese Sicht als die Einstellungen.

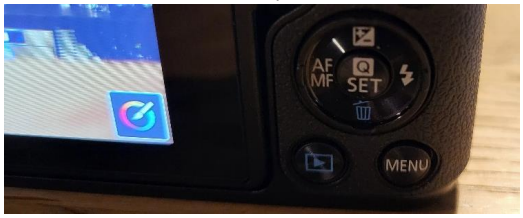


Abbildung 1

b. Der erste Tab ist mit ‘SHOOT1’ beschriftet und es hat eine Kamera Symbol am Tab (Abbildung 2). Der User kann aus dem rausschliessen, dass dieser Tab die Einstellungen von der Fotokamera ist. Bevor der User die Einstellungen durchsucht, muss er zuerst überlegen, wie er durch diese Tabs navigiert. Die Tastenelemente oberhalb vom Menütaste sind angeordnet, wie auf der Fernbedienung oder Gaming Controller (Nicht ganz eindeutig, deshalb hier ein kleiner **Gulf of Execution(2,0)**). Deshalb glaubt er, er kann mit diesen Tasten die Tabs und die einzelnen Einstellungen navigieren. Der User navigiert also mit diesen Tasten durch die Seiten dieses Tabs und sucht nach Keywords wie ‘Smartphone’, ‘Bluetooth’ oder ‘WLAN’. In diesem Tab findet der User nichts, dass seinem Ziel näherbringt. Der zweite Tab handelt sich um Bildereinstellungen (Abbildung 3). Das Icon ähnelt einem Videodatei, wegen diesem Startknopf. Nach dem Durchfliegen der Seiten weiss der User aber nicht ganz, was dieser Tab für ein Typ der Einstellungen ist (**Gulf of Evaluation(2,1)**). Der nächste Tab hat ein Symbol von einem Werkzeugsymbol. Schon auf der ersten Seite vermutet der User, dass dieser Tab der richtige ist. Unten findet er eine Option die ‘Wireless communication settings’ genannt ist (Abbildung 4). Nachdem diese Option für 1 oder 2 Sekunden angewählt ist, erscheint ein «Tooltip?», dass diese Einstellung erklärt. ‘Configure Wi-Fi, NFC, and Bluetooth settings’ wird in diesem Tooltip beschrieben (Abbildung 5). Das tönt fast nach dem, was der User vorhatte. Er entscheidet sich aber trotzdem weitere Seiten zu durchsuchen, vielleicht sieht er genau die Einstellung, die er sich vorgestellt hatte. Nämlich ‘Mit Smartphone verbinden’. In den nächsten Seiten findet er nichts dergleichen. Auch nicht in dem letztem Tab. Deshalb geht über zu der Einstellung ‘Configure Wi-Fi, NFC and Bluetooth settings’. Der User drückt auf den ‘SET’ Button, als ‘OK’ (Es wird nirgendwo erklärt wo man klicken muss für ‘OK’). Es erscheinen neue Optionen. Die Option ‘Send images to smartphone’ ist die Einstellung, die er wollte, denn diese Option entspricht seinem Ziel. (Abbildung 6)

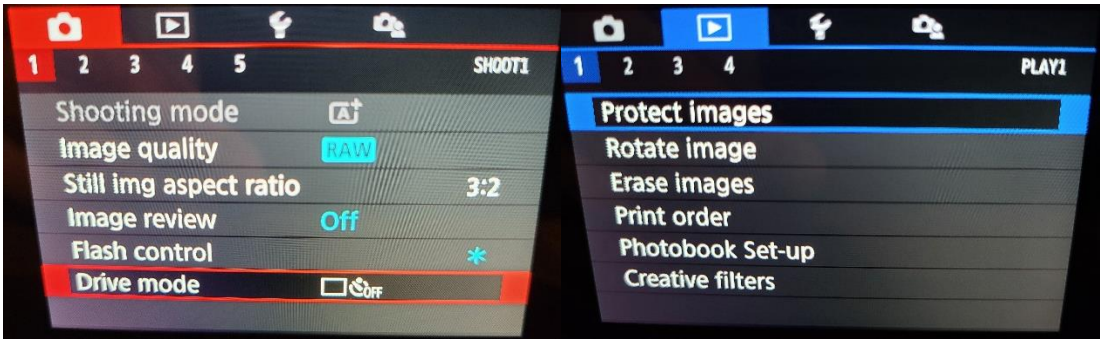


Abbildung 2 & 3

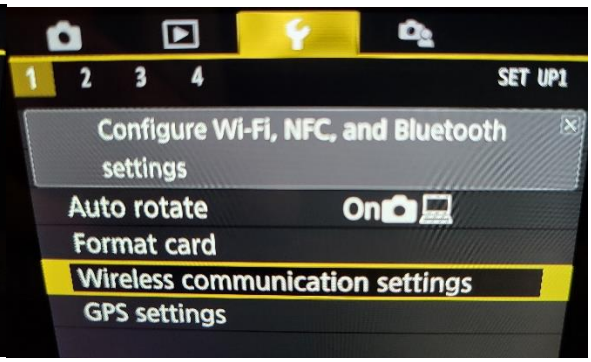


Abbildung 4 & 5

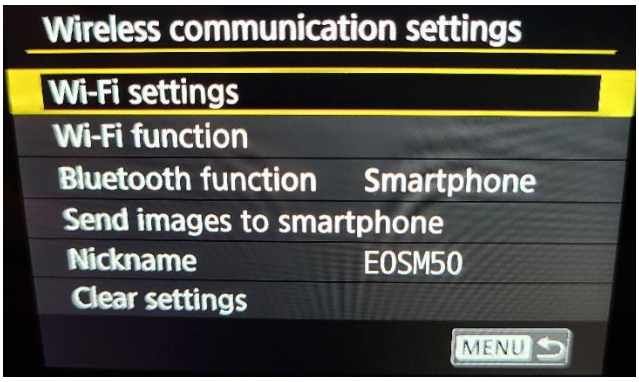


Abbildung 6

c. Als er darauf gedrückt hatte, bekommt er den Hinweis, dass das Smartphone nicht verbunden ist. Man müsste zuerst mit dem Smartphone verbinden, bevor man weiter macht (Abbildung 7). Das ist ein **Gulf of Execution(3,1)**, denn der User wird zurückgewiesen, um das Smartphone zu verbinden, was mit zusätzlichen Schritten verbunden ist. (Oder es ist ein **Gulf of Evaluation(3,2)**, denn dieses Feedback hatte der User nicht erwartet. Jetzt muss der User herausfinden, wie er zuerst mit dem Smartphone verbinden muss.)

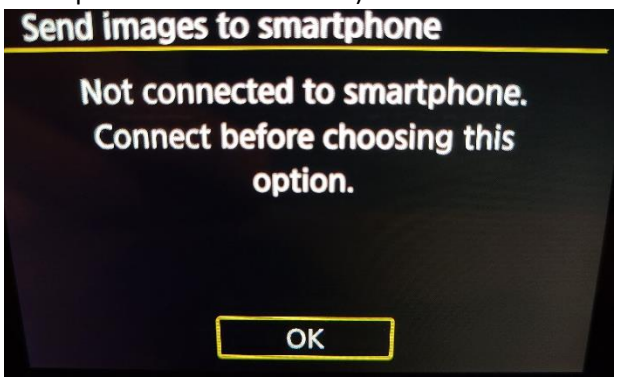


Abbildung 7

Deshalb klickt er auf 'OK' und nun wird ein weiteres Menu angezeigt. Es werden 2 Einträge gezeigt (Abbildung 8). Beide Einträge haben ein Smartphone Symbol was bedeutet das diese Einträge von Smartphones sind, die gescannt worden sind oder schon vorher verbunden waren. Ausserdem hat diese Sicht keine Überschrift wie 'Verbinde mit Smartphone' oder ähnliches und deshalb ist dies verwirrend für den User (**Gulf of Evaluation(3,3)**). Der User weiss nicht was machen, denn er hat ein

Button erwartet wie ‘Scan’, um mit seinem Smartphone zu verbinden (**Gulf of Execution(4,3)**). In dieser Sicht gibt es noch 2 Signifier: Nämlich die erste ist diese 2 kleine Kreise oben in der Mitte, das Signalisiert, dass dieser Menu 2 Seiten hat und der User sich im ersten befindet, denn der erste Kreis ist grösser als der andere. Weiter gibt es noch ein anderen Signifier, nämlich die Buttons, die sich links und rechts befinden. Das zeigt dem User, dass sie mit einem Klick auf einer dieser Buttons auf die andere Seite navigieren können. (die erste Sicht aber fühlt sich wie ein Ultimatum, eines der beiden Smartphones anzuwählen)

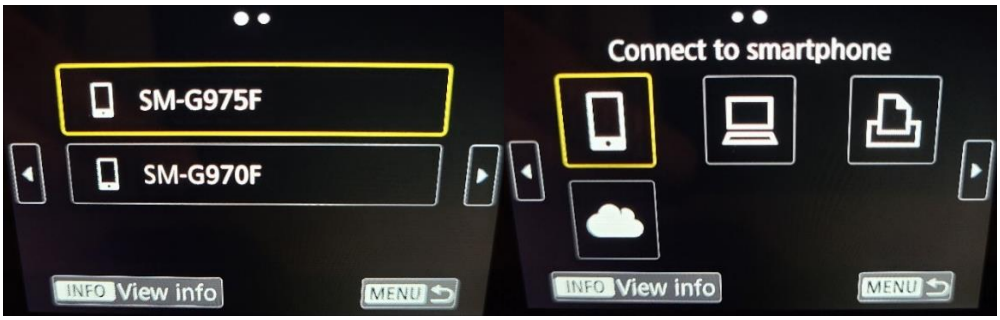


Abbildung 8 & 9

Auf der nächsten Seite ist der Überschrift ‘Connect to smartphone’ und das Symbol mit dem Handy ist ausgewählt (Abbildung 9). Aber wenn man weiter nach rechts navigiert steht ‘Connect to PC’ und das Symbol mit dem Computer wird ausgewählt. Der User wählt den ‘Connect to smartphone’ Button aus. In der nächsten View fragt die Kamera nach, welches Gerät man verbinden möchte. Der User wählt ‘Register a device for connection’. Nachdem kommt der Hinweis, dass der User die Camera Connect App herunterladen sollte (Abbildung 11). Das ist ein kleiner **Gulf of Execution(5,3)**, denn der User hatte nicht erwartet, dass man für das Verbinden eine App herunterladen muss. Das entsprach nicht seinen Vorstellungen. Ausserdem wird gefragt, ob ein QR-Code angezeigt werden soll für das Download der App. Der User klickt auf ‘Do not display’ und entscheidet das selbst herunterzuladen. Nach dem Klick sucht die Kamera nach ‘Access points’ (Abbildung 12). Der User weiss nicht, wie er diese Sicht interpretieren soll (**Gulf of Evaluation(5,4)**).

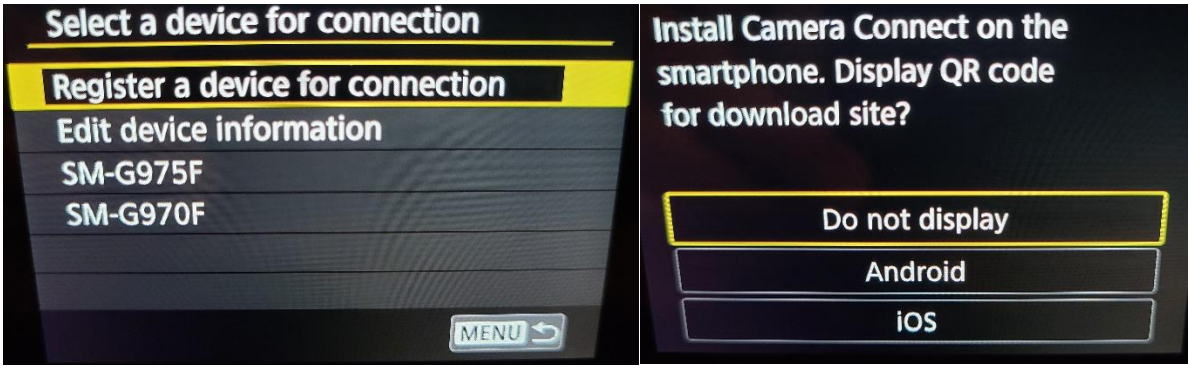


Abbildung 10 & 11

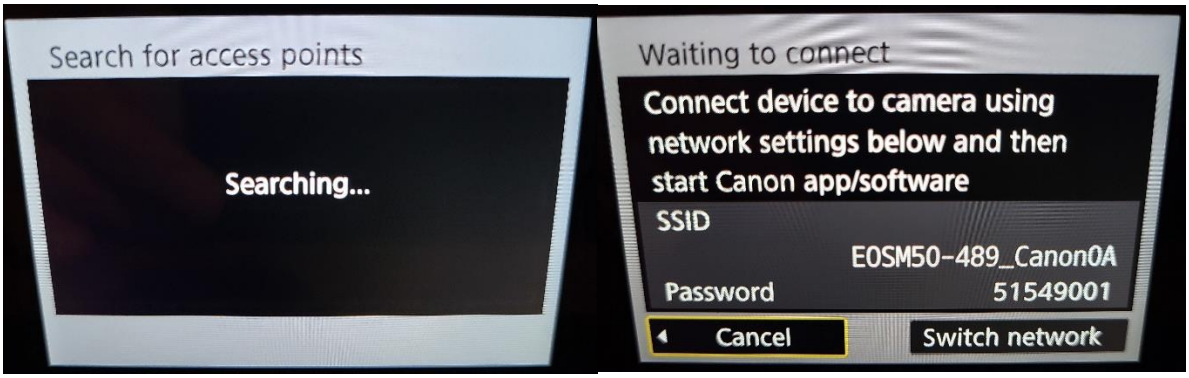


Abbildung 12 & 13

Währenddessen ladet der User im Google Play Store die App herunter. Bevor die App heruntergeladen wird, zeigt die Kamera nun plötzlich an, dass sie ein Wi-Fi Netzwerk erstellt hat (Abbildung 13). Nachdem die App heruntergeladen wurde, schaltet der User das WLAN ein vom Smartphone und verbindet sich mit diesem Netzwerk, wie man auf der Benutzeroberfläche der Kamera gesehen hat (Abbildung 14 & 15). Der User öffnet dann die App und akzeptiert die Lizenzvereinbarung und die Erfassung der Informationen zum Status der Verwendung. Der User lässt dann eine Berechtigung der App zu. Im Willkommens-Sicht überspringt er alle Seiten und drückt dann auf Start. Später gewährt der User den Zugriff auf Standortinformationen. Danach kommt eine Anzeige ‘Kamera erkennen’ (Abbildung 16). Der User wählt seine Kamera an. Die Verbindung wird hergestellt (Abbildung 17) und der User drückt auf OK an der Kamera. Der User sieht dann an der Kamera, dass die Verbindung abgeschlossen ist, und es steht Wi-Fi on. Dann klickt er auf Menu. In der App sieht er das das Smartphone mit der Kamera verbunden ist (Abbildung 18). Dann klickt er auf OK. Und wieder auf OK.

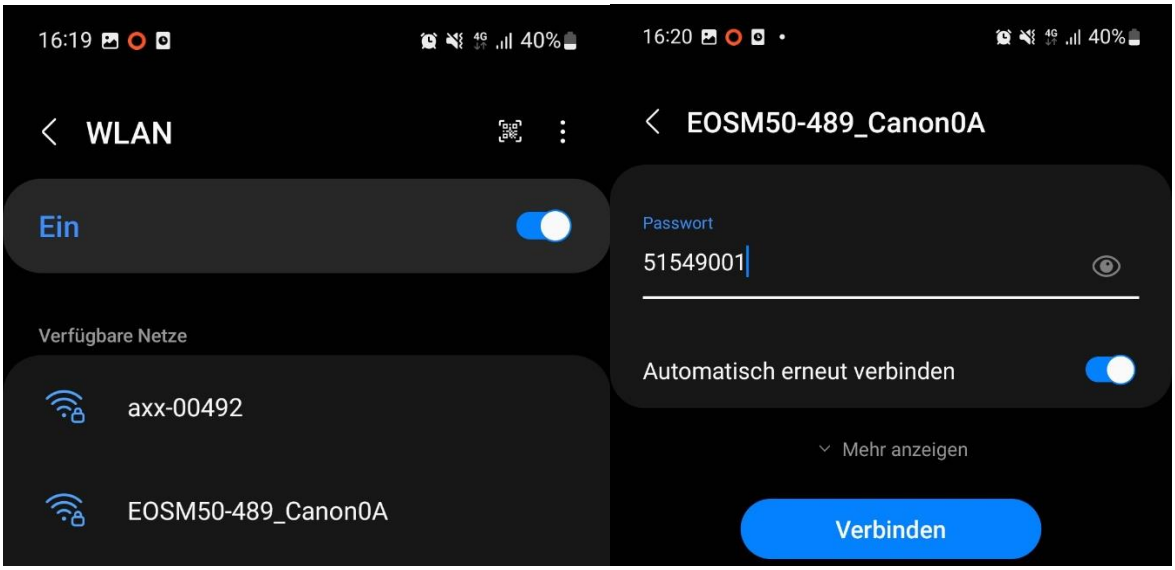


Abbildung 14 & 15

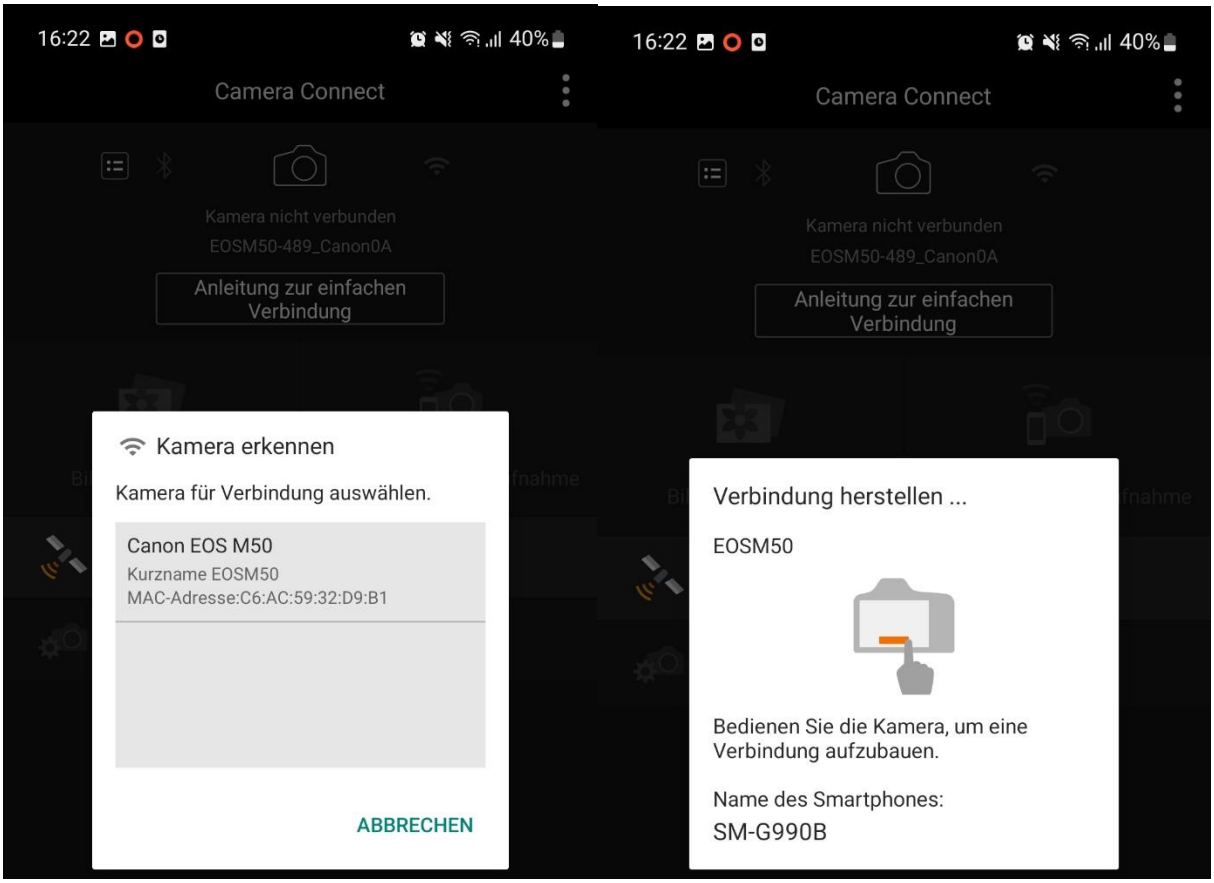


Abbildung 16 & 17

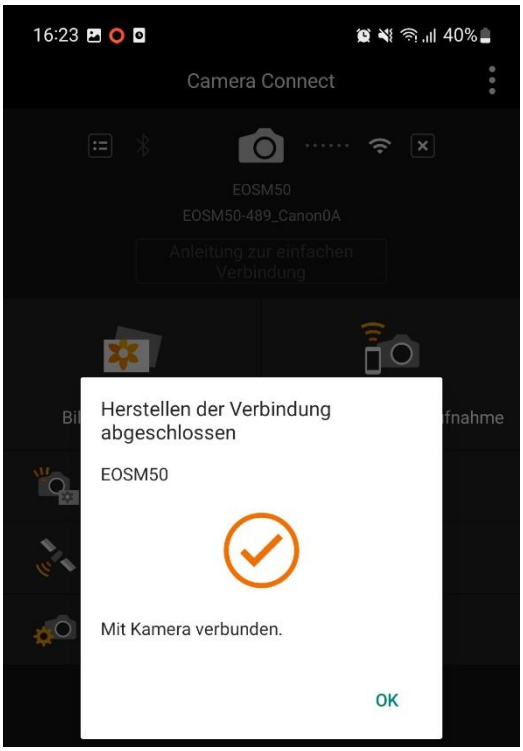


Abbildung 18

d. Der User will zu den Bildern gehen und deshalb klickt er auf der App auf den ‘Bilder auf Kamera’ Button (Abbildung 19). Schon erscheinen alle Bilder der Kamera (Abbildung 20).

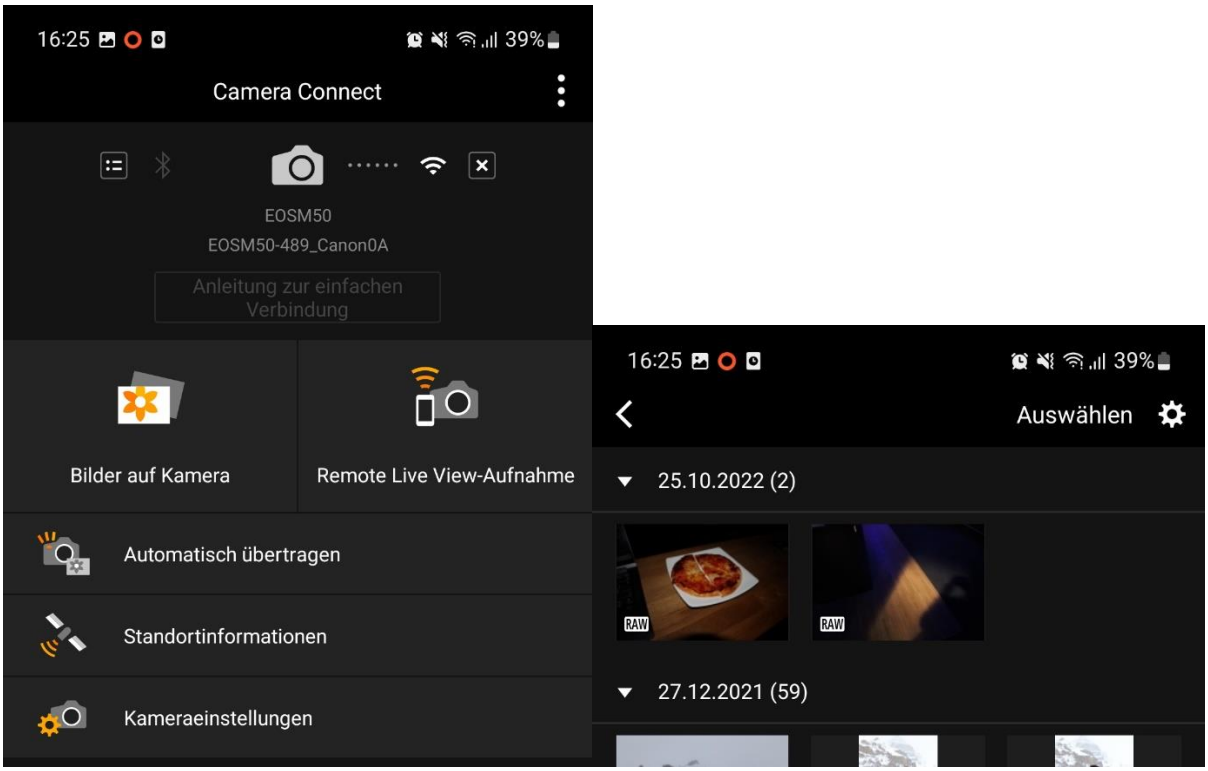


Abbildung 19 & 20

Der User wählt auf das Bild, die er vor kurzem geschossen hatte. Zum Herunterladen hat er eine Option, und zwar der ‘Import’ Button (Abbildung 21). Es wird nicht explizit Download genannt, was manche User verwirren könnte. Deshalb ist dies ein kleiner **Gulf of Execution(6,4)**, denn die Vorstellungen der User ist nicht dasselbe. Dieser Gulf of Execution könnte zu anderen Aktionen vom User führen. Weil sie nicht sicher sind welcher Button zum Herunterladen ist, könnten sie vielleicht probieren den ‘Apps’ Symbol zu klicken oder vielleicht sogar aus dem Bild Ansicht zurück zum Gesamtsicht gehen und dort den Download Button suchen.
Nachdem der User auf Import geklickt hat, kommt ein Hinweis in welcher Speichergrösse das Bild heruntergeladen werden soll. Der User wählt Originalgrösse und klickt auf OK. (Abbildung 22)

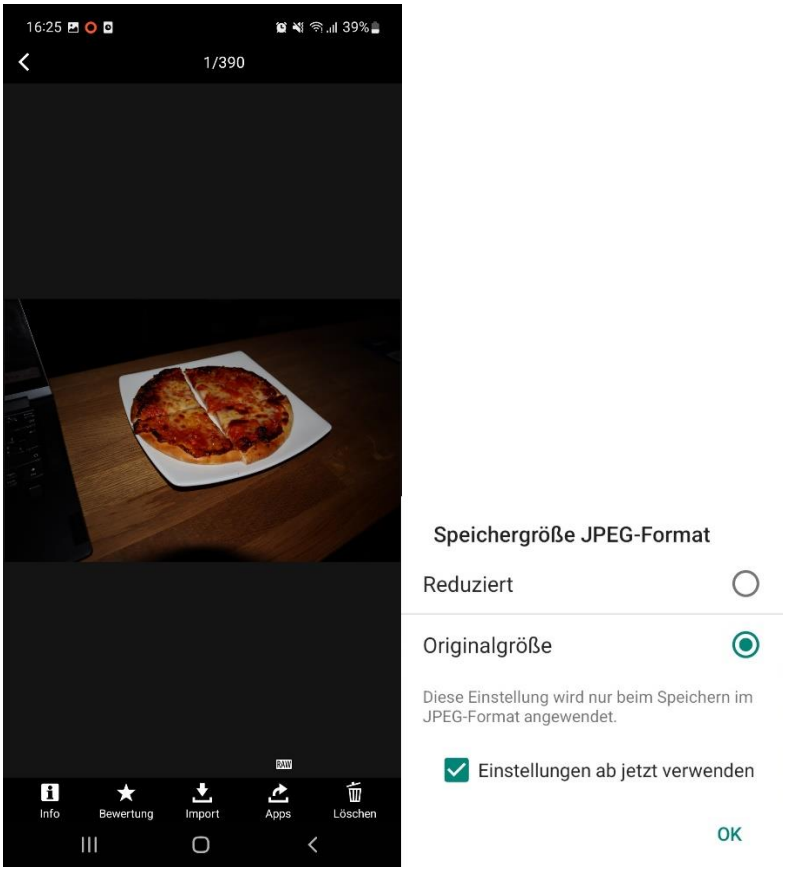


Abbildung 21 & 22

Es erscheint ein Progressbar und schon nach einer Sekunde wurde das Bild heruntergeladen. (Abbildung 23)

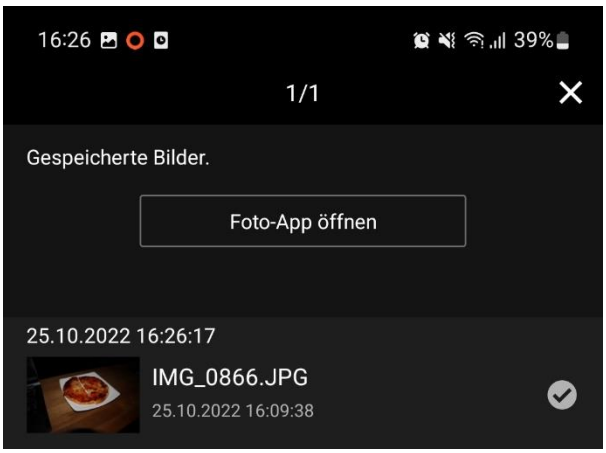


Abbildung 23

e. Auf der App kann der User auf ‘Foto-App öffnen’ klicken, um das Bild in der Galerie zu sehen. Nach dem Klick sieht er direkt, dass das Bild auf seiner Galerie ist. (Abbildung 24)

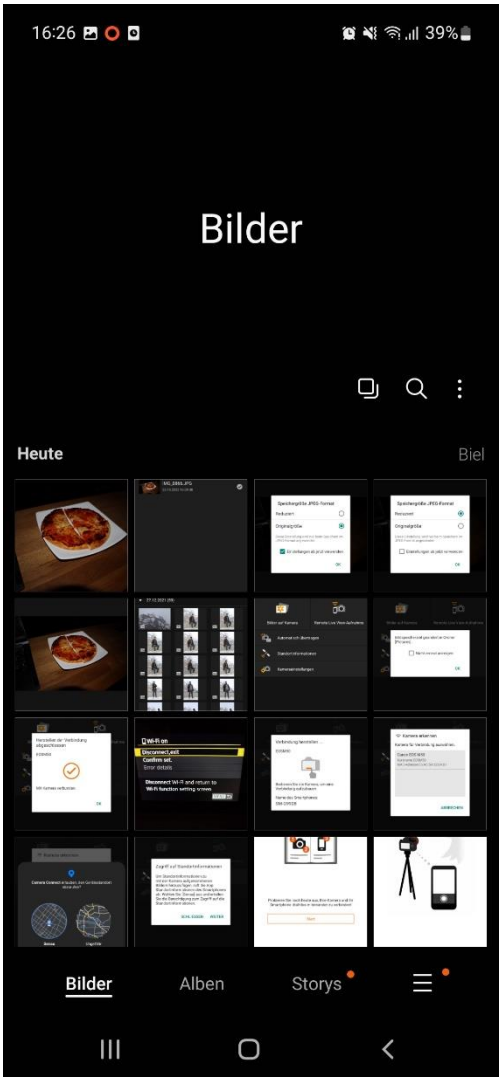


Abbildung 24

5. Wahrnehmen: Der User sieht sein Bild auf seinem Smartphone.
6. Interpretieren: Das Bild wurde heruntergeladen.
7. Vergleichen: Ziel erreicht. Smartphone verbunden mit der Kamera. Bild wurde von der Kamera zum Handy heruntergeladen.

Insgesamt **6 Gulf of Executions** und **4 Gulf of Evaluations** in dieser Interaktion.

Appendix E

Creating Crypto Wallet

Krypto-Wallet erstellen

Der User hat Basiswissen was Kryptowährungen und Blockchains sind. Der User wird zum ersten Mal ein Krypto-Wallet erstellen.

1. **Ziel**: Der User will einen Krypto-Wallet erstellen, um damit mit Web3 Applikationen zu interagieren und Kryptowährungen zu senden und zu empfangen.
2. **Planen**: Der User hat einige Optionen:
    a. Web-Wallet herunterladen (z.B 'Metamask')
    b. Desktop-Wallet herunterladen
    c. Mobile-Wallet herunterladen
    d. Hardware-Wallet kaufen und einrichten

    Der User wählt Option a.

3. **Spezifizieren**:
    a. Auf Google 'Metamask' eintippen und suchen
    b. Auf den ersten Link klicken, der keine Werbung ist
    c. Danach auf der Seite auf den Download Button klicken
    d. Den Wallet installieren und starten
    e. Wallet einrichten
4. **Ausführen**:
    a. Öffnet den Browser Edge. In die Suchleiste 'metamask' eintippen und suchen
    b. Auf den ersten Link klicken, der keine Werbung ist (Abbildung 1). Das heisst der User klickt auf 'metamask.io'.
    c. Der User sieht die Startseite von Metamask. Auf dieser Seite gibt es 2 Buttons mit hohem Affordance. Diese Buttons haben einen blauen Hintergrund. Der User klickt oben rechts dann auf Download. (Abbildung 2)

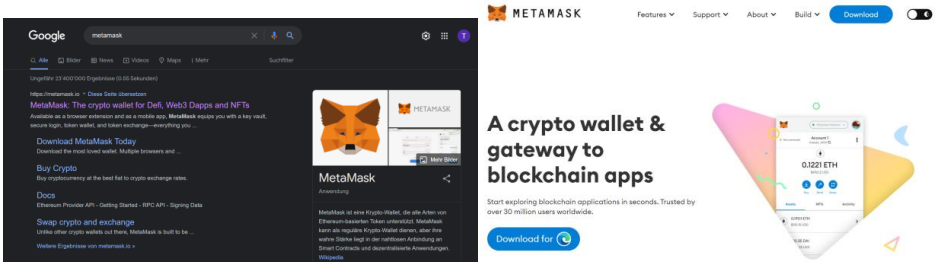


Abbildung 1 & 2

Dann kann der User auswählen auf welcher Plattform man installieren möchte (Abbildung 3). Edge wurde schon automatisch ausgewählt. Der User entscheidet sich für die Edge Extension. (Der User hat nicht gewusst, dass dieser Wallet eine Edge Extension ist. Nicht die gleichen Vorstellungen von System und User.)

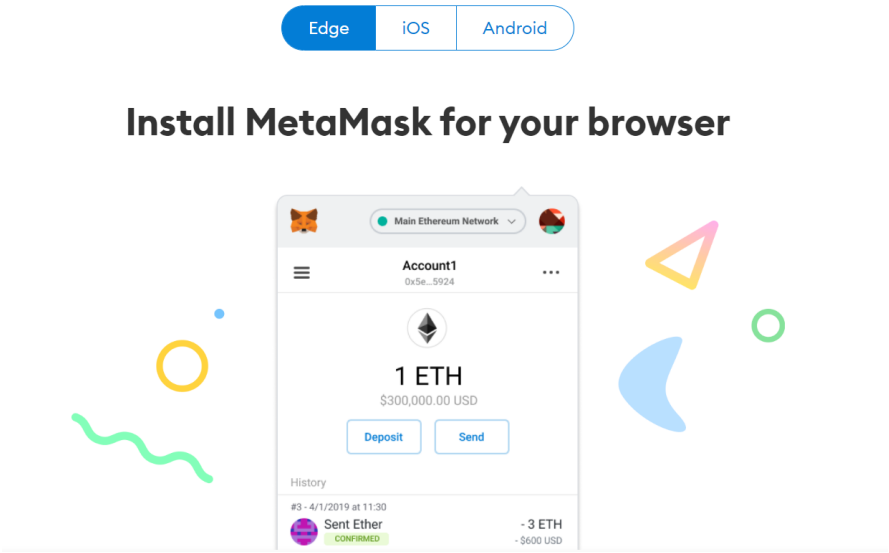


Abbildung 3

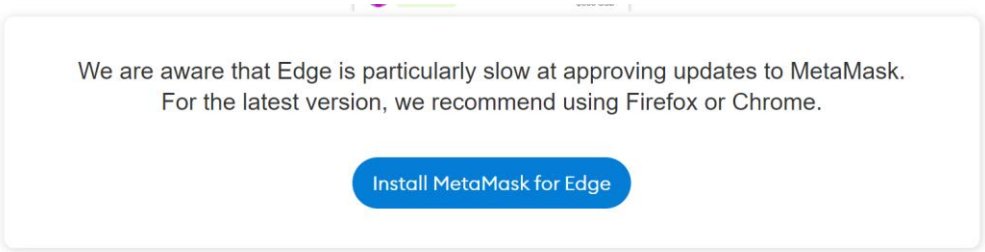


Abbildung 4

Dann klickt der User auf ‘Install Metamask for Edge’ (Abbildung 4). Der User wird weitergeleitet zu Microsofts Edge Add-ons Seite. Der User sieht, dass diese Extension auf Edge über 1 Millionen Benutzer hat. Das gibt dem User auch ein sichereres Gefühl dieser Software. Er sucht dann den ‘Download’ Button, aber es gibt nur den Button ‘Get’, was für den User das gleiche bedeutet. Deshalb klickt der User auf ‘Get’ (Abbildung 5). Dann erscheint ein Pop-Up zum Hinzufügen der Erweiterung. Der ‘Abbrechen’ Button hat ein höheres Affordance und wurde schon vorgewählt, damit der User nicht gleich automatisch auf ‘hinzufügen’ klickt und die Beschreibung der Erweiterung liest. Der User akzeptiert diese Berechtigungen und klickt auf ‘Erweiterung hinzufügen’. (Abbildung 6)

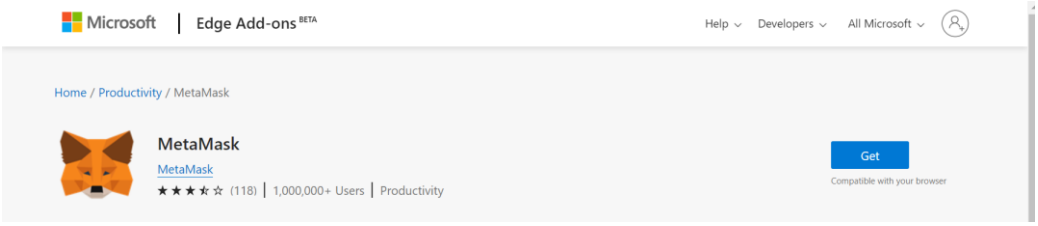


Abbildung 5

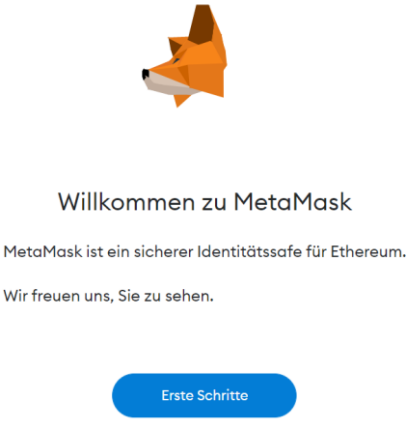


Abbildung 6 & 7

d. Eine neue Seite wird geöffnet (Abbildung 7). Metamask heisst den User willkommen und beschreibt kurz was Metamask macht. Es wird beschrieben, dass Metamask ein sicherer Identitätssafe für Ethereum ist. Der User ist ein wenig von der Beschreibung verwirrt, denn er hatte gedacht, dass dieser Wallet auch für andere Blockchains/Netzwerke funktioniert. Und mit dieser Beschreibung klingt es so, dass dieser Wallet nur für Ethereum gedacht ist (kleiner **Gulf of Evaluation(0,1)**). Der User interpretiert, dass er jetzt den Wallet einrichten muss. Der User klickt dann auf 'Erste Schritte'.

   Im ersten Schritt fragt Metamask ob sie Nutzerdaten sammeln können. Der User sieht auf den ersten Blick, dass die Übersetzung auf Deutsch sehr schlecht ist. Trotzdem entscheidet er auf 'Ich stimme zu' zu klicken. (Abbildung 8)

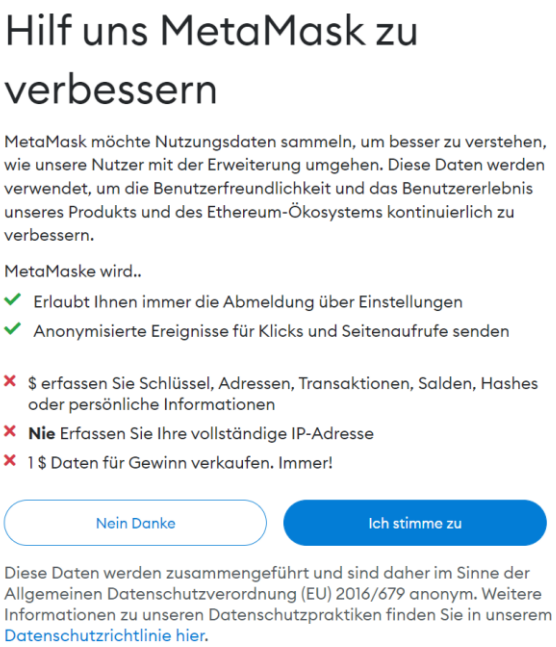


Abbildung 8

Nachdem man sich entschieden hat, wird man gefragt, ob man neu bei Metamask sei. (Abbildung 9)

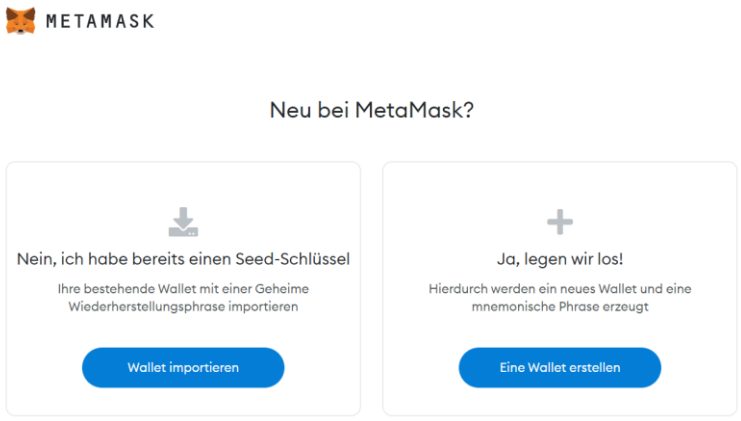


Abbildung 9

Da der User noch keinen Seed-Schlüssel hat, klickt er auf ‘Eine Wallet erstellen’.

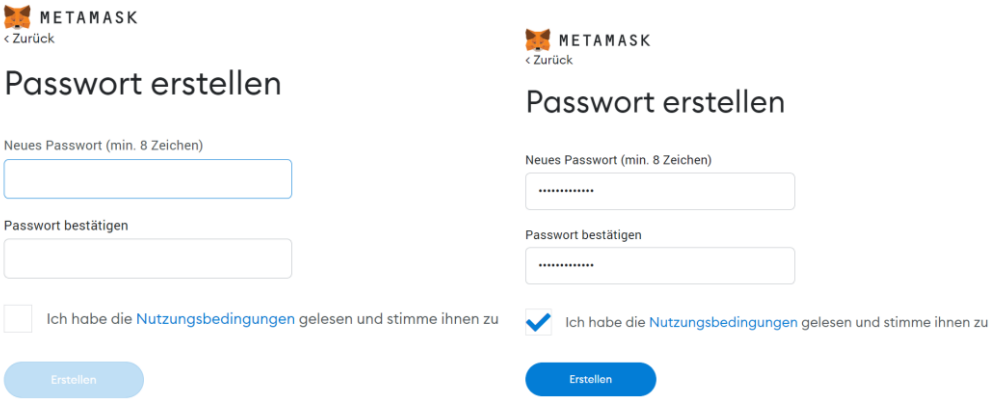


Abbildung 10 & 11

Der User muss jetzt ein neues Passwort erstellen (Abbildung 10). Nachdem er sein Passwort eingegeben hat, ist der ‘Erstellen’ Button immer noch ausgegraut. Er setzt noch das Kreuz ein und direkt wird der Button wieder farbig, was ein gutes Feedback ist (Abbildung 11). Besser gesagt ist das ein Constraint, dass den User zwingt ein Passwort auszuwählen, bevor er weitermacht. Der ausgegraute Button ist ein Signifier zum Signalisieren, dass der Button inaktiv ist. Der User klickt dann auf den ‘Erstellen’ Button.

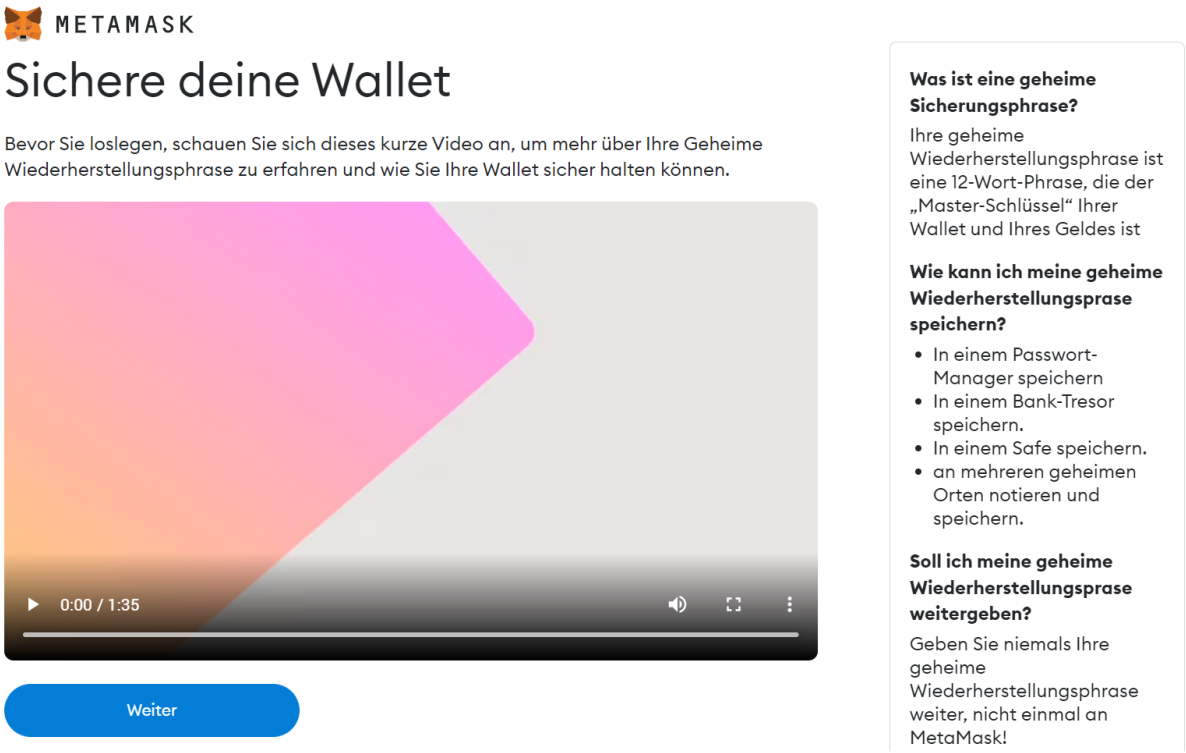


Abbildung 12

Der User wird über das richtige Sichern der Wallet aufmerksam gemacht (Abbildung 12). Der User lernt neue Begriffe wie ‘geheime Sicherungsphrase’, und ‘Wiederherstellungsphrase’. Diese 2 Begriffe bedeuten das Gleiche aber wurden wegen der schlechten Übersetzung anders geschrieben. Nachdem der User das Video angeschaut hat, drückt er auf ‘Weiter’. (Potentiell Gulf of Evaluation wenn Video nicht gesehen)

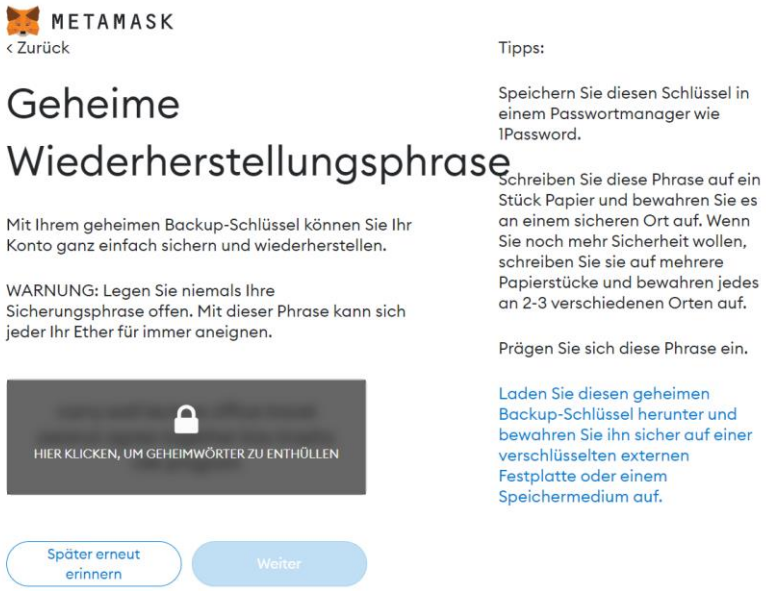


Abbildung 13

Im nächsten Schritt kann man den geheimen Wiederherstellungsphrase anzeigen lassen (Abbildung 13). Der User wird auf den Text ‘Hier klicken, um Geheimwörter zu enthüllen (‘Signifier’) aufmerksam gemacht. Dieser Text ist ein Signifier, dass den User eine zusätzliche

Information gibt, dass man auf diesem ‘zensierten’ Feld draufklicken kann. Da der ‘Weiter’ Button noch grau ist drückt der User auf den ‘zensierten’ Feld. Die 12 geheimen Wörter werden aufgedeckt. Der User schreibt diese dann auf ein Stück Papier, wie im Video beschrieben und ignoriert dabei den Rat diese Wörter auf ein Passwortmanager einzutragen. Der ‘Weiter’ Button ist nun blau und der User klickt drauf.

Es erscheint ein Fenster. Der User wird gebeten seine Geheimschlüssel aufzuschreiben beziehungsweise in der richtigen Reihenfolge auszuwählen. (Abbildung 14)

Abbildung 14

Abbildung 15

Metamask will damit sichergehen, dass der User diese auch aufgeschrieben/ irgendwo notiert hat. Der User versucht zuerst diese Wörter einzeln via Drag&Drop auf dem Textfeld zu ziehen (Abbildung 15). Er merkt aber schnell, dass das nicht funktioniert, denn die Wörter bewegen sich nicht (kein Gulf of Evaluation, obwohl User das nicht erwartet hatte verstand er trotzdem den Zustand des 'Systems'). Das ist ein kleiner **Gulf of Execution(1,1)**, weil der User sich das anders vorgestellt hat und nun überlegen muss, wie man Wörter auswählt. Der Mauszeiger ändert sich zum 'Verknüpfungsauswahl', wenn man den Zeiger auf ein Wort bewegt, was bedeutet das man den Button anklicken kann. (Abbildung 16)

Abbildung 16

Der User versucht dann die Wörter nur mit einem Mausklick auszuwählen. Dies hat funktioniert, denn die ausgewählten Wörter werden zum Textfeld übertragen und das Wort, das angeklickt wurde, wechselt zum dunkelblauen Hintergrund (als Signifier, dass dieses Wort schon ausgewählt wurde). Wenn man dann wieder auf den Button klickt, wird das Wort vom Feld zurückgenommen und der Button wird wieder ausgegraut. Als der User mit dem Maus über die ausgewählten Wörter zieht, wechselt der Mauszeiger zum 'Verschiebezeiger' (Abbildung 19). Das ist ein Signifier, dass man die Wörter verschieben kann. Nachdem der User die geheimen Wörter in der richtigen Reihenfolge ausgewählt hat, drückt er auf 'Bestätigen'. (Kleiner Gulf of Evaluation, wenn man Prozess unerwartet unterbricht und dann wieder auswählen muss, kann es für den User schwierig sein zu interpretieren welche Wörter schon ausgewählt worden sind. Meiner Meinung nach sollten die Buttons aufleuchten, die noch auswählbar sind. Denn diese Buttons haben eine hohe Affordance/Signifier und nach

einer Unterbrechung fällt es für den User einfacher diese Buttons zu klicken, obwohl diese schon ausgewählt worden sind.)

Der User hat beim Auswählen auf das falsche Wort geklickt. Um ein Wort wegzunehmen, hat man 4 Optionen. Die erste Option wäre, dass man auf dem Wort anklickt, entweder das obere Wort oder untere um das Wort zu 'deselecten'. Eine weitere Option wäre oben das Wort anzuwählen und dann mit gedrückter Maustaste nach unten verschieben oder aus dem Feld rausschieben oder dorthin verschieben zum Papierkorb, wenn es auftauchen würde.

Der User will die Option anwenden, wo man das Wort draufklickt, um das gewählte Wort zu entfernen. Der User bewegt mit dem Mauszeiger zum Wort und währenddessen ändert sich der Umriss des Wortes/Buttons (Abbildung 17). Danach klickt er auf das gewählte Wort und beim Klick füllt sich der Button mit einem dunkleren Hintergrund als Feedback, dass der Button geklickt wurde (Abbildung 18). Der User weiss, dass das System den Tastendruck ‚wahrgenommen' hat. Aber nach diesem Druck auf den Button änderte sich nichts. (Das ist ein **Gulf of Execution(2,1)**, denn nach dem Klick ist das Wort nicht verschwunden. Somit wendet er eine andere Option an.)

Abbildung 17

Abbildung 18

Abbildung 19

Er wählt das Wort aus und drückt lange drauf, so dass er es verschieben kann und währenddessen schaut er nach, ob ein Papierkorb erscheint, um das Wort dann dorthin zu verschieben. Dieser Papierkorb erscheint nicht. (Das wäre ein **Gulf of Execution(3,1)**). Der User versucht dann das Wort aus dem Feld rauszuziehen. Der Mauszeiger ändert sich ausserhalb vom Feld zum ‚Nicht verfügbar' Zeiger (Abbildung 20) und man kann diese Aktion nicht durchführen. Das ist ein ‚Modeless Feedback'. (Keine Gulf of Evaluation, denn der Mauszeiger war ein gutes Feedback, um den User zu informieren, dass man das Wort nicht außerhalb vom Feld ziehen kann.)

Abbildung 20

Am Schluss bleibt noch die Option auf den gleichen Button zu klicken, wo der User vorher für die Auswahl geklickt hat, um es zu löschen. Das hat funktioniert und der User hat seinen Fehler nun behoben. Der Bestätigen Button leuchtet nun und der User klickt darauf.

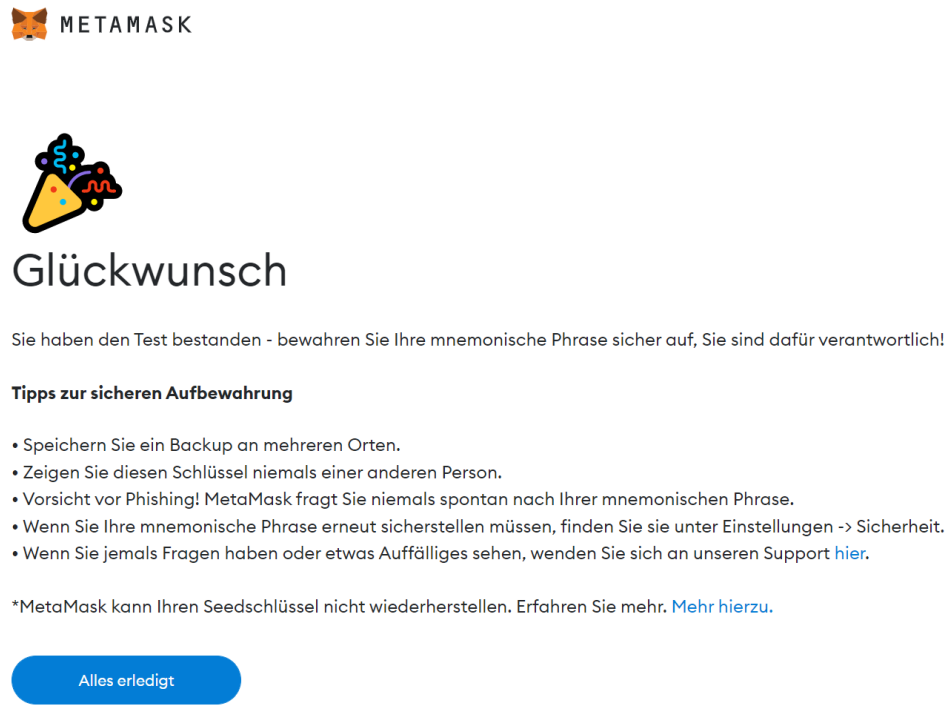


Abbildung 21

Dem User wird ein Feedback gezeigt, dass sie den 'Test' bestanden haben (Abbildung 21). Metamask gibt dann wieder Tipps zur sicheren Aufbewahrung. Der User interpretiert, dass die Einrichtung abgeschlossen ist und klickt, dann auf 'Alles erledigt'.

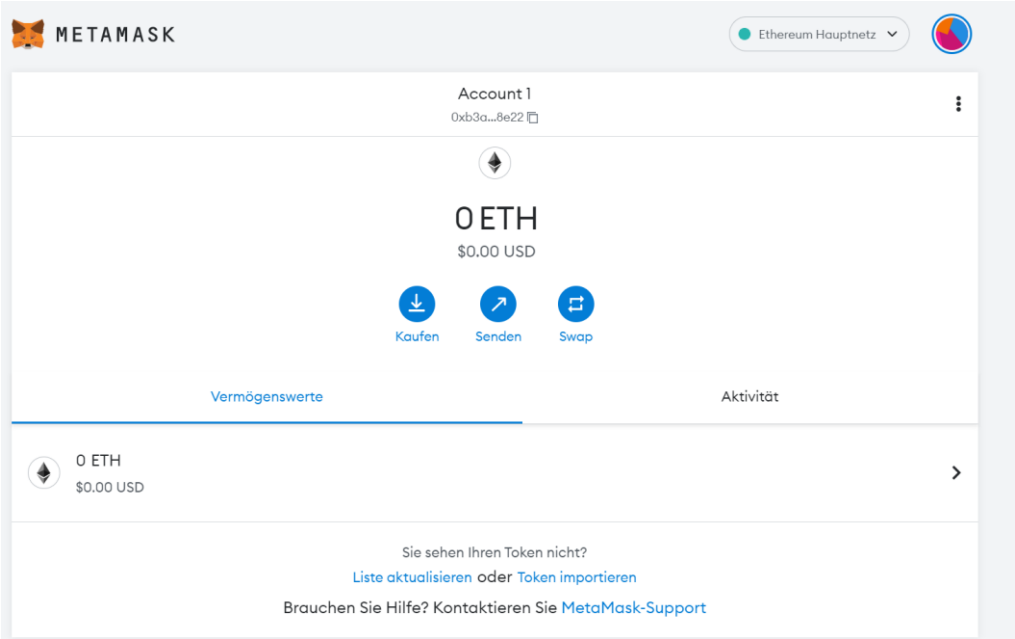


Abbildung 22

Der User ist nun in der Übersichtsseite von seinem Metamask-Wallet gelandet (Abbildung 22). Der User sieht, dass sein Account 0 ETH hat, was der User von einem neuen Wallet erwartet hat.

5. **Wahrnehmen**: Metamask auf meinem Edge Browser installiert. Neuer Wallet erstellt. Geheimen Phrase gesehen und notiert. Wallet neu. Kein Crypto drauf. Neue öffentliche Adresse.
6. **Interpretieren**: Ein neuer Wallet erstellt. Wallet ist im Ethereum Blockchain eingetragen
7. **Vergleichen**: Ziel erreicht. Neue Crypto-Wallet erstellt.

Insgesamt **3 Gulf of Executions** und **1 Gulf of Evaluations** in dieser Interaktion.

Appendix F

Swapping Crypto Coins

Swap Crypto-Coins

Der User kennt sich ein wenig aus mit Crypto und Dezentralisierte Anwendungen. Er hat schon ein Krypto-Wallet erstellt.

1. **Ziel**: Der User will seine Kryptowährungen in USDC-Kryptowährung swappen/umwandeln.
2. **Planen**: Der User hat verschiedene Optionen:
    a. In Exchanges Crypto zu USDC umwandeln. Verkauf von Kryptowährungen für USDC
    b. In Metamask Wallet Coins swappen
    c. In einer Web3 Plattform Crypto zu verkaufen (Uniswap, ..)

    Der User entscheidet sich in seiner Metamask Wallet den Swap durchzuführen.
3. **Spezifizieren**:
    a. Browser öffnen
    b. Auf Metamask klicken und mit Password anmelden
    c. Im Wallet dann auf den Swap Button klicken.
    d. Anschliessend dort Ethereum und USDC auswählen.
    e. Betrag auswählen wieviel man umtauschen möchte.
    f. Dann auf Swap klicken.
    g. Im Wallet kontrollieren, ob die Coins umgetauscht worden sind
4. **Ausführen**:
    a. Den Browser öffnen
    b. Dann auf Metamask Button klicken. Dort dann mit Password anmelden. Schliesslich auf Login klicken.
    c. Der User sieht die Übersicht seines Wallets (Abb.1). In der Mitte sind 3 Buttons platziert mit einem blauen Hintergrund. Diese Buttons fallen auf und sind zudem äusserst wichtige Buttons. ‘Kaufen’, ‘Senden’ und ‘Swap’ sind Funktionen, die sehr häufig genutzt werden. Der User sieht da sofort den Swap Button und klickt auf ihn. Es erscheint ein Pop-Up der den User fragt, ob ‘intelligente Transaktionen’ aktiviert werden soll (Abb.2). Der User liest den Inhalt kurz durch und aktiviert anschliessend die neue Funktion. Nun sieht der User die Swap Seite (Abb.3).

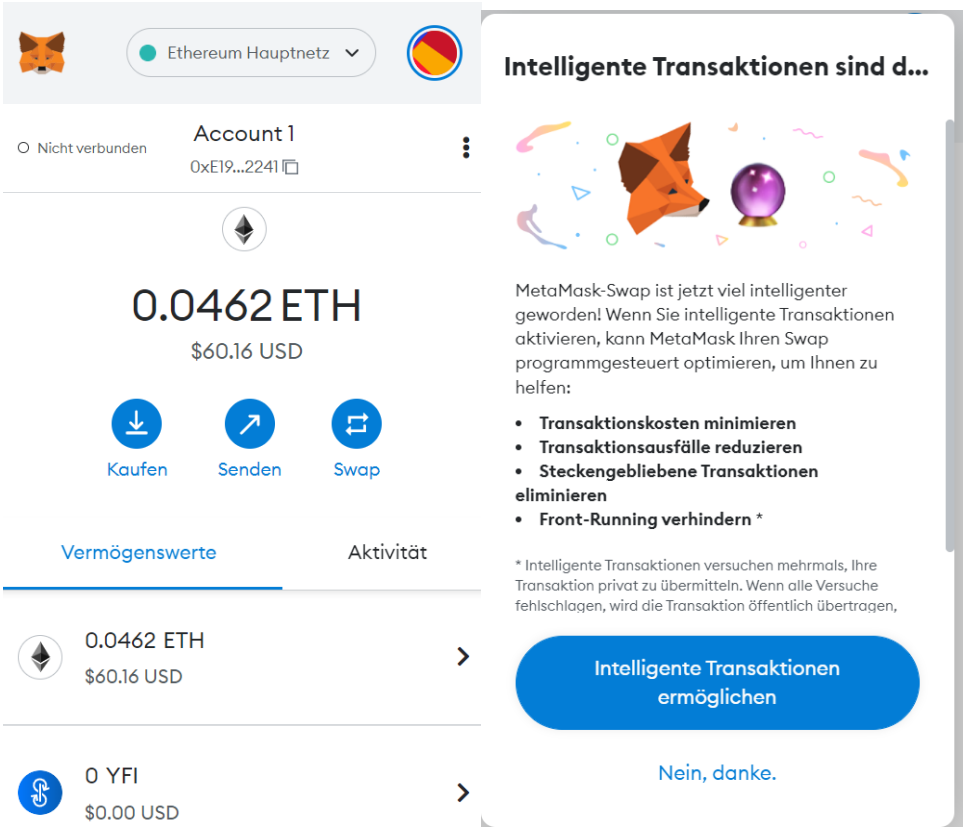


Abbildung 1 & 2

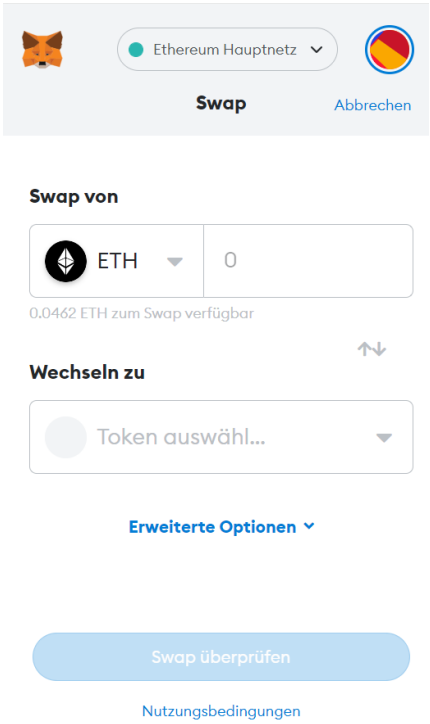


Abbildung 3

d. Der User sieht nun die Swap Seite und sieht zuoberst den Coin den er wechseln/swappen will. Da ist ETH schon ausgewählt. Da der User ein bisschen ETH zur Verfügung hat, lässt er diese Auswahl so sein. Der User klickt auf den

Balken 'Token auswählen' und schon erscheint eine Dropdown-Auswahl an Kryptowährungen (Abb.4). Kryptowährungen wie DAI, USDC, USDT und WBTC werden als erstes gezeigt, denn diese Coins sind die, die am meisten getraded/gehandelt werden. Da wurden bewusst bei der Auswahl die Anordnung nach Häufigkeit gewählt. Wenn man ein anderen Coin auswählen möchte müsste man hinunterscrollen oder den Namen in die Suchleiste eingeben. Der User klickt auf USDC. Der 'Swap' Button ist noch ausgegraut (Abb.5).

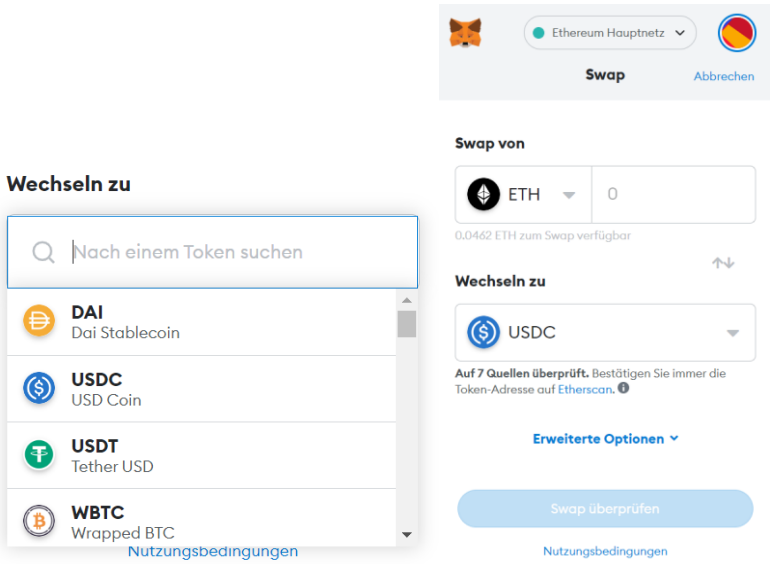


Abbildung 4 & 5

e. Der User weiss, dass er den Betrag noch eingeben muss wie viel ETH er swappen will. Der User gibt 0.01 ein und gleichzeitig sieht man rechts wie viel USD der eingegebene Betrag ungefähr wert ist. (15.45$). Dem User fällt auf, dass es keine Option hat den maximalen Betrag auszuwählen wie andere Crypto Exchanges. Das wäre ein kleiner **Gulf of Execution(1,0),** denn die bestimmte Option, die der User sich vorgestellt hatte, existiert nicht. Der User müsste dann seinen maximalen Betrag irgendwo notieren oder kopieren und dann ins Feld einfügen. Ausserdem gibt es keine Option den Betrag als USD auszuwählen, denn durch die starken Kursschwankungen von Kryptowährungen ist es mühsam mit Krypto Geld zu senden.
Der User sieht nicht direkt wie viel USDC er für den Swap bekommt (**Gulf of Evaluation(1,1)**. Der Button 'Swap überprüfen' leuchtet auf, deshalb klickt der User drauf (Abb.6). Der User weiss aber nicht so genau, wie er diesen Button interpretieren soll. Was wird überprüft? Wieso passiert diese Überprüfung nicht in 'real-time'? Das ist ein kleiner **Gulf of Execution(2,1),** denn für den User macht das nicht Sinn, wieso man nicht einfach auf 'Swap' drücken kann. Die Vorstellung von seinem Plan stimmen mit der Ausführung nicht überein. Dennoch ist dies ein kleiner Gulf of Execution, denn der User muss nicht lange überlegen, um die bestnächste Aktion zu tätigen. Dies wäre das Drücken auf den 'Swap überprüfen' Button. Nach dem Klick auf den 'Swap überprüfen' Button wird angezeigt, dass Metamask verschiedene Quellen untersucht für den besten Preis und somit wird dann auch klar, was der Button genau macht (Abb.7). Nach der Überprüfung sieht man direkt wie viel USDC man für den Swap bekommt (Abb.8). Ausserdem sieht

man oben, dass die Kurse automatisch nach einer gewissen Zeit aktualisieren. Weiter unten steht auch wie viel die ‘Gasgebühren’ sind. In Crypto ist es fast normal, dass man bei jeder Transaktion eine Gebühr zahlen muss. Aber wenn man zum ersten Mal eine Transaktion macht, kann man leicht verunsichert sein was ‘Voraussichtliche Gasgebühr’ bedeutet.
Es kann auch bedeuten wie viel man insgesamt ausgibt. Das ist ein **Gulf of Evaluation(2,2)**, denn man kann diese Information anders interpretieren. Man weiss nicht genau wie viel man insgesamt für den Swap ausgibt. In unserem Fall sind die Gebühren 0.0102, also mehr als wir eigentlich swappen möchten. Wenn man den Maus-Cursor über den Info Button bewegt kann man sehen was die Gebühren genau sind (Abb.9). Der User entscheidet sich den Swap abzubrechen, da es zu teuer ist. Das ist ein sehr grosser **Gulf of Execution(3,2)**, denn der User muss jetzt einen neuen Plan ausdenken weil sein vorheriger Plan nicht aufgegangen war.

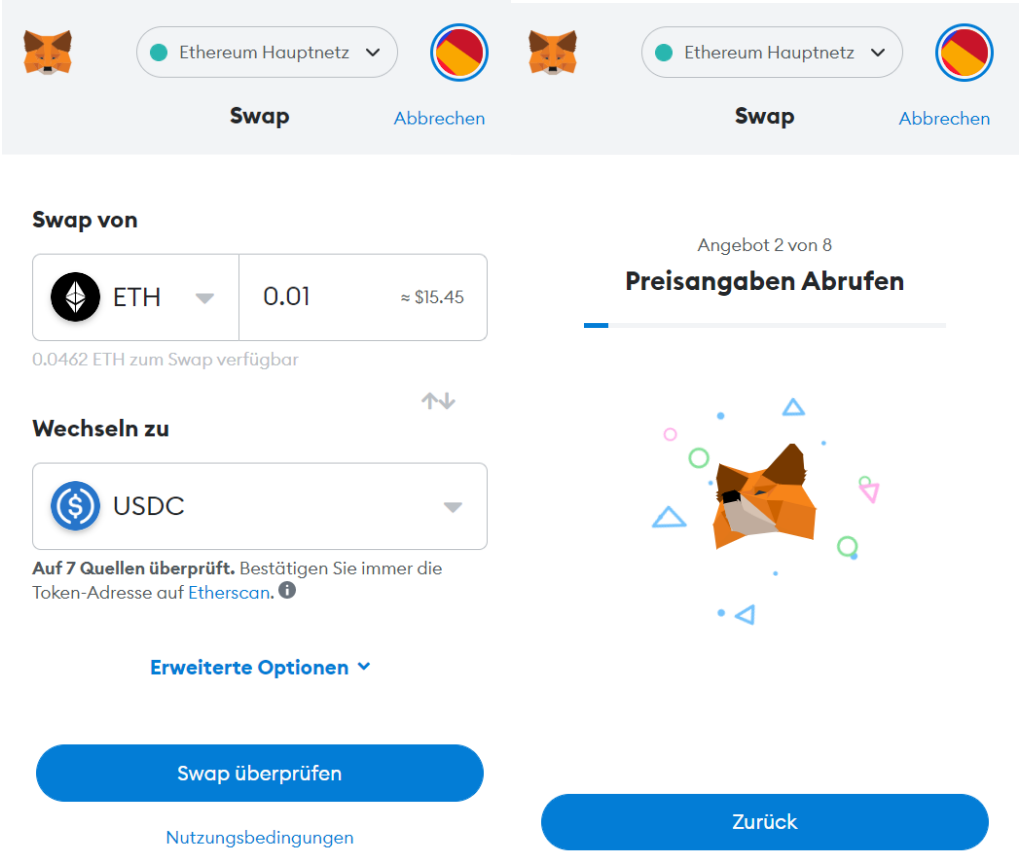


Abbildung 6 & 7

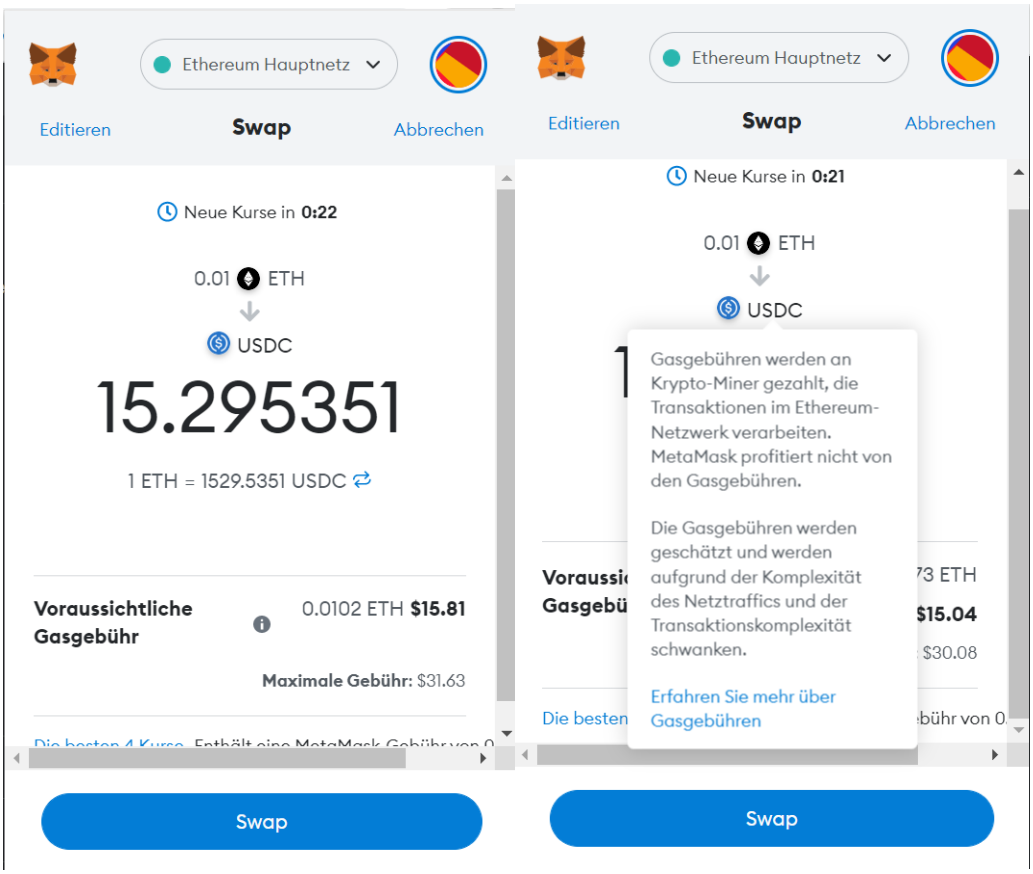


Abbildung 8 & 9

5. **Wahrnehmen**: Gasgebühren für diesen Swap sind viel. Die Kurse werden regelmässig aktualisiert. Die Gasgebühren sind nach Infobox nicht in den Händen von Metamask sondern vom Ethereum-Netzwerk.
6. **Interpretieren**: Gasgebühren zu teuer für diesen Swap. Der Grund dafür ist die darunterliegende Technologie (Ethereum).
7. **Vergleichen**: Ziel nicht erreicht. Abgebrochen von Benutzer. Gasgebühren zu teuer.

Nächste Iteration:

1. **Ziel**: Der User hat ein paar Avalanche Coins in seiner Wallet und will diese zu USDC umtauschen. Da im Avalanche Netzwerk die Gebühren weniger sind als im Ethereum Netzwerk.
2. **Planen**: Die Optionen in dieser Iteration sind ähnlich:
    a. Der User kann seine Coins mit der Metamask Wallet umtauschen. Via Swap Funktion
    b. Der User kann eine Decentralized Exchange brauchen, um den Swap zu betätigen
    c. Er kann einen Freund fragen, dass er ihm USDC sendet, wenn der User AVAX Coins mit gleichem Wert sendet

    Der User wählt die erste Option aus.

3. **Spezifizieren**:
    a. Browser öffnen
    b. Metamask öffnen

c. Passwort eingeben und einloggen
d. Swap Button finden und klicken
e. Tokens auswählen, die der User swappen möchte (AVAX & USDC)
f. Eingeben wie viel AVAX man umtauschen möchte.
g. Bestätigen und Swappen
h. Wallet überprüfen ob Swap erfolgreich

4. **<u>Ausführen</u>**:
   a. Den Browser öffnen
   b. Dann auf Metamask Button klicken.
   c. Dort dann mit Password anmelden. Man kann oben schon sehen, dass der User sein Metamask Wallet mit dem Avalanche Netzwerk geknüpft hat.(Abb.10)

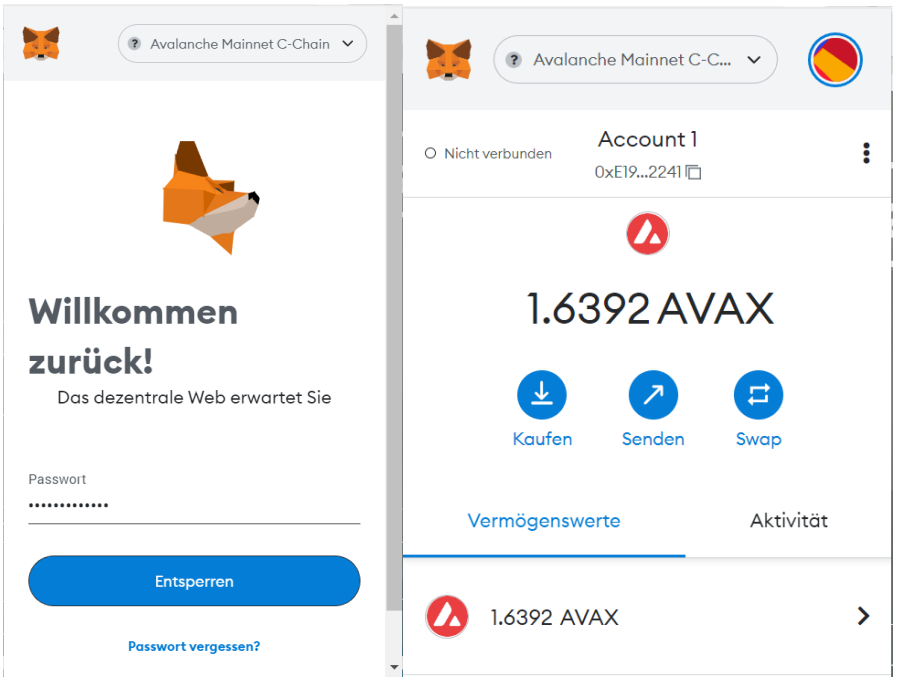


Abbildung 10 & 11

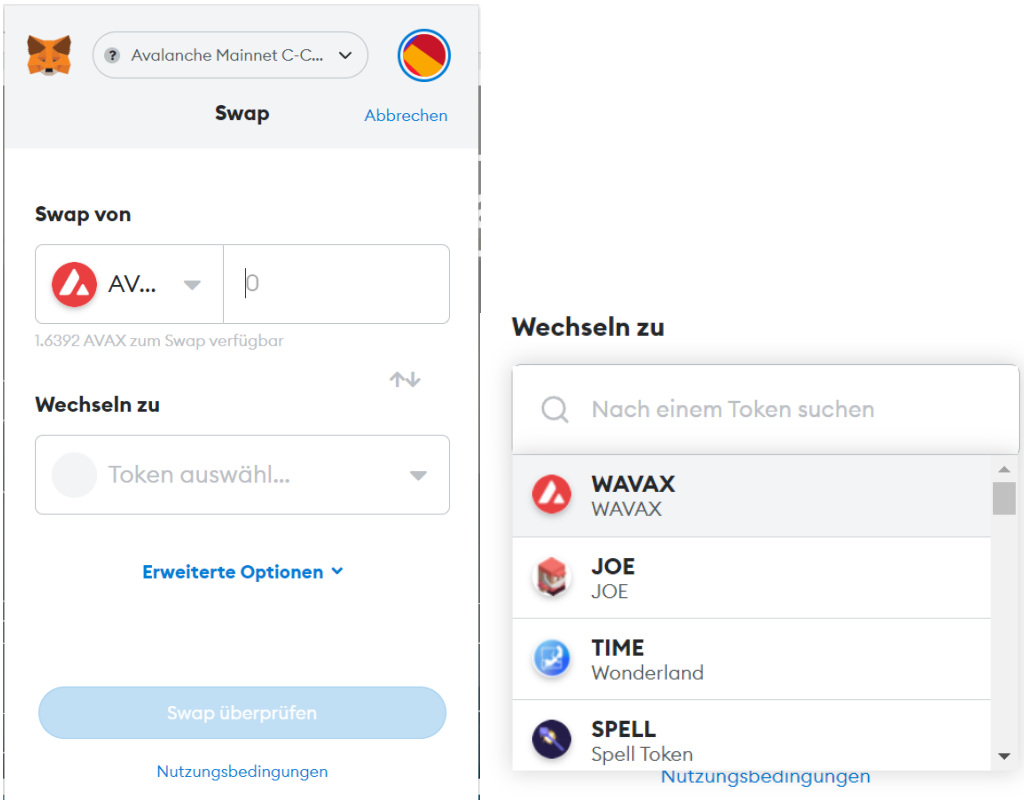


Abbildung 12 & 13

d. Dann auf den Swap Button klicken (Abb.11). Nun sieht man die Swap Seite (Abb.12). Der User will 1 seiner Avax Coins zu USDC umwandeln. Deshalb klickt er auf den 'Token auswählen' Balken und sucht den USDC Token aus (Abb.13). Er kann entweder auf 'Nach einem Token suchen' Balken 'USDC' schreiben oder er scrollt hinunter, bis er den USDC Token findet. Diesmal befindet sich USDC nicht unter den ersten Coins in der Auswahl. Wie es aussieht, sind die Optionen nicht alphabetisch sortiert. Deshalb sucht der User mithilfe des Search-Balkens. Das ist ein sehr kleiner **Gulf of Execution(4,2),** denn der User erwartet hat die gleichen Aktionen wie in der vorherigen Iteration zu tätigen.

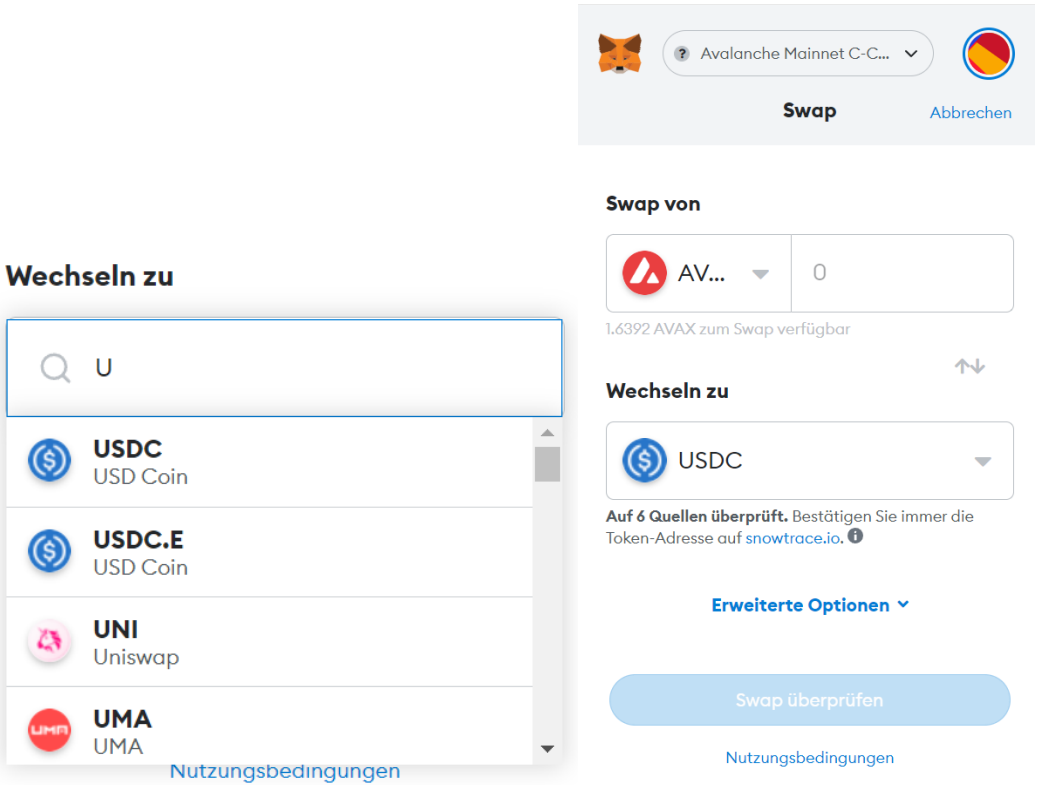


Abbildung 14 & 15

Schon mit ‘U’ findet er den gewünschten Coin (Abb.14). Doch er ist sich nicht sicher wieso es 2 verschiedene Versionen von USDC gibt. Das ist ein **Gulf of Evaluation(4,3)**. Er wählt USDC aus statt USDC.E. (Abb15) Die zweite Variante ist eine gebrückte Form von USDC vom Ethereum Netzwerk. Und die erste Version USDC für den Avalanche Netzwerk.

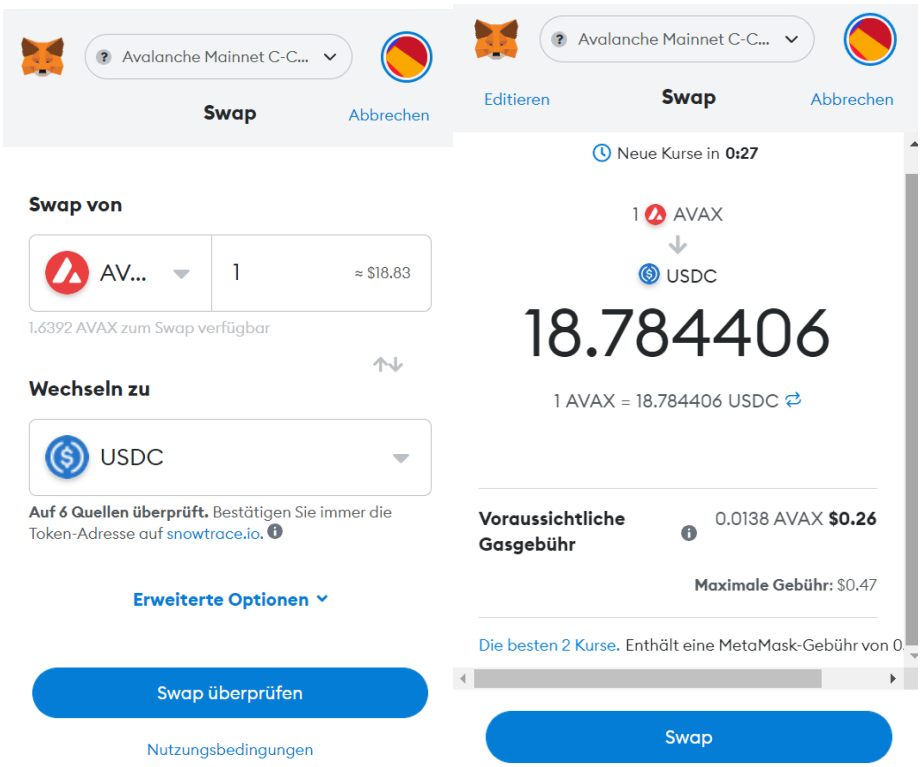


Abbildung 16 & 17

e. Der User schreibt wie viel Avalanche Coins er umtauschen will (Abb.16). Er wählt 1 da er nur 1.6392 AVAX zu verfügung hat. 1 AVAX ist ungefähr 18.83$ Wert. Er sieht nicht direkt wie viel USDC das machen würde. Dann klickt er auf 'Swap überprüfen' und sieht dass er ungefähr 18.784406 USDC bekommen würde (Abb.17). Dieser Kurs ändert sich ständig und es wird oben angezeigt wann der Kurs aktualisiert wird. Man sieht auch das die voraussichtlichen Gasgebühren etwa 0.01348 AVAX oder 0.26$ kosten.

f. Danach klickt er auf den Swap Button.
Es wird angezeigt das die Transaktion verarbeitet wird (Abb.18) und ungefähr nach 3 Sekunden wurde die Transaktion vollständig und die 18.7844 USDC wurden dem User gutgeschrieben (Abb.19).

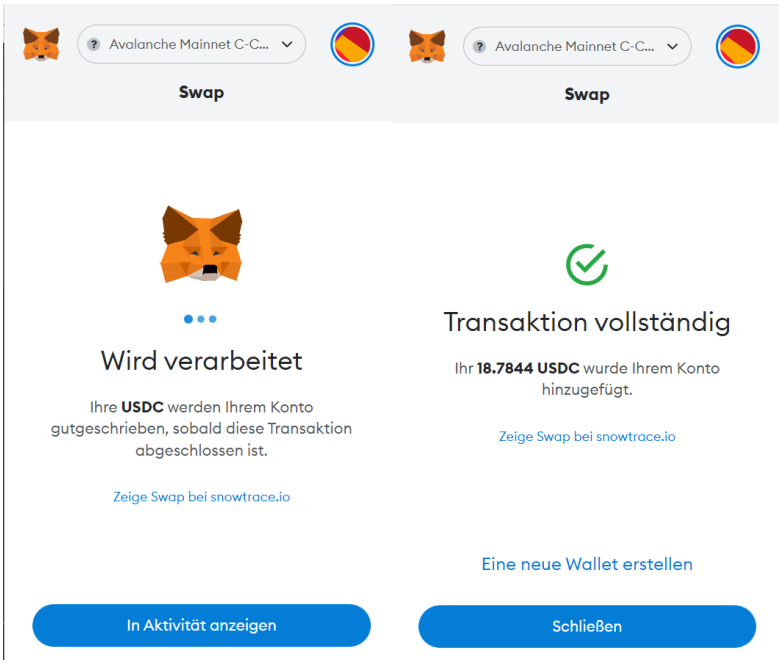


Abbildung 18 & 19

Um die Transaktion zu verfolgen kann man auf ‘Zeige Swap bei snowtrace.io’ klicken. Diese Seite zeigt dann alle Transaktionen im Blockchain an, die für den Swap benötigt wurden. (Abb.20) (Abb.21)

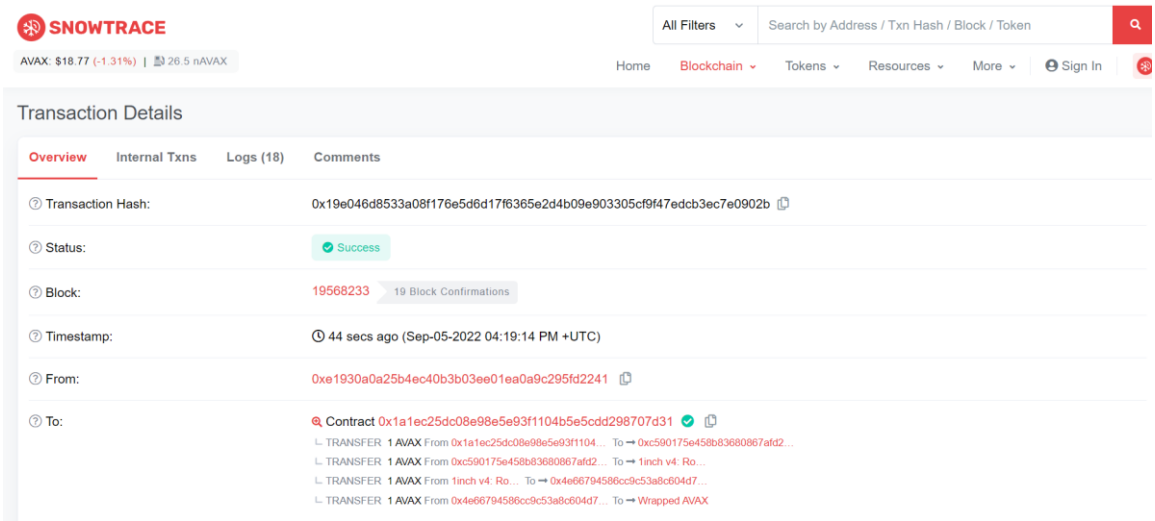


Abbildung 20

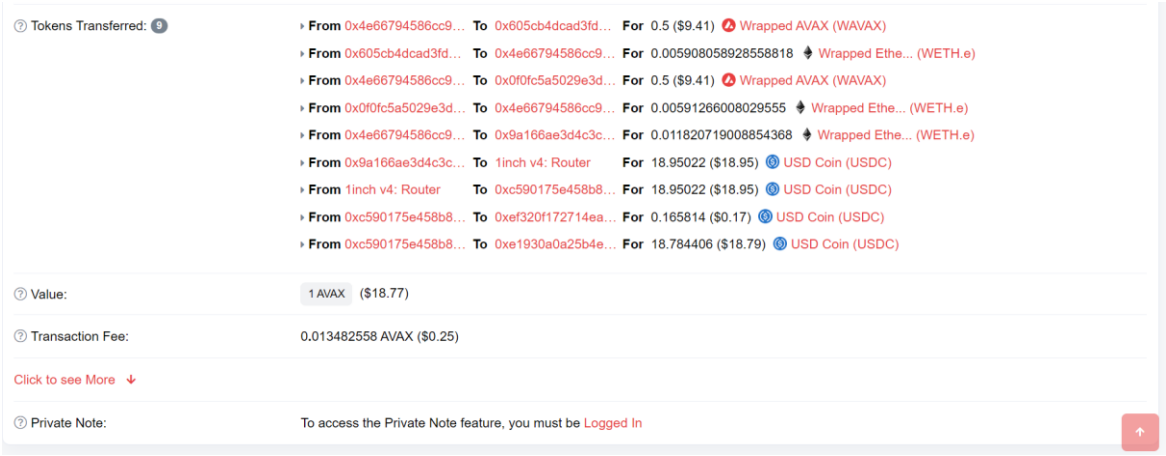


Abbildung 31

Hier kann man genau sehen was mit dem 1 AVAX passiert beim Swap. Und beim Status kann man erkennen, dass die Transaktion funktioniert hat.

g. Danach klickt er auf ‘Schliessen’ und landet in der Übersichtsseite. Und da kann man sehen, dass die USDC wirklich gutgeschrieben wurden. (Abb.22)

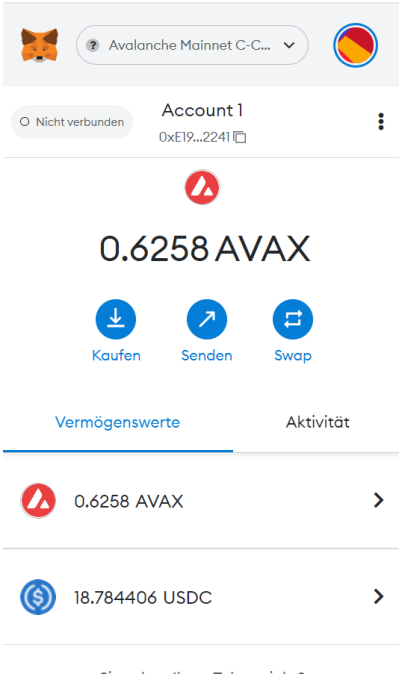


Abbildung 22

5. **<u>Wahrnehmen</u>**: Die gewünschte Coins sind jetzt im Wallet. Metamask hat 6 Quellen überprüft für den bestmöglichen Kurs.
6. **<u>Interpretieren</u>**: Der Swap hat eins von meinen AVAX Coins genommen und mir 18.78 USDC gesendet. Metamask
7. **<u>Vergleichen</u>**: Der Swap hat funktioniert. Um die Transaktion genau zu überprüfen wird ein Background von Blockchain nötig. Aber der Status der Transaktion ist für jeden User einfach zum Verstehen.

Insgesamt **4 Gulf of Executions** und **3 Gulf of Evaluations** in dieser Interaktion.

Appendix G

Zalando

Zalando Bezahlstatus überprüfen

Der User ist ein jahrelanger Kunde bei Zalando. Um Produkte zu bestellen, braucht er immer die Zalando App auf dem Smartphone.

1. **Ziel**: Der User hat vor ein paar Wochen etwas von Zalando bestellt. Jetzt will er überprüfen, ob er seine Zalando Rechnung eigentlich schon bezahlt hat.
2. **Planen**: Seine Optionen sind:
    a. In der Zalando App nachschauen, ob seine Bestellung bezahlt wurde
    b. In den Paketen nachschauen, ob es noch offene Rechnungen gibt
    c. In der UBS-App nachschauen ob Zahlungen betätigt worden sind

    Der User hat sich für die erste Variante entschieden.

3. **Spezifizieren**:
    a. Zalando App öffnen
    b. In der App zum Menü gehen und dann auf Bestellungen klicken
    c. Und dort nachschauen, ob die Bestellung bezahlt wurde
4. **Ausführen**:
    a. Am Smartphone auf den Zalando App klicken
    b. Der User sieht die Startseite von Zalando (Abbildung 1). Oben befindet sich die Navigationsleiste mit 5 Icons. Die erste ist ein Icon mit dem Zalando Logo was der User als Homepage interpretiert. So ein Homepage Button ist gewöhnlich immer ganz links oben platziert. Die User sind also gewöhnt, dass der Button, der zu der Homepage führt, immer oben links ist. Der zweite Button ist für die Suchfunktion und Kategorie Anzeige, was man am Logo erkennen kann. Der dritte Button ist für die Favoriten des Users, was leicht erkennbar am Icon ist und es hat zusätzlich eine Nummer oberhalb des Icons, was damit sagen will, wie viele Produkte der User favorisiert hat. Zu guter Letzt ist ganz rechts der Profil Button. Der User klickt auf diesen Button, denn er glaubt, dass seine Bestellungen dort im Profil sichtbar sind. Nach dem Klick öffnet sich das Profil-Menü (Abbildung 2). Hier kann der User vieles Einstellen. Seine Bestellungen findet er schon bei der ersten Unterkategorie ‘Bestellungen und Rücksendungen’. Der User klickt auf diesen Button.

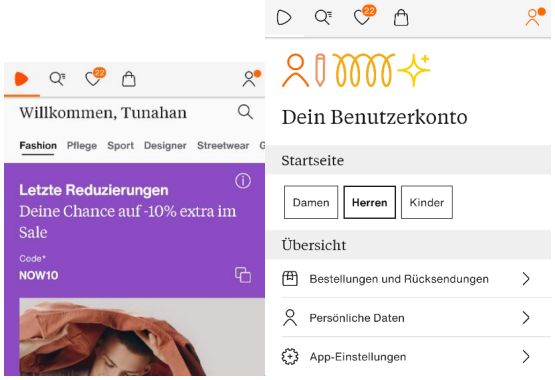


Abbildung 1 & 2

c. Der User will jetzt seine letzte Bestellung ansehen. Oben wird glücklicherweise direkt die letzte Bestellung angezeigt (Abbildung 3). Er sieht bei dieser Bestellung wie viel die Bestellung gekostet hat, wie viele Artikel er bestellt hat und wann sie bestellt wurde. Es ist nicht ersichtlich, ob die Bestellung schon bezahlt wurde (**Gulf of Execution(1,0)** und **Gulf of Evaluation(1,1)**). Auf der Seite ist nicht klar ersichtlich, ob es eine detaillierte Sicht für die Bestellung existiert. Aber der User sieht einen kleinen Pfeil nach rechts. Das ist ein Signifier und dieser Pfeil macht den User aufmerksam, dass man auf die Bestellung drücken kann, um eine detaillierte Sicht zu sehen für die Bestellung. Dieser Button wurde bewusst nicht wie ein Button designt, um darauf zu klicken. Denn für das wäre der Button viel zu klein. Der User klickt dann auf seine Bestellung. In dieser Sicht sieht der User jetzt detaillierte Informationen über seine Bestellung (Abbildung 4). Zuoberst sieht er, dass die Bestellung versandt wurde und an welche Adresse. Der User sieht dann, dass seine Bestellungen in 2 Pakete unterteilt worden ist. Bei beiden Paketen kann man die Sendung verfolgen. Dieser Button hat eine orangene Schrift was den Button von den anderen Elementen abhebt. Dennoch hat der User nicht gesehen, ob die Bestellung bezahlt wurde oder nicht (**Gulf of Execution(2,1)** und **Gulf of Evaluation(2,2)**). Der User probiert dann alles anzuklicken, was man hier alles anklicken kann. Trotzdem hat er nichts gefunden, und es gibt kein Feedback wo der User suchen muss für diese wichtige Information (**Gulf of Evaluation(2,3)**). Das ist auch ein grosser **Gulf of Execution(3,3)**, denn der User ist in einer Sackgasse und muss jetzt einen neuen Plan machen.

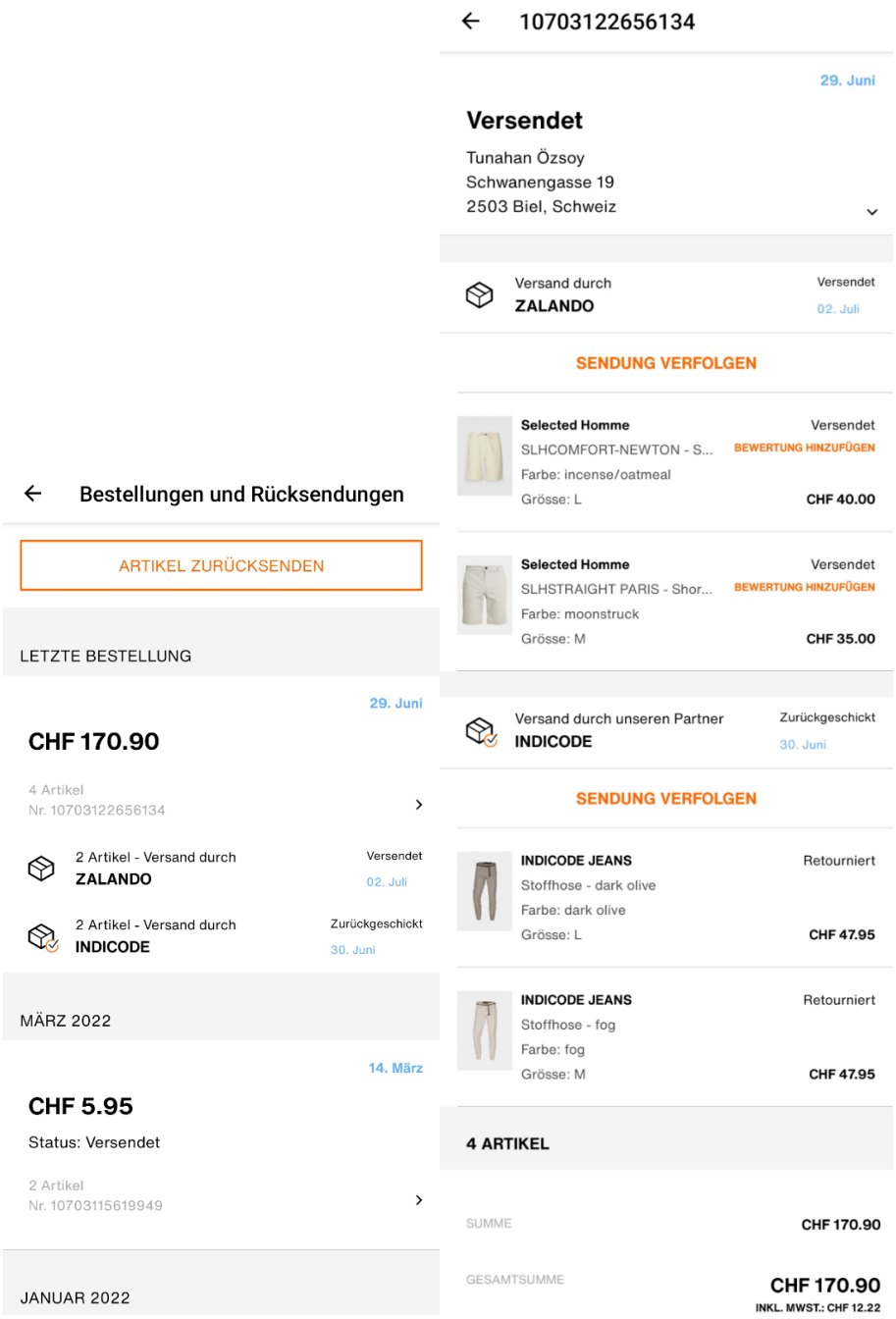


Abbildung 3 & 4

5. **<u>Wahrnehmen</u>**: Man sieht, dass die Bestellung versendet worden ist und ein paar Artikel retourniert. Man sieht auch die Gesamtsumme der Bestellung aber nicht, ob sie schon bezahlt wurde.
6. **<u>Interpretieren</u>**: Der Bezahlstatus ist nicht in der App ersichtlich. Vielleicht aber in der Webversion also auf dem Browser.
7. **<u>Vergleichen</u>**: Der Bezahlstatus ist nicht in der App ersichtlich. Ziel nicht erreicht. Nächstes Mal in der Webversion ausprobieren

Nächste Iteration:

1. **Ziel**: Der User hat vor ein paar Wochen etwas von Zalando bestellt. Jetzt will er überprüfen, ob er seine Zalando Rechnung eigentlich schon bezahlt hat.
2. **Planen**: Seine Optionen sind:
    a. Auf der Zalando Webseite nachschauen, ob seine Bestellung bezahlt wurde
    b. In den Paketen nachschauen, ob es noch offene Rechnungen gibt
    c. In der UBS-App nachschauen ob Zahlungen betätigt worden sind

    Der User hat sich für die erste Variante entschieden.
3. **Spezifizieren**:
    a. Browser öffnen
    b. Auf dem Browser in das Suchfeld 'Zalando' eingeben
    c. Auf den richtigen Link klicken
    d. In Zalando einloggen
    e. Ins Profil gehen und dort dann auf Bestellungen
    f. Dort auf die gewünschte Bestellung anklicken und schauen ob bezahlt oder nicht
4. **Ausführen**:
    a. Browser öffnen
    b. Am Browser nach Zalando suchen in Google.
    c. Dort auf den ersten Link anklicken. Also zalando.ch (Abbildung 5)

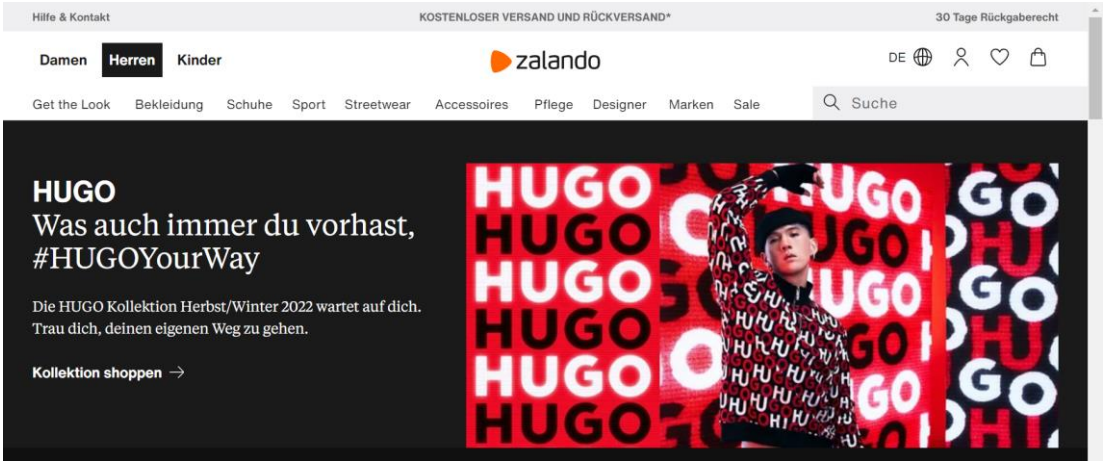


Abbildung 5

    d. Hier in der Webversion ist die Anordnung der Buttons anders. Das Logo ist in der Mitte, die zu der Homepage zurückbringt. Der User klickt auf den Profilsymbol und auf den Anmelden Button (Abbildung 6). Dieser Button hat einen hohen Affordance anzuklicken, wegen dem schwarzen Hintergrund, dass diesen von den anderen Buttons sehr abhebt. Dann landet der User zum Login-Page. Email und Passwort werden automatisch eingegeben, denn der User hatte seine Anmeldedaten gespeichert. Der User klickt dann auf Anmelden (Abbildung 7).

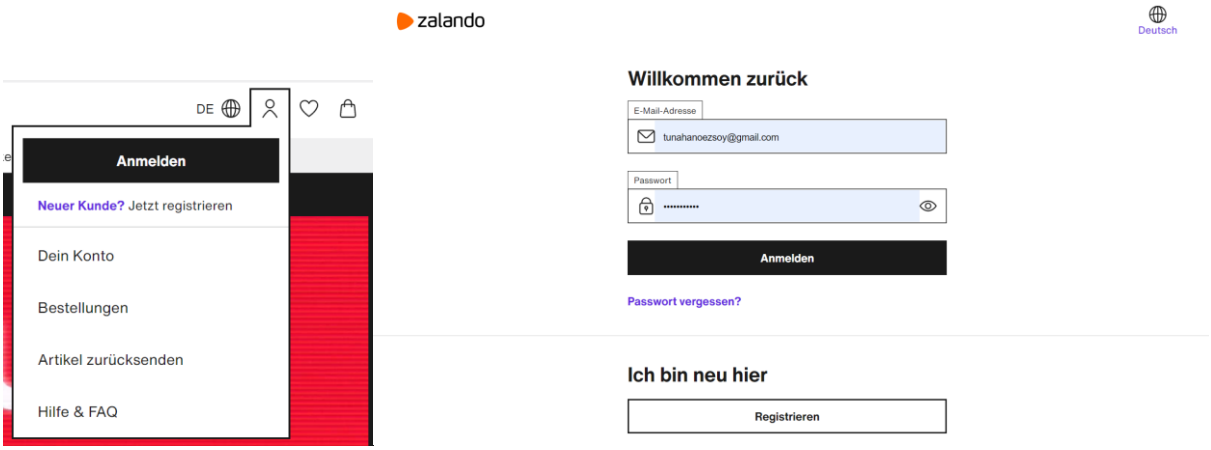


Abbildung 6 & 7

e. Jetzt erscheint wieder die Startseite. Der User will zu seinen Bestellungen navigieren. Er sieht den Profil-Symbol oben rechts und bewegt mit der Maus drauf. Dann auf Bestellungen draufklicken (Abbildung 8). Die gewünschte Bestellung aussuchen. (Abbildung 9)

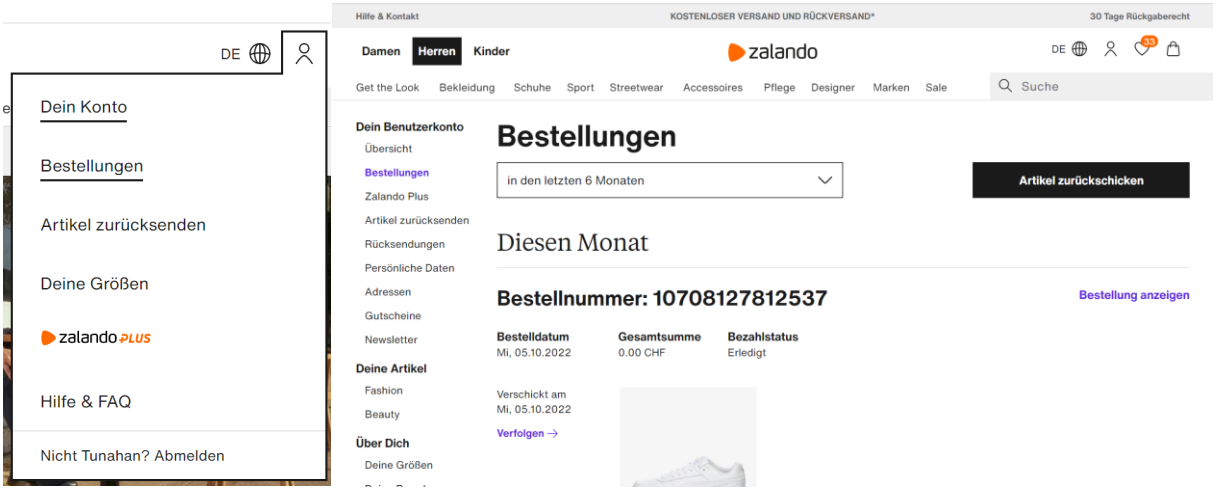


Abbildung 8 & 9

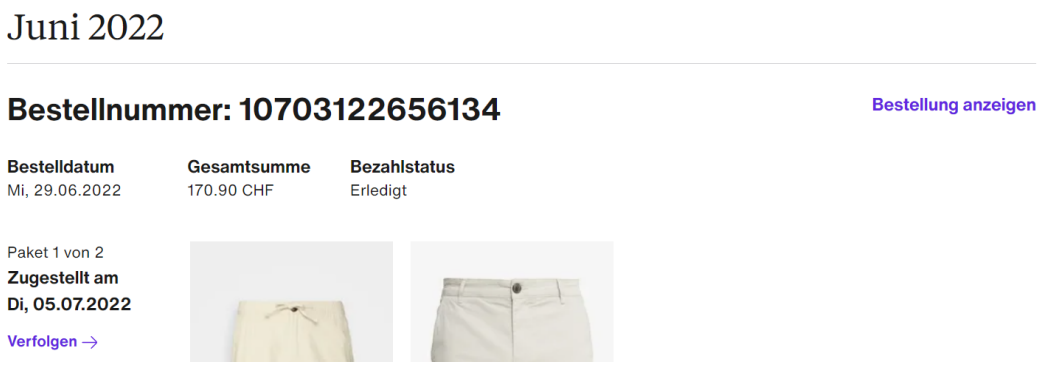


Abbildung 10

f. Der User muss nicht einmal auf die Bestellung draufklicken, denn er sieht direkt, dass der Bezahlstatus von dieser Bestellung erledigt ist. (Abbildung 10)

5. **Wahrnehmen**: Wenn man auf Bestellungen geht, sieht man alle Bestellungen und für jede Bestellungen hat es mehr Informationen als auf dem App.
6. **Interpretieren**: Auf der Website hat es mehr Informationen als auf der App.
7. **Vergleichen**: Der Bezahlstatus ist nun ersichtlich. Für die Bestellung ist es sogar erledigt. Der User weiss jetzt, dass er die Rechnung bezahlt hat. Ziel erreicht.

Insgesamt **3 Gulf of Executions** und **3 Gulf of Evaluations** in dieser Interaktion.

Erklärung

gemäss Art. 30 RSL Phil.-nat.18

Name/Vorname: Özsoy Tunahan

Matrikelnummer: 16-116-840

Studiengang: Informatik

Bachelor ☑ Master ☐ Dissertation ☐

Titel der Arbeit: Uncovering Gulf of Execution and Gulf of Evaluation in User-Computer Interactions -- A Comparative Study between Manual Analysis and Questionnaire-Based Approach

LeiterIn der Arbeit: PD Dr. Kaspar Riesen

Ich erkläre hiermit, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls der Senat gemäss Artikel 36 Absatz 1 Buchstabe r des Gesetzes vom 5. September 1996 über die Universität zum Entzug des auf Grund dieser Arbeit verliehenen Titels berechtigt ist.
Für die Zwecke der Begutachtung und der Überprüfung der Einhaltung der Selbständigkeitserklärung bzw. der Reglemente betreffend Plagiate erteile ich der Universität Bern das Recht, die dazu erforderlichen Personendaten zu bearbeiten und Nutzungshandlungen vorzunehmen, insbesondere die schriftliche Arbeit zu vervielfältigen und dauerhaft in einer Datenbank zu speichern sowie diese zur Überprüfung von Arbeiten Dritter zu verwenden oder hierzu zur Verfügung zu stellen.

2503 Biel, 14.6.2023

Ort/Datum

Unterschrift

Bibliography

[1] Thomas T. Hewett, Ronald Baecker, Stuart Card, Tom Carey, Jean Gasen, Marilyn Mantei, Gary Perlman, Gary Strong, and William Verplank. Acm sigchi curricula for human-computer interaction. Technical report, New York, NY, USA, 1992.

[2] Peter Budnick and Rachel Michael. What is Cognitive Ergonomics? Online: `https://web.archive.org/web/20110928150026/http://www.ergoweb.com/news/detail.cfm?id=352`, 2001. [Accessed 10-February-2023].

[3] Backgrounder on the Three Mile Island Accident. Online: `https://www.nrc.gov/reading-rm/doc-collections/fact-sheets/3mile-isle.html`. [Accessed 10-February-2023].

[4] Report of The Presidents Comission On THE ACCIDENT AT THREE MILE ISLAND. Online: `https://web.archive.org/web/20110409064628/http://www.threemileisland.org/downloads/188.pdf`, April 2011. [Accessed 10-February-2023].

[5] B. Shneiderman. Direct manipulation: A step beyond programming languages. *Computer*, 16(8):57–69, aug 1983.

[6] Donald A Norman and Stephen W Draper, editors. *User centered system design*. Lawrence Erlbaum Associates, Mahwah, NJ, January 1986.

[7] Don Norman. *The design of everyday things*. Basic Books, London, England, 2 edition, November 2013.

[8] interaction-design.org. Gulf of Evaluation and Gulf of Execution. Online: `https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction/gulf-of-evaluation-and-gulf-of-execution`. [Accessed 10-February-2023].

[9] Donald A Norman. *The psychology of everyday things.* Basic books, 1988.

[10] Jo Vermeulen, Kris Luyten, Elise van den Hoven, and Karin Coninx. Crossing the bridge over norman's gulf of execution: Revealing feedforward's true identity. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 1931–1940, New York, NY, USA, 2013. Association for Computing Machinery.

[11] Alan Cooper, Robert Reimann, David Cronin, and Christopher Noessel. *About face: the essentials of interaction design.* John Wiley & Sons, 2014.

[12] Anne-Marie Feyer and Ann M Williamson. Human factors in accident modelling. *Encyclopaedia of Occupational health and safety*, 4, 1998.

[13] Janni Nielsen, Torkil Clemmensen, and Carsten Yssing. Getting access to what goes on in people's heads? reflections on the think-aloud technique. In *Proceedings of the Second Nordic Conference on Human-Computer Interaction*, NordiCHI '02, page 101–110, New York, NY, USA, 2002. Association for Computing Machinery.

[14] M.J. van den Haak and M.D.T. de Jong. Exploring two methods of usability testing: concurrent versus retrospective think-aloud protocols. In *IEEE International Professional Communication Conference, 2003. IPCC 2003. Proceedings.*, pages 3 pp.–, 2003.