Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering

Integration and Evaluation of Explanations in the Context of a Navigation App

Integration und Evaluation von Erklärungen im Kontext einer Navigationsapp

Bachelorarbeit

im Studiengang Informatik

von

Zhongpin Wang

Prüfer: Prof. Dr. Kurt Schneider
Zweitprüfer: Dr. Jil Klünder
Betreuer: Larissa Chazette

Hannover, 30.07.2020

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 30.07.2020

_____

Zhongpin Wang

Zusammenfassung

Integration und Evaluation von Erklärungen im Kontext einer Navigationsapp

Die Erklärbarkeit wird mit zunehmender Komplexität von Softwaresystemen immer wichtiger. Das Fehlen notwendiger Erklärungen kann die Benutzerfreundlichkeit schaden, die Softwaretransparenz verringern, das Vertrauen der Benutzer negativ beeinflussen und andere nicht funktionale Anforderungen (NFRs) beeinträchtigen. Der Trade-off-Effekt zwischen Erklärbarkeit und anderen NFRs sollte sorgfältig abgewogen werden, um User-Experience (UX) zu optimieren und zu vermeiden, dass andere Softgoals verletzt werden. Daher ist es notwendig, den Bedarf des Benutzers an Erklärungen zu ermitteln und deren Auswirkungen auf die anderen NFRs und UX zu analysieren. Darüber hinaus sollten Experimente durchgeführt werden, um die geeignete Art der Darstellung von Erklärungen und deren Granularität zu verstehen. Als Testobjekt in dieser Studie wurde eine Navigations-App für Android entwickelt, die wesentliche Funktionen wie Ortssuche, Routenempfehlung und Echtzeitnavigation enthält. Zwanzig Teilnehmer nahmen separat an einem synchronen Remote-Usability-Test teil. Sie gaben ihr Feedback zu allen drei Versionen der App, das drei verschiedenen Granularitätsstufen von Erklärungen entsprach: keine, kurze und detaillierte Erklärungen. Insgesamt zeigen die Ergebnisse, dass Erklärungen notwendig sind, sollte jedoch sorgfältig entworfen werden, um die Systemqualität zu unterstützen.

Abstract

Integration and Evaluation of Explanations in the Context of a Navigation App

Explainability is receiving more and more attention as the complexity of software systems grows. The lack of necessary explanations may reduce the usability, decrease software transparency, impact negatively on users' trust, and impair other non-functional requirements (NFRs). The trade-off effect between explainability and other NFRs should be considered carefully to optimize the user experience (UX) and avoid hurting other soft goals. Therefore, it is necessary to determine the user's demand for explanations and analyzing their impact on the other NFRs and UX. Furthermore, experiments should be conducted to understand the appropriate way to present explanations and their granularity level. As the test object in this study, a navigation app for Android was developed, containing essential functions such as place search, route recommendation, and real-time navigation. Twenty participants joined a synchronous remote usability test separately. They gave their feedback on all three versions of the app, corresponding to three different granularity levels of explanations: no, brief and detailed explanations. Overall, the results show that explanations are necessary, but should be carefully designed to support system quality.

Contents

Chapter 1

Introduction

Nowadays, software systems have penetrated various industries and the daily life of people. The complexity of software systems has been increased rapidly by using hard explainable algorithms like machine learning or embedding a knowledge-based system. One of the earliest works related to explanations in intelligent system is published by Gregor and Benbasat [1]. Recently, researchers have tried to study the influence of explanation in intelligent system like explainable artificial intelligence (XAI) [2]. The aim is to improve the system's transparency [3], the user's trust [4], and other associated non-functional requirements (NFR) such as usability, understandability and so on [5].

For some specific scopes of application, the system must provide explanations. For instance, if the creditors in the United States take action, applicants must be notified with specific reasons, according to §1002.9(b)(2) [6]. Another example is the *General Data Protection Regulation* (GDPR). Recital 71 specifically points out that the data subject should have the right to obtain an explanation of the decision reached [7]. Both cases require absolute transparency of the software systems to make the automated decision-making process trustworthy and reliable.

While the need for explanation is growing, it was important to study its relations with other soft goals. Some researchers have been investigating explainability as an NFR [8, 9]. The advantage of using the NFR framework is that the impact of explainability on other soft goals can be built based on the existing knowledge. However, few experiments were conducted to study the relations and the end-users' needs for explanations in software systems.

Therefore, a navigation app for android is developed to enable the analysis of the effects of explanations through a real experiment. The app contains essential functions like place search, route recommendation, real-time navigation, and so on. The reason for choosing the navigation app is that possible recommendations can be made, and the system itself is complex enough to test the user's immediate reaction while using a high fidelity app.

The app is designed with three versions supporting different granularity levels of explanations: **no**, **brief** and **detailed** explanation.

The experiment is conducted in the form of a synchronous remote usability test (sRUT), in which the participant and evaluator are geographically separated and connected through an online meeting program. 20 university students were recruited and finished the one-hour session test. The impacted NFRs were derived by extracting and classifying the related codes from the participants' responses.

The results show that receiving explanations is perceived as useful, but should be provided cautiously in proper forms and granularity. Moreover, software developers need to guarantee the quality of the explanations to make them useful for the targeted user groups, easy to understand, and concise enough without hurting the interface.

Moreover, the results indicate that the participants had different needs for the explanations before and after testing all three versions. Thus, the system should provide explanations gradually along with the user's learning process, instead of explaining all at once at the beginning.

By using the transparency SIG [10], the impact of explainability on system transparency can be analyzed. The related NFRs were used as the intermediary between explainability and transparency. Figure 1.1 illustrate the influences and the analyzing approach.

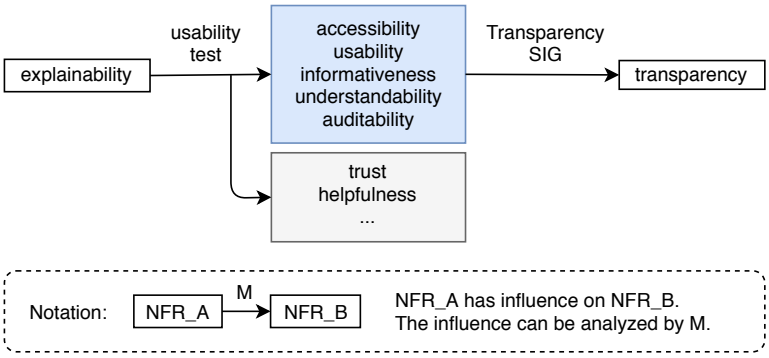


Figure 1.1: The influence of explainability on transparency and other NFRs

The conclusion is that more explanations do not always mean better transparency. Not only should the developers consider the quality, form, and granularity of the explanations, but also the dynamic learning process of the users. It also needs to be discussed whether software transparency should be taken as an intrinsic characteristic of a system, or be assessed considering the end-users' perception as well.

In the next Chapter 2, the related work is presented along with the definition of explanation, trust, and transparency. Chapter 3 defines the research goal and questions in this work and describes briefly about the app design concepts. Some terms related to the app features are defined and

unified to help understand the results later. The setup and design of the usability test is also introduced. The results for all three versions and their reasoning are documented in Chapter 4. Afterward, Chapter 5 discusses the topics extracted from the results and provides some practical instructions based on this study. Chapter 6 describes the possible limitations of this work and lists out the potential threats that may influence the validity of the results. The last Chapter 7 concludes the work.

Chapter 2

Background and Related Work

In this section, the related work and the definitions of explanation, trust, and transparency, and their relationships are presented.

2.1 What is an explanation?

First, some concepts needed to be clarified. The term *interpretability* is sometimes considered as synonym of *explainability*. Doshi-Velez and Kim [11] defined *interpretability* as "the ability to explain or to present in understandable terms to a human" in the context of ML systems. They treated explainability and interpretabilty as the same concept. On the other hand, Tomsett et al. [12] defined *explainability* as a concept from the system's side to provide clarification. Meanwhile, *interpretability* is from the users' side to interpret the explanations they perceived. Same as mentioned by Chazette and Schneider [9], this study takes *interpretability* as a subjective aspect, and *explainability* as an objective aspect. Furthermore, the concept interpretation is not only limited to the topic explanation but also refers more generally to the processing of the perception in users' mind, described as part of the seven stages of action by Norman [13]. Figure 2.1 is the modified version of the graph including explanation and interpretation. In the context of software engineering, the explanation is provided by the system, which is part of "The World," thus it is objective. Interpretation happens in the user's mind after they have perceived explanation, and therefore it is subjective.

With the rapid development of artificial intelligence, more and more researchers began to study eXplainable AI (XAI) [2, 14]. Adadi and Berrada [15] discussed the definition of XAI and the need for such an explainable system. They pointed out the technical challenges and current research limitations on the explainable intelligent systems. Most of the existing intelligent systems are just one type of AI. The fields of explainable AI planning [16] and explainable agent [17, 18] are receiving more and more attention. However, most of the recent studies related to explainability focus

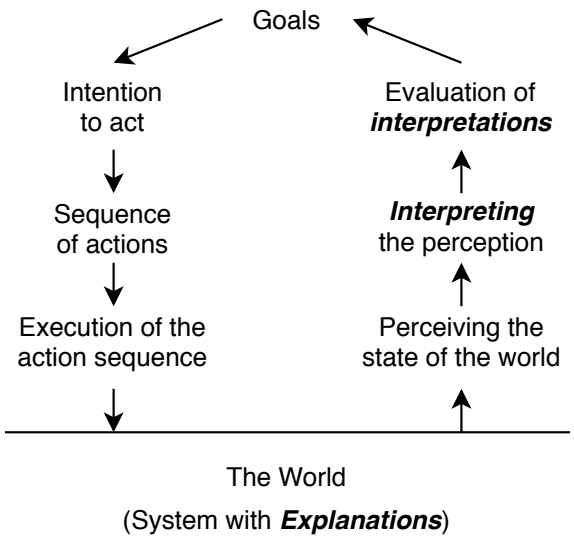


Figure 2.1: Explanation and interpretation in Norman's seven stages of action

on the possibility of explaining an inner mechanism of a system implementing ML. But explainability is only a part of the explanation concept. For example, an explanation could also refer to a description of a functionality or a guide of the use. More generally, explanations are the answers to the questions that could be raised from the end-users [8]. Normally, these questions are formulated in the *why* question form, but they could also be *what* or *how* questions and so on. Assume that a user asked, "How does this recommendation algorithm work?". It is equivalent to the question, "Why am I receiving this recommendation?" Therefore, the interrogative words, especially *why*, are not the criterion of classifying the need for explanations, because explanation and the question form is a complicated linguistic and philosophy problem [19].

Nevertheless, users and developers often have different mental models [20]. Software engineers should ask the question: Do end-users also treat the specific information as explanations according to their mental model, or they just perceive them as some extra information that might be needed? For instance, if the navigation app suggests an unusual route instead of the usual one, and marks the usual way with a congestion sign, will the user interpret it as an explanation for the recommendation, or just as extra information that indicates the traffic condition? The answer depends on whether users have raised the questions referring to the mechanism or not. Furthermore, instead of using the congestion sign, the system could provides explicit text like "Although this route is unusual, it is faster considering the current traffic condition on the usual route." This information could be

more easily interpreted as an explanation. Thus, one cannot assure that the information is, by all means, considered as an explanation.

The definitions from Köhl et al. [8] and Tomsett et al. [12] are based on the condition that the user does interpret the information as an explanation. Due to technical constraints, system can currently only explain actively and try to predict the users' needs. An optimal solution would be having an intelligent agent, which can interactively answer all ad-hoc questions asked by end-users.

In this thesis, explanation is defined as a type of information, for which a user may ask to help understand explanandum.

2.1.1 Explainability as an NFR

Köhl et al. [8] proposed to treat explainability requirement as a non-functional requirement to satisfice rather than satisfy it. Besides, it has an impact on the other soft goals. Thus, it exists interdependence between explainability and other NFRs.

Chazette and Schneider [9] performed an online survey to study the impact of explainability on usability and UX. The result shows the double-edged sword effect of explainability on the other NFRs like usability and informativeness. Based on their work, it is easier to understand explainability and its impact on the known NFR framework and improve the UX following the user-centered design (UCD) principle.

2.2 Transparency and Trust

2.2.1 The Relations among Explainability, Transparency and Trust

The interactions among explainability, transparency, and trust have attracted the attention of many researchers from different fields.

Kizilcec [21] studied the trade-off effect between transparency and trust by providing different types of explanations in an online assessment platform: no explanation, a purely procedural explanation, and explanation with additionally providing data. The no explanation version provides only the result of the assessment. The purely procedural explanation version provides description of assessment process. The most detailed version provides also detailed data used in the assessment. The result shows that "expectation violation reduced trust overall, but interface transparency moderated this effect, such that providing some transparency with procedural information fostered trust, while additional information about outcomes nullified this effect."

Wang and Benbasat [22] experimented using different types of explanations to test their effects on trust in the context of a recommendation agent.

The study confirms that explanations improve the initial trust of consumers.

Pu and Chen [23] investigated the different forms of presenting explanations in a recommendation agent. In the experiment, users were asked to evaluate the effect of explanations in two different forms. One of the forms gave separate explanations for each item in the product list. The other grouped similar items and explained the groups of products. The experiment shows that explanations for product recommendations generally enhanced the users' trust. Moreover, grouping explanations could reduce users' effort of perceiving recommendations in comparison to separate explanations. After grouping the products, the participants intended more to return to the agent for detailed information.

Pieters [24] discussed the relations among explanations, transparency, and trust in the view of information security. The author divided explanations into *explanation-for-trust* and *explanation-for-confidence*. Too high or too low level of details could lose the users' trust and failed to explain in the context of security-sensitive applications and expert systems integrated with AI.

Another example regarding the Clinical Decision Support System (CDSS) is from Bussone et al. [25]. The provided explanations in CDSS helps clinicians to assess the system's suggestions better and have more trust in the system.

2.2.2 Transparency and Trust as NFRs

Leite and Capelli [10] treated transparency as a non-functional requirement and discussed its definition. They explored the background of research on transparency and analyzed it as a quality issue in software engineering. Based on the Softgoal Interdependence Graph (SIG) [26], they considered transparency as a soft goal and created the *Transparency SIG* to show the interdependence between transparency and other soft goals. The following soft goals formed the second level of decomposition: Accessibility, Usability, Informativeness, Understandability, and Auditability. Using the *Transparency SIG*, the influence of explanations on the software transparency can be indirectly analyzed with other related NFRs as an intermediary.

Zinovatna and Cysneiros [27] discussed how transparency and privacy impact each other. By linking Leite's Transparency SIG and other existing knowledge, they created new SIGs containing transparency, privacy and other soft goals to help develop a system meeting both requirements.

Cysneiros and Leite [28] recently published a conference paper, which focused on trust in Software Engineering. They introduced the concept called *Corporate Social Responsibility* (CSR), which is used to enhance customers' trust by creating a trust paradigm for corporate. By adopting CSR into the software development processes, it is possible to deliver trustworthy software using the existing NFR framework. Therefore, they considered trust as an

NFR along with ethics and transparency, and connected them with the NFRs that they may interact. Their paper presented a simple SIG related to trust and other soft goals.

In fact, trust has raised researchers' concerns for decades. Hoffman et al. [29] purposed ann expanded a trust model in 2006, which has already included soft goals like usability, privacy, and security. Based on this model, Pavlidis [30] provided a methodology that the developers can follow to build a trustworthy information system and analyzed the relations further between trust and other properties of the model.

In 2001, Yu and Liu [31] suggested to consider trust as an NFR and used the NFR framework notation to study its effect on other soft goals. Besides that, they demonstrated a description framework to model intentional relationships among strategic actors with examples in the context of a bank card system.

Chapter 3

Research Goal and Design

In this chapter, the research goal and questions are presented and explained. The app was designed with concepts and features to support the study on the research questions. The table and figures of the design illustrate the app's functionalities. On top of that, some frequently used terms are defined to ease the understanding in Chapter 4. Combining the table and figures with the definitions together, the essential functions in this app can be understood. Within the definitions of terms, it is explained why these features or factors are treated as an explanation according to the definition of explanation in Chapter 1. At the end of this chapter, the overview of the usability test design is introduced with its settings and process.

3.1 Goal Definition

The goal definition is formulated with the template structured by Wohlin, et al. [32].

> We **analyze** perspectives of end-users about the demand for explanations in a navigation app **for the purpose of** investigation the impact of explanations **with respect to** non-functional requirements and their relation to transparency **from the point of view** of end-users **in the context of** synchronous remote usability test under the think-aloud method for three different design versions.

3.2 Research Questions

To realize the research goal, the Goal-Question-Metric Paradigm is used to derive correspondent research questions. Each research question focuses on one aspect of explanation's effect in this study. The metrics are represented in Figure 3.1 as child nodes of the research questions.

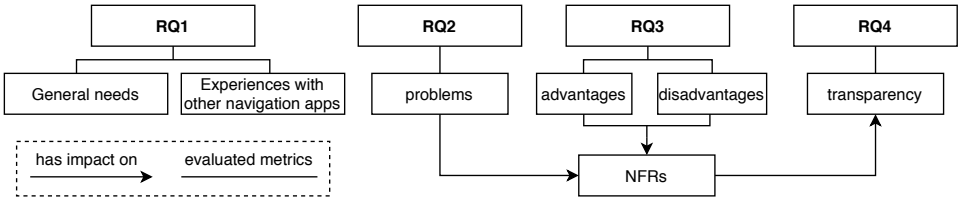


Figure 3.1: Research questions and related metrics

RQ1 Do end-users expect explanations in a navigation app?

The navigation app is one of the daily used software systems. It usually provides location information and route recommendations. While using the navigation app on the market, there could be problems related to understanding and use. Under the circumstances, explanations could be asked for by end-users. Therefore, it is necessary to understand users' general needs for explanations and expectations for such system. One question related to users' prior experiences is asked before they test the app. The participants are first requested to recall the situation of using a navigation app, then name the problems they confronted. This question was open and purely based on the thinking model of the test participants since they are formulated before the participants interact with the test app to avoid any potential influence. The goal was to analyze whether the problems were related to other non-functional requirements or the lack of essential explanations.

RQ2 How necessary is it to provide explanations to the end-user?

This research question focuses on the users' needs for explanation in the context of the test app. The participants are asked to test the first version, which does not provide any explanation. If the participants have problems while using the app, explanations are needed. Users could also ask explicitly for explanations to help them understand and use the app.

Furthermore, it is important to investigate, whether end-users understand the app correctly without explanations. Due to cognitive bias [8], end-users may not ask for explanations if the content of the app does not look suspicious, even in cases where they do not understand the system.

RQ3 What is the appropriate level of the explanation granularity, and in what form should they be provided?

The thinking model of users is normally different from developers. Although a professional software designer should avoid misunderstandings with the help of the experience and active communication with end-users, gaps in understanding might still exist. Therefore, the participants are provided

with versions with brief and detailed explanations, in order to find out, which granularity level of explanation fits the users' thinking model more. Besides the amount of information, how the information is shown also affects the perception of users, e.g., showing explanations by request or automatically, represented by graphic design (such as symbols or colors). To understand how these characteristics impact on users' perception, participants were asked about the advantages and disadvantages of each type of explanation. Their responses were analyzed in order to assess a possible impact on NFRs, similar to the method applied in the work of Chazette and Schneider [9].

RQ4 How does the explanation impact the transparency of the system?

Software transparency can be affected by different NFRs. To analyze the impact of explanations on transparency, we can work on the impact of explainability on the other NFRs first. Then by using the Transparency SIG, we can indirectly analyze the effect of explanations on transparency. As shown in Figure 3.2, the relation between explanations and NFRs is internally, It could be observed by conducting experiments and collecting users' feedback externally, following the UCD principle. The impact of explainability on transparency can be derived from the results of RQ2 and RQ3 using Transparency SIG as shown in Figure 3.1.

3.3 App Design

3.3.1 Design concept

In Section 2.1, it was pointed out that we need to take the difference between users' and developers' mental models into consideration, so that the information the app provided is perceived as an explanation rather than extra information by users. Taking this navigation app as an example, there is a weather panel indicating the current weather. When it is shown on the home screen, it cannot be considered as an explanation but only an extra information. However, if the weather is presented in the route screen, then they could be potentially needed by the end-users to help understand the travel option recommendation, and thus be considered as an explanation for the decision. Because users may subconsciously connect weather with the decision process of travel option. The connection between information and recommendation (or generally explanandum) is important to differentiate extra information and explanation.

In this app, there are two types of explanations: recommendation and guide. The questions regarding recommendation are similar to "why does the app suggest me to do this?" or "how does the app come up with this decision?" Guide is related to the questions asking for the guidance of use.

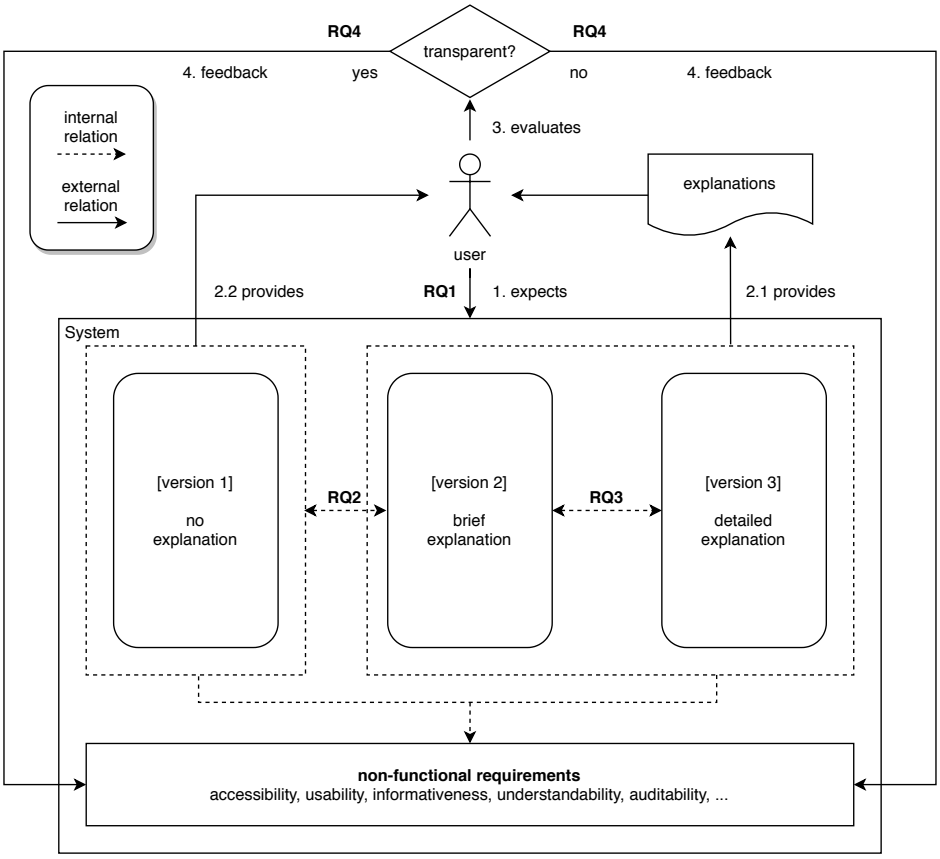


Figure 3.2: The internal and external relations

Users may ask questions similar to "how to use the app" or "what does this icon, button, text, etc. mean?"

As described in Figure 3.2, users usually expect the system before using the app (1), which is related to 3.2. The navigation app provides users with three different levels of explanations (2.1/2.2). Users evaluate the use of the app (3) and give feedback on the system (4), which externally reflects the internal relations between explanations and NFRs. Comparing the results of v1 and v2, 3.2 can be answered. Comparing the results of v2 and v3, 3.2 can be derived from the users' feedback. Regarding 3.2, it could be difficult for users to understand the concept of software transparency. Thus, 3.2 is indirectly derived from users' feedback related to the other NFRs.

To support the experiment, features are implemented in different versions and can be switched over by tapping the version buttons. Table 3.1 presents all features that might be tested during the experiment.

Table 3.1: The design concept for three versions

type	**feature**	**factor**	**design versions**		
			v1 (no)	v2 (brief)	v3 (detailed)
recommendation	route	construction	-	1. icon 2. tooltips containing details	same as v2
		accident	-	1. icon 2. tooltips containing details	
		traffic load	-	polyline segment (colors: blue, orange, red)	
	travel option	preference*	-	text: "Your preferred travel option" (if preference is set)	1. same as v2 2. weight indicator (↑/↓: positive/negative)
		weather	-	weather icon and temperature	
		distance	always shown	always shown	
guide	tips	/	-	1. descriptions (tip button, rough duration, precise duration) 2. controlled by tip button	1. descriptions (tip button, rough duration, precise duration, weight indicator) 2. first time uncontrolled
	tags	/	-	-	tags (best travel option, fastest route, construction, accident,)
	star*	/	-	recommended travel option marked with a star	-

*: implemented, but not relevant to the result

"preference" and "star" in Table 3.1 were implemented but did not receive enough valid feedback. Therefore, they are not mentioned in Chapter 4. All features in type "recommendation" are explainable.

feature: route

Route refers to the route recommendation, e.g., options of routes from A to B. v1 provides no traffic information. v2 and v3 show the icons for road construction and accident, and mark the congestion road in yellow or red.

Users may ask for the traffic information to help understand the explanandum road recommendation. Therefore, **traffic information** is considered as an **explanation**.

feature: travel option

Travel option refers to driving, walking or cycling. There are three factors for this feature, which are *preference*, *weather* and *distance*.

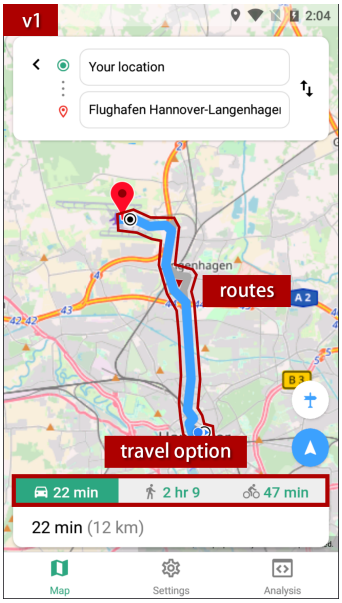


Figure 3.3: The interface of v1 (no explanation)

factor: weather

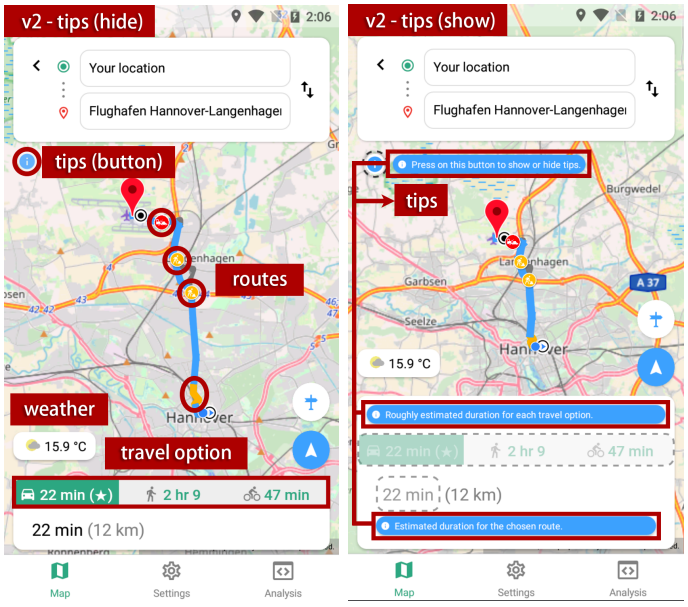
Weather is used to recommend the travel option. For example, if it is raining heavily, then driving is weighted more than walking and cycling. v1 provides no weather information. v2 provides weather icon with temperature for the current location.

Users may ask for the factors including weather information to help them understand the explanandum travel option recommendation. Therefore, **weather information** is considered as an **explanation**.

feature: tips

Tips refer to the hints shown with the info icons, which describes meanings or use of the interface. v1 provides no tips. v2 provides tips controlled by the tip button. v3 provides tips and shows them directly for the first time of use (uncontrolled for the first time).

Users may ask for tips to help understand the difference between explanandum duration behind the travel option (roughly estimated time) and the explanandum duration below (time specifically for the selected route). For v3, they may also ask for tips to help understand the explanandum weight indicators. Therefore, **tips** are considered as **explanations**.



(a) tips (hide) (b) tips (show)

Figure 3.4: The interface of v2 (brief)

feature: tags

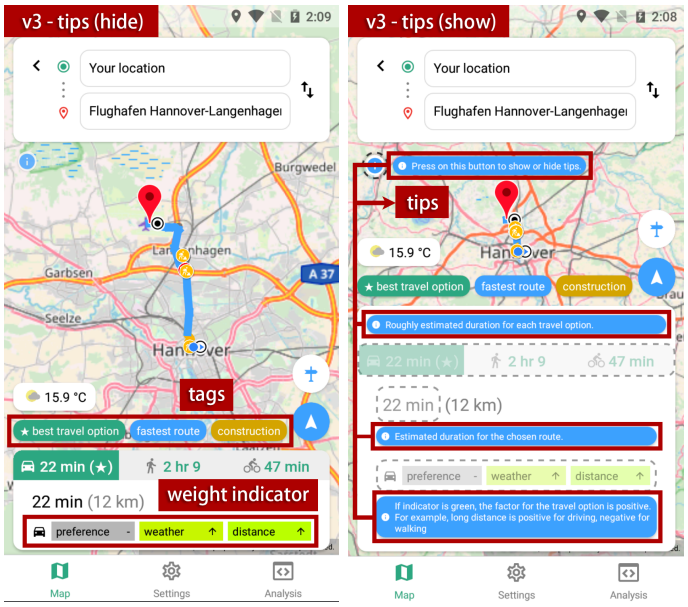
Tags refer to "best travel option", "construction", "accident", and "fastest route" badges. v1 and v2 provide no tags. v3 provides all four tags.

Users may ask for the meaning of the explanandum star, and icons on the routes. Therefore, **tags** are considered as **explanations**.

feature: travel option - weight indicators

Weight indicators refer to the three fields containing "preference", "weather", and "distance". The arrows and color scale as background indicate them as positive or negative factors. v1 and v2 provide no weight indicators. v3 provides the three indicators.

Users may ask for the detailed mechanism, of which factors and how these factors are calculated, to help them understand the explanandum travel option recommendation. Therefore, **weight indicators** are considered as **explanations**.



(a) tips (hide) (b) tips (show)

Figure 3.5: The interface of v3 (brief)

3.4 Usability Test

In this study, a synchronous remote usability test (sRUT) was conducted. In comparison to the conventional laboratory test, sRUT means that evaluators and participants are geographically separated and connected via online conference software.

3.4.1 Setup

The requirements of the usability test are

- **Android Emulator**

  - install test app “BtMap”
  - acquire test key (provided before testing)

- **Computer**
  - operating system: Windows
  - stable and fast internet connection
  - install “BlueStacks” (Android emulator)
  - install “Webex Meetings”

Test Key is used to restrict the use of the app only to test participants temporarily, since the app is only designed for research use. Test key is unique for each participant. It is provided before the usability test and suspended after the test.

Webex Meetings is an online conference software from Cisco, which mainly supports audio, video, screen sharing, and recording functions. For more details see `https://www.webex.com/`.

3.4.2 Test Design

The think-aloud method was used during the test, which requires participants to say their thoughts out loudly. The test consisted in five phases: 1) In the first phase, participants were asked about their experience with map applications and questions related to RQ1. 2) On the second phase, participants were requested to finish two tasks finding routes to some location, then questions for encountered problems were raised related to RQ2. 3)/4) In phases three and four, participants were asked to finish the tasks again with v2 and v3. Then they reported the advantages and disadvantages caused by different granularity and present forms of explanations. 5) In phase five, they chose a favorite version and expressed their feelings if their favorite version was not provided by the software company but one of the other two versions. Then, participants were allowed to test the app freely, proposed their suggestions, and asked for any questions.

The total duration of the test was about one hour. The detailed script of the remote usability test can be found in Appendix A.

Chapter 4

Results

The experiments have been conducted with 20 participants. All of them are students studying in German universities and born in the 1990s. During the interview, both open and closed questions are asked. The scale for the closed questions is unified **from 0 to 7**, in which zero stands for the least/worst and seven for the most/best. On average, the participants evaluated their knowledge of similar navigation apps as good (M: 5.15, SD: 1.36). The average frequency of using a navigation app according to their self-assessments is about three to four days (M: 3.7, SD: 2.15) a week. However, most of them said that they may use the navigation app much more frequently while traveling.

Table 4.1 defines the code classes used in figures for the code statistics as legends. The code classes related to Transparency SIG are defined by Leite and Cappelli [10].

4.1 Results for RQ1

RQ1 Do end-users expect explanations in a navigation app?

The question related to the experiences of the participants with other typical navigation app is asked at the beginning to help understand their general needs. They were required to describe the situation they had met, in which they could not understand the content or the functions of other navigation apps, and needed an explanation for their confusion.

Figure 4.1 is a stacked bar chart presenting the number of codes for the series *other_ apps.n*. The notation **.n** means the **negative codes** related to *other navigation apps*. The number on the bottom of each bar indicates the amount of the codes, e.g., "**5**" for the blue bar means **five** codes related to *user-friendliness*. In the legend, codes related to the four NFRs are numerated within the bracket, e.g., *user-friendliness* is marked with **(1)** and thus refers to the **first** NFR **Usability**.

Code Classes	Definitions
(1) Usability	**The quality of being able to provide good service**
simplicity	The quality of being free from difficulty or hardship or effort
intuitiveness	The quality of being spontaneously derived from or prompted by a natural tendency
user-friendliness	The ability to use easily
time consumption*	The time needed to finish task
attractiveness*	The ability to cause an interest, desire in, or gravitation to the system
(2) Informativeness	**The quality of providing or conveying information**
clarity	The ability to be free from obscurity and easy to understand
completeness	The quality of being complete and entire; having everything that is needed
correctness	The quality of being conform to fact or truth
comparable	The ability to be compared
consistency	The ability to express logical coherence and accordance with the facts
accuracy	The quality of being near to the true value
decision making*	The ability to help users make decision
helpfulness*	The quality of helping in a particular situation
(3) Understandability	**The quality of comprehensible language or thought**
conciseness	The ability to express a great deal in just a few words
(4) Auditability	**The ability to examine carefully for accuracy with the intent of verification**
controllability	The ability of being certain of something
UX*	**Person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service ****
impression*	The feeling that users get about the software system
Trust*	**The belief that something is true or correct or that users can rely on it**

*: not relevant to the Transparency SIG
**: definition from ISO 9241-210:2010(en)

Table 4.1: Definitions for code classes used in figures of code statistics

As shown in Figure 4.1, four of the codes refer to the usability problem, e.g., hard to find the settings or to switch the map view between 2D and 3D mode. 13 of the codes refer to the problem with **informativeness**, e.g., the destination of the train is not clearly informed, or the recommended route is not sensible. Most of the issues regarding these codes can be solved by providing appropriate explanations as guidance or reasons for the decision making. For instance, the app could have explained the meaning of the icons to the user when the user asked for the guidance of the train direction information or provided the user with the reason why choosing this route. One of the participants said, "The app hides some information from the user sometimes. ... always chooses the other bus stop rather than the one I prefer, which makes me very confused." The inclusion of such information on the app would make it more intuitive.

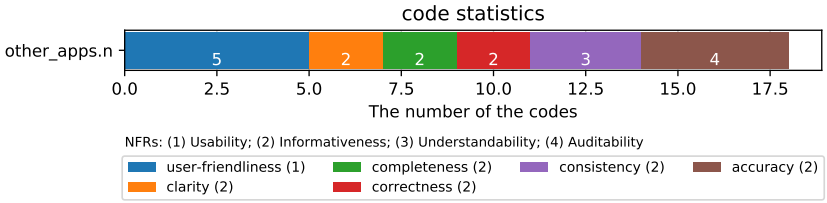


Figure 4.1: Problems encountered while using other navigation apps

4.2 Results for RQ2

RQ2 How necessary is it to provide explanations to the end-user?

To find out the need of explanation in the navigation app, the participants are asked to test the first version (v1: no explanation) and the second version (v2: brief explanation) in order.

4.2.1 The Test Result for v1

feature: route

After finishing all the tasks and get familiar with v1, they were asked to rate their desire to have the app explain the mechanism of the route recommendation. The average desire was rated as 4.10 (SD: 2.27), which may indicates the explanation, in this case, could be potentially helpful but not essential. The participants were then asked to point out the potential factors they think that could be considered by this app to recommend the route. Most of them said only that distance or duration are the possible factors considered, despite that the app also takes construction sites, accidents, and the traffic load into account. Overall, the users could only think of about half (52.63%) of all four factors. And yet, they felt that they had understood the mechanism, and therefore, do not necessarily need the explanation. This cognitive bias is caused by the differences between users' and developers' mental models.

feature: travel option

Same as the route, the users were asked to rate their desire for the reason of recommending the best travel option. The rating is, on average, 3.28 (SD: 1.94), which indicates the explanation might not be perceived as necessary. However, all the received answers contain only the factor distance or duration, despite that the user preference and the current weather also have weight considered in the computation. Giving *preference*, *weather*, and *distance/duration* each one point, the participants could only achieve one (33.33%) of three points on average. The participants overestimated their knowledge on this recommendation process generally.

first impression on v1

To understand the users' first impression on the first version without explanation, the participants were asked to describe their general feeling and view after using v1. The codes have been extracted from the answers and sorted into positive and negative aspects as shown in Figure 4.2. The post-fix p/n of the y-labels stands for positive and negative. Most of the

participants assessed the first version as positive leading to more positive codes. Nine of the participants commend the **simplicity**, as this version provides the essential functions needed from a navigation app and has the minimum description on its interface. Totally 17 codes are related to the positive evaluation on the **usability** and only three negative codes regarding the **informativeness** are mentioned, probably because of the lack of enough information.

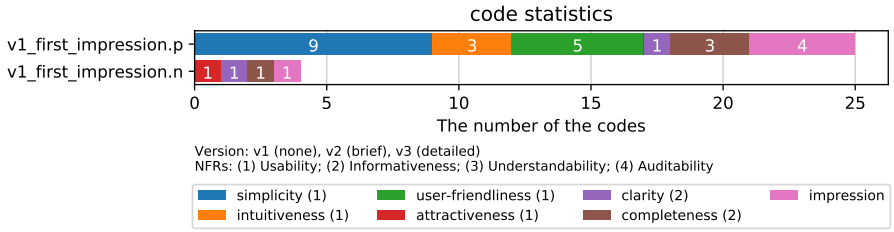


Figure 4.2: The code statistics for v1

At the end of Phase 2 (for v1), participants were asked directly to evaluate the following four aspects (using the same scale from 0 to 7):

"Is this app easy to understand?" related to **understandability** (M: 6.00, SD: 1.19). "Is this app easy to use?" related to **usability** (M: 5.90, SD: 1.12). "Do you trust the app overall?" related to **trust** (M: 5.53, SD: 1.06). "Do you feel that you are in control while using the app?" related to **controllability** (M: 6.35, SD: 0.86). The feedback are overall good according to these ratings.

4.2.2 The Test Result for v2

feature: route

In this version, route information is provided to end-users. After they have finished all tasks and get used to the new interface, they were asked to list the possible factors that they perceived as having an influence on the route recommendation. This time, 65.28% of all four factors could be correctly recognized by the participants, which is 12.65% higher than the percentage for v1. Since v2 provides traffic information like road construction, accident, and congestion, it is easier for the participants to consider them as factors of the recommendation algorithm. But the participants still did not perceive all factors completely. It was noticed that the users' need for explanation of such algorithm was low until they found something unusual. For example, a route with a longer distance was recommended due to the congestion on the other route. The recommendations were adopted if they did not violate the users' expectation, since users were not willing to make efforts to get through the details of the mechanism, not to mention asking for an explanation. The desire for the explanation for route recommendation was only rated on

average as 3.13 (SD: 1.95), slightly less than before. More than one-third of the participants had not fully understood this recommendation mechanism, but overall, they had less interest in it. One of the reasons could be that this mechanism is not, or at least has not been critical to them, because the recommendation matches their expectation. The other reason is that they believed they have fully understood it. Of the codes, as shown in Figure 4.3 15 refer to the positive influence on the **informativeness**, in which four refer to improvement on the **completeness** and six on the **decision making**. Overall, this feature achieved significant improvement with 19 positive codes over one negative code.

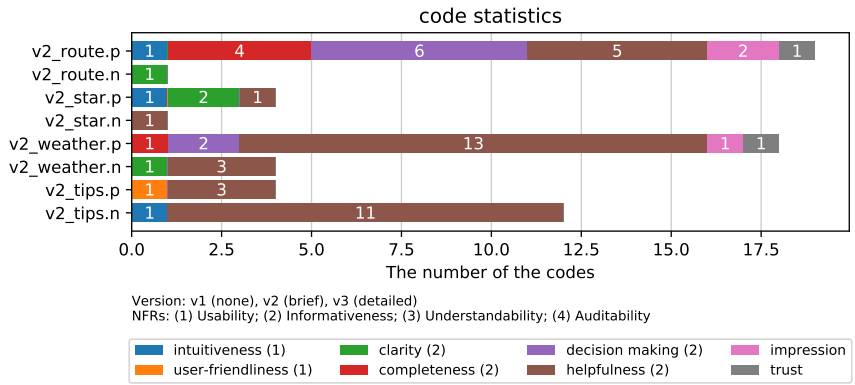


Figure 4.3: The code statistics for the features in v2.

factor: weather

All the participants expressed that the current weather condition influences their choice of the travel option. Nineteen of the codes are related to the **informativeness**, e.g., "The weather information is quite caring and helpful." 77.8% of the valid answers have affirmed that this information is necessary and useful. Regarding the negative codes of the **clarity**, some participants were not sure about the location used to present the weather. For example, if user plans to travel from city A to city B, it is not clear whether the weather shown is for A or for B. The possible improvement could be adding a short location description of the current weather. 16 of the codes are related to the positive effect on the **informativeness**, and only four are associated with the negative aspect. Overall, weather information can be considered as helpful and informative.

feature: travel option

During the test of v2, the participants were guided to set their preference of travel option, so that the app will consider their preferred travel option

as more significant. At this moment, all three factors including the current weather are presented directly and have been noticed by the participants. Then they were asked to rate their need for an explanation of the recommended travel option again. Different from the route recommendation, the participants tended to have more desire for an explanation with an average score of 4.30 (SD: 2.09) (comparing to 3.28 in v1). In v2, the participants were positively impressed by preference settings and weather. Positive impressions motivated users to have more interest in the recommendation mechanism. Thus, the rating is increased. Comparing to this feature, users' needs for the explanation of route recommendations were decreased. From the perspective of users, the perceived factors for the route recommendation were too typical to be attractive. Most of the participants intuitively thought that distance/duration was related to the decision process, which was only part of the factors used for the travel option recommendation algorithm. And distance/duration is quite typical to be used in such an algorithm. Thus, they could lose their interest in this algorithm.

feature: tips

During the test, 10 (50%) of the participants expressed explicitly that the tips are redundant. The other seven (35%) said that the tips are not necessary but can be included for someone who needs them. Only three (1.5%) participants valued this feature as useful. Some said that the content of the tips is redundant, but the form of guidance is suitable and intuitive to use. The codes present the fact that those tips are not helpful to the majority of the participants, as 11 refer to the negative influence on the **helpfulness**.

first impression on v2

Same as before, the participants were asked about their first impression on the second version with a brief explanation. Generally, this version has significantly more positive codes (23 times) than negative codes (6 times). Three refer to the positive impact on the **completeness**, e.g., "This version is complete comparing to the last version. With the weather information, this app looks more professional." Nine refer to the positive impact on **impression**, and two to positive effect on the **trust**.

Likewise, the participants have rated v2 regarding the **understandability** (M: 6.45 ↑, SD: 0.74), **usability** (M: 6.23 ↑, SD: 0.60), **trust** (M: 6.13 ↑, SD: 0.48), **controllability** (M: 6.25 ↓, SD: 0.84). The up and down arrows denote the improvement and impairment respectively in comparison to v1.

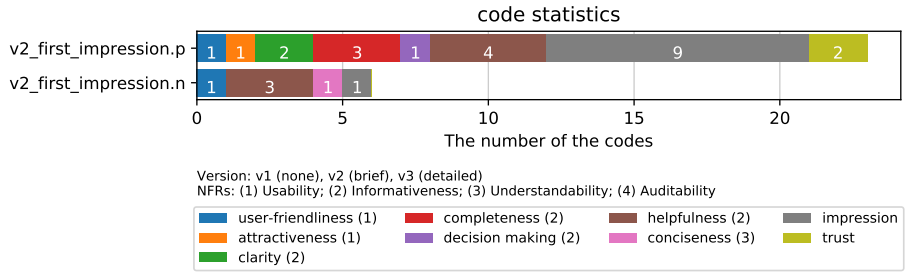


Figure 4.4: The code statistics for v2.

4.2.3 Analysis of the Results

In this subsection, the necessity of the explanation to the end-user is discussed. Both qualitative and quantitative analyses are provided. The qualitative analysis is performed with several statistical tests to determine the significance of the explanation impacts.

Statistical Tests

Statistical tests can be conducted with the **data sets** including two features (route, travel option) and four aspects (understandability, usability, trust, controllability) for both v1 and v2. The aim is to see if the changes between both versions have a significant influence on: 1. the users' desire for an explanation (if *route* and *travel option*); 2. the users' evaluation (if *understand*, *use*, *trust*, and *control*). The null hypothesis $H_0[data\_set]$ is defined generally as:

> $H_0[data\_set]$: There is no significant difference between v1 and v2 in view of the $[data\_set]$.

$data\_set \in \{route, travel\ option, understand, use, trust, control\}$

The normality of the data sets is checked with the *Kolmogorov-Smirnov* test. If both groups for v1 and v2 are normally distributed, both the *Wilcoxon signed-rank* test and *t*-test are performed. Otherwise, only the *Wilcoxon signed-rank* test can be performed. Both tests are suitable for the dependent sample with two groups. Table 4.2 shows the test results of the *Kolmogorov-Smirnov* tests and Table 4.3 shows the test results for the *Wilcoxon signed-rank* test and *t*-test at $\alpha = .05$.

According to the test results, the null hypothesizes of data sets *travel option* and *trust* can be rejected with the significance level of .05, i.e., the differences of the explanation between v1 and v2 do influence the users' desire for the explanation related to the *travel option*, and on the users' evaluation of trust. The other hypothesizes cannot be proven wrong.

Data Set	v1	v2
route	D = .17825, p = .49354. normally distributed	D = .13774, p = .7939. normally distributed
travel option	D = .13459, p = .81558. normally distributed	D = .19843, p = .36199. normally distributed
understand	D = .35, p = .01075. not normally distributed	D = .32664, p = .02117. not normally distributed
use	D = .18652, p = .43687. normally distributed	D = .25087, p = .13514. normally distributed
trust	D = .20521, p = .32317. normally distributed	D = .35464, p = .00934. not normally distributed
control	D = .33098, p = .01873. not normally distributed	D = .22937, p = .20849. normally distributed

Table 4.2: *Kolmogorov-Smirnov* test results for v1 and v2 at $\alpha = .05$

Data set	t-test	Wilcoxon signed-rank test
route	t = -1.583392, p = .12984. not significant	z = -1.2927, p = .19706. not significant
travel option	t = 4.350682, p = .00034. significant	z = -3.2958, p = .00096. significant
understand	-	z = -1.467, p = .14156. not significant
use	t = 1.508539, p = .14787. not significant	z = -1.5115, p = .13104. not significant
trust	-	z = -2.8114, p = .00496. significant
control	-	z = -0.3112, p = .75656. not significant
Notice: Some cells are empty because at least one of both versions are not normally distributed.		

Table 4.3: *t*-test and *Wilcoxon signed-rank* test results between v1 and v2 at $\alpha = .05$

Qualitative Analysis

There may be a reason why the differences in the evaluations were not significant enough. Some of the participants expressed that they have rated the first version too high after trying the second version, so that the improvement could not be described properly using the scores. It seems that the participants tended to take what was provided instead of asking more at first. Nevertheless, they may still demand other features or information

when they need them later or after comparing the app with other similar products. The indication of this inference is that no one chose v1 as their preferred version after they had used all three versions. Comparing the *v1_first_impression.n* (in Figure 4.2) with the *overall_if_v1.n* (in Figure 4.5), the participants have changed their minds criticizing v1 for the lack of **completeness** and lost their **trust** on the first version. More specifically, no one has voted for v1 as their most trusted version.

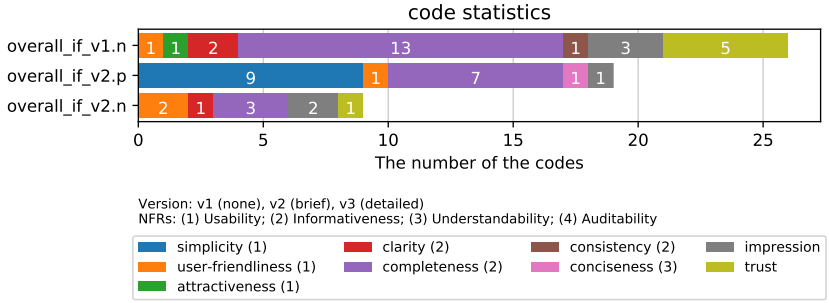


Figure 4.5: The overall impression on v1 and v2

To sum up, the first version was not perceived as informative and clear enough to use and to understand. Explanations should be carefully designed and provided with high quality.

4.3 Results for RQ3

RQ3 What is the appropriate level of the explanation granularity, and in what form should they be provided?

After testing the second version (v2: brief explanation), the participants are asked to check the third version (v3: detailed explanation). In this section, the test result of v3 is presented first. By analyzing the test result of v3 and comparing it to the v2, the answer to RQ3 can be derived.

4.3.1 The Test Result for v3

feature: tips

In the third version, tips are shown automatically when the app is used for the first time. Eight (40%) of the participants said that the uncontrolled form is good and helpful, e.g., "It makes me feel that the app is more reliable and professional. And the text can help me understand the app.", "It reminds me of some aspects that I might miss.", etc. Meanwhile, nine (45%) of the participants conveyed that they did not like this way of showing the information and was impressed negatively, e.g., "The uncontrolled tips

feel like ads," "I was shocked seeing so much text at the first moment. I just wanted to find a route. It is time-consuming to read them all." The remaining three (15%) participants remained neutral. Of the codes, eight refer to a good **impression**, while ten refer to a negative **impression**. The uncontrolled form is, therefore, controversial. The possible reasons for the negative impression could also be the value of the tips. If they are all helpful and does not take much area of the screen, the uncontrolled way could also be valued as meaningful. Therefore, the developers should present the text carefully with understandable, concise, and helpful content.

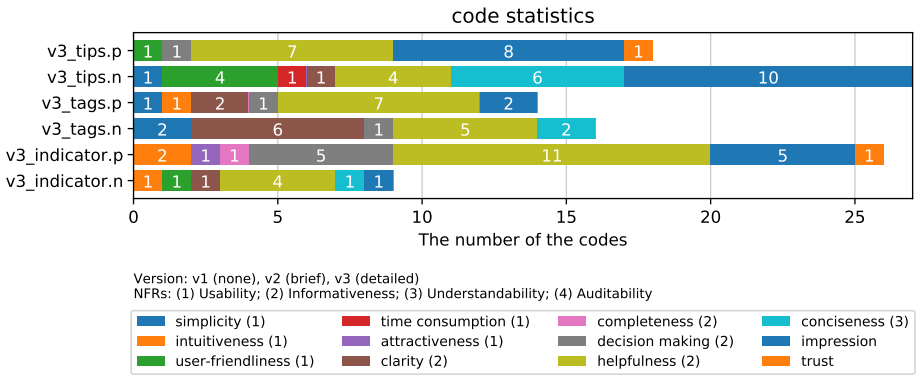


Figure 4.6: The code statistics for the features in v3

feature: tags

Tags, or badges, describe the icons like "star," "construction," and "accident," and colors of the route. After using v3, five (25%) of the participants valued this feature as positive, while six (30%) valued it as negative. The other nine remain neutral, either because the tags are unnecessary but not disturbing, or some of them are confusing. For the users, who can or at least think they can understand the meaning of the icons, they prefer not to have these tags, and indeed, icons should be intuitively understandable and clear. On the other hand, if some of the end-users are not familiar with the common metaphors or symbols, it could be problematic for them to understand, not to say, to use the app. Thus, it is necessary to consider the different needs of end-users and try to explain without disturbing the other users. As shown in Figure 4.6, the amount of positive codes for v3 is slightly less than negative codes. During the test, some users explicitly complained about one specific tag, because this tag caused confusion. Therefore, six of the codes refer to the negative impact on **clarity**.

feature: travel option - weight indicator

The weight indicator is aimed to provide the end-user with an explanation for the algorithm's internal reasoning process of the travel option recommendation. Nine of the participants said that this feature is helpful, e.g., "These three indicators helped me understand the reason for the app's selection of the current travel option.", "It helps the user to consider all the factors and then decide for themselves.", and so on. Only two expressed that they do not need this feature and are not curious about the reason for the recommendation. The other nine stayed neutral for different reasons. Some found that this feature is hard to understand but helpful. Some said that they do not need all three indicators. Others may prefer another form of showing this information. Derived from the codes, this feature has more positive effect, especially when it helps users make decision (**decision making**: 5 times). The four negative codes related to the **helpfulness** show that the content of the indicators could still be improved and simplified.

first impression on v3

Again, the participants were asked for their first impressions on the third version. Figure 4.7 shows that the numbers of positive and negative codes do not vary a lot. Three refer to the positive effect on the **completeness**, as this version has provided them with more useful information. Five of the codes refer to the positive **impression**. On the other hand, seven of the negative codes refer to the **helpfulness**, saying that, "This version has more detailed content. But it does not offer me more help.", "...I feel that the developers are trying to convey more of their ideas. But the experience of using the app is almost the same." Thus, apparent information could be redundant and hurt the **conciseness** of the interface. This inference is indicated by the five negative codes regarding conciseness.

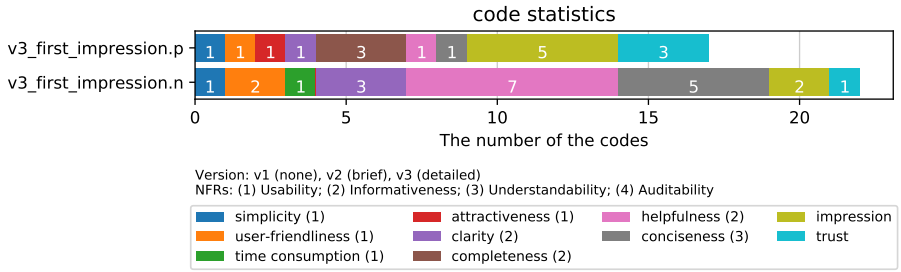


Figure 4.7: The code statistics for v3.

Likewise, the participants have rated v3 in view of the understandability (M: 5.93 ↓, SD: 1.21), usability (M: 6.05 ↓, SD: 0.81), trust (M: 6.53 ↑, SD: 0.57), controllability (M: 6.28 ↑, SD: 0.80). The arrows denote the trend of

the scores in comparison to v2.

4.3.2 Analysis of the results

In this section, v2 and v3 are compared to analyze the influence that the explanation granularity and the explanation form may have on the NFRs and UX.

Statistical Tests

Analog to subsection 4.2.3, the statistical tests are performed to evaluate the changes between v2 and v3, but only for the four aspects (understandability, usability, trust, controllability), since both features **route** and **travel option** are not changed in the third version. The statistical tests generally show the impact of the explanation changes on the users' evaluation. The null hypothesis $H_0[data\_set]$ is defined broadly as:

> $H_0[data\_set]$: There is no significant difference between v2 and v3 in view of the $[data\_set]$.

$data\_set \in \{understand, use, trust, control\}$

Equally, the normality of the data sets is checked with the *Kolmogorov-Smirnov* test. If both groups for v1 and v2 are normally distributed, both *Wilcoxon signed-rank* test and t-test are performed. Otherwise, only the *Wilcoxon signed-rank* test can be performed. Both tests are suitable for the dependent sample with two groups. Table 4.4 shows the test results of the *Kolmogorov-Smirnov* tests and Table 4.5 shows the test results for the *Wilcoxon signed-rank* test and t-test at $\alpha = .05$.

Data set	**v2**	**v3**
understand	D = .32664, p = .02117. not normally distributed	D = .22544, p = .2247. normally distributed
use	D = .25087, p= .3514. normally distributed	D = .27473, p = .07963. normally distributed
trust	D = .35464, p = .00934. not normally distributed	D = .30249, p = .04041. not normally distributed
control	D = .22937, p = .20849. normally distributed	D = .223, p = .23522. normally distributed

Table 4.4: *Kolmogorov-Smirnov* test results for v2 and v3 at $\alpha = .05$

The statistical tests reject the $H_0[trust]$, i.e., there is a significant difference between the evaluation of v2 and v3 in view of the trust, which

Data set	t-test	Wilcoxon signed-rank test
understand	-	z = -1.6449, p = .101. not significant
use	t = -0.760199, p = .45647. not significant	z = -0.0942, p = .92828. not significant
trust	-	z = -2.0616, p = .0394. significant
control	t = 0.148724, p = .88334. not significant	W = 17, W(critical) at N = 8 is 3. not significant
Notice: Some cells are empty because at least one of both versions are not normally distributed.		

Table 4.5: *t*-test and *Wilcoxon signed-rank* test results between v2 and v3 at $\alpha = .05$

may be caused by the changes on the granularity and form of the explanation. The other hypotheses cannot be rejected by the statistical tests, possibly due to the same learning effect described in Subsection 4.2.3.

Qualitative Analysis

Comparing the overall impression on v2 and v3, it is possible to see that the v3 has significantly more negative codes than v2. Six of the negative codes for v3 are related to the **user-friendliness**, three refer to the **helpfulness**, and four refer to the **conciseness**. Overall, the third version is more complex to use and to understand than the second version due to the additional explanations, which are not always demanded by users. Furthermore, more explanations need more room on the interface, and thus, affect the conciseness negatively. However, 15 (78.9%) of the participants explicitly expressed that explanation has a positive effect on their trust. In contrast, only one expressed that too much explanation will hurt trust. Overall, v3 had 15 votes for being the most trust version, while v2 has only eight votes (some of the participants have voted for both v2 and v3). The code statistics also conform to the changes in the average scores as describe before, as the score of the **understandability** and **usability** decreased slightly, in the meanwhile, the score of trust increased. After being asked about their favorite version, 12 (60%) of the participants voted for v2, and eight (40%) for v3.

In conclusion, most of the participants were willing to trade **trust** for **simplicity**, and **conciseness**, whereas some preferred thorough information were satisfied with the conciseness and simplicity. Although, it is hard to determine the exact granularity, and the feedback regarding the presenting form could also be affected by the quality of the explanation, considering this effect could help developers optimize the embedded explanations and maximize the overall positive UX. Moreover, the granularity and form

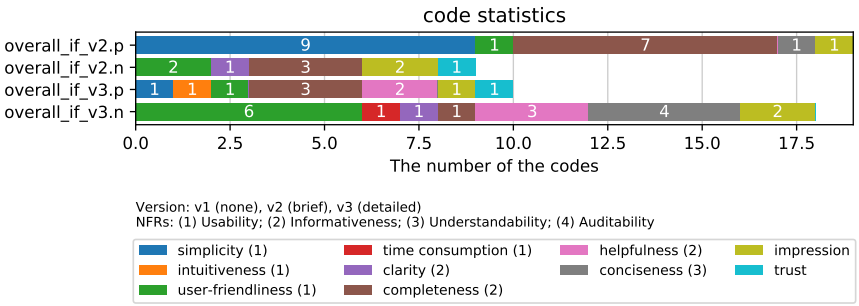


Figure 4.8: The overall impression on v2 and v3

depend on the users' needs for explanations. Discussion can be found later in Section 5.2.

4.4 Results for RQ4

RQ4 How does the explanation impact the transparency of the system?

The codes of the first and overall impression related to the NFRs in the Transparency SIG are divided into three soft goals **usability**, **informativeness**, **understandability**. Using the Transparency SIG, the relationships between **transparency** and these three soft goals can be qualitatively derived since all these soft goals help **transparency**. Figure 4.9 shows both the first and overall impressions of the three versions.

first impression A positive impact on usability could contribute to a positive influence on the perceived system transparency. As described before, v1 has the least explanation and thus the simplest and the most concise interface. According to the Transparency SIG, simplicity and conciseness are soft goals that help usability and understandability respectively and these other two soft goals (usability and understandability) may help software transparency. If the participants have not seen the next two versions, they might have mistaken v1 as transparent. This phenomenon leads to an interesting question of whether we should blind users from the real mechanisms to achieve improvement of other soft goals such as simplicity and conciseness, or tell them the whole truth to maximize transparency. This question is discussed later in Section 5.1. Different from v1, v2 has mainly positive codes that refer to **informativeness**, which helps improve its **transparency**. v2 contains more explanation than v1 with a low compensation of the other soft goals. By contrast, v3 has similar amount of positive codes but more negative codes on each soft goals. Therefore, more explanation does not always lead to higher transparency. It also depends on

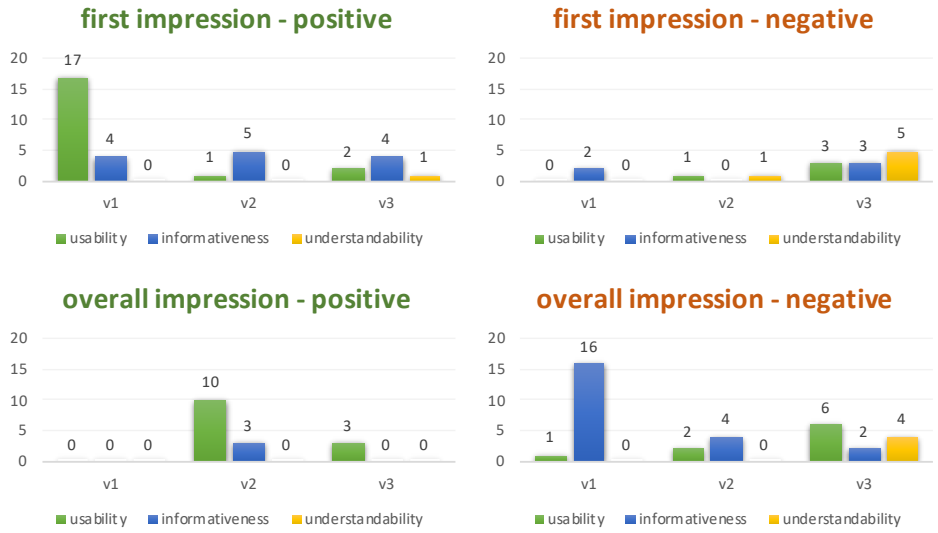


Figure 4.9: The first and overall impression chart for all three versions considering **usability**, **informativeness** and **understandability**

the granularity, the form, and the quality of the provided explanation.

overall impression Different from the first impression, the overall impression gave the participants chances to calibrate their evaluation considering all three versions. As mentioned before, no one has voted for the first version as their favorite version; the positive codes of v1 are hence left out. Observing the negative codes of v1, they are mostly aggregated into **informativeness**, and accordingly may hurt the **transparency** of the app. Same as the first impression, transparency of the app has not been changed as one of the intrinsic characteristics. Whereas the derived transparency from the users' feedback sharply decreased, along with the increase of the users' knowledge on the app. The number of positive codes of v2 related to **usability** increased significantly, possibly because the participants were familiar with the app. The positive codes of v2 regarding **informativeness** reduced slightly, perhaps affected by v3, since v3 covers more explanation than v2. Generally speaking, v2 should have good transparency, deduced indirectly from the users' feedback using the Transparency SIG. On the contrary, v3 has only few positive, but many negative codes, especially regarding **usability** and **understandability**. The conclusion is the same that **transparency** depends not only on the amount of the explanation but the granularity, the form, and the quality also matter.

Chapter 5

Discussion

5.1 The Need of Transparency and Explanation

In most cases, there is no need to maximize transparency, since the most thorough way to do so is to provide the end-users with the system's documents, including all technical details and design concepts. Non-professional users may neither understand nor need to know such information. Thus regularly, a system should not present all explanations to the users. On the other hand, if the system does not offer any explanation, some problems in use may appear quite quickly because it is difficult for a complex system to be intuitively and correctly understandable by most of the users. Given these two extreme cases, software developers need to find the balance between explainability and other NFRs, similar to the trade-off effect of transparency.

From the users' perspective, a system should be able to solve their problems efficiently and effectively. Users subconsciously have their expectations and understanding of a system before using it. If the system automatically delivers results that match their expectations, they may trust the system with a low transparency level equally or even more than the system with medium or high transparency levels. This phenomenon was discovered by Kizilcec [21] while experimenting with the effects of transparency on trust. The reason is that users may not evaluate the system thoroughly by examining the provided information. Likewise, the results of this thesis indicate that the first version of the app was trusted as transparent at first, described in Section 4.4. Moreover, if the users' first expectation is violated, more explanations and high transparency may even hurt the users' trust, because this information could be confusing and hardly understandable. Same as the third version in this study, the participants had not expected the navigation app to be so complicated, and thus their expectations are violated. In the meantime, the app provided more information, trying to explain the algorithms in more details. But it is time and effort consuming for the participants to process information and did not have much improvement in

the actual use. The app should thus not provide these types of explanations, at least not for all targeted groups, not at the beginning, and not compulsory to be read.

The need for transparency is also dependent on the type of system. If the system in case is a transparency sensitive system such as a bank system, data collection system, or autopilot system, then users may demand explicitly for high transparency to increase their trust or prevent misuse of their data. Whereas the system is not transparency sensitive, then the need for transparency is more dynamic, rising along with the growth of the users' knowledge. Considering the learning process, the app should allow users to learn the system step by step, and provide explanations gradually, instead of hoping the users understanding all mechanisms at once. For this reason, it is not recommended for the non-critical system to give too many explanations at first, but slowly during the use. Users can be motivated if they can use the app directly without much effort. After using the system for a while and getting familiar with the app, it could be easier for them to understand the explanations to meet their need for higher transparency.

It is recommended to follow the user-centered design principle to evaluate the users' need for transparency and explanation. By analyzing the users' feedback, the internal relations between design concepts and soft goals could be reflected, as shown in Figure 3.2.

5.2 The Granularity and Form of Explanations

During the experiment, some explanations were shown with different granularity and forms leading to different user feedback. For instance, 14 of the participants said that they needed an explanation for the travel option recommendation. However, three of them prefer not having all three indicators and thus evaluate this explanation as mediocre. Two of them expressed that they did not like the way it is presented, suggesting using a "rating scale with a star" (e.g., 2.0 $\star$) instead of "+/-" for positive or negative.

Kulesza et al. [33] used the terms **soundness** (nothing but the truth) and **completeness** (the whole truth) to study the effect of granularity in the music recommendation system. Please notice that the definition of completeness here describes the characteristic of explanations. It is different from the definition in Chapter 4. The conclusion in their paper is that the most sound and complete explanations help users the most build the mental models. Furthermore, it has the lowest perceived cost of learning and the highest perceived benefits. In the context of their study, the participants were asked whether they are willing to spend time attending seminars to learn the detailed mechanisms to help improve the program. Comparing to the learning cost of figuring out the mechanisms by themselves, the cost

of attending seminars is relatively low. However, in this thesis, it may not be realistic to require users to attend seminars and study algorithms. Therefore, the context of explanation in this thesis is more nearly daily use with a common navigation app. Despite the differences, these two terms are suitable to describe granularity.

The results of the experiment indicate that the complete explanation of the weight indicators is not always necessary. Most complete does not imply best UX, because a part of the explanation may be easily understood, and thus be considered as unnecessary or redundant. For the soundness, the app has always provided the participants with nothing but the truth, because gaining the trust of users through a certain degree of deception is a controversial ethical issue. However, it could be helpful and also moral, if a system abstracts or simplifies the explanations to reduce the users' effort of perceiving information. For instance, a system should usually avoid mathematical equations to explain an algorithm. A much better way could be using examples or metaphors in daily life to explain a complicated process.

The form of showing explanations could also be critical. Pu and Chen [23] discussed the effect of the explanation forms on UX when providing explanations of recommendations in an e-commerce system. One of the designs was to provide explanations separately for each product. The other clustered similar items under several categories and explained the categories instead. It reduces the burden of analysis for users and organizes the interface better. A well-designed interface can help users better understand the ideas easily without affecting the efficiency of the system.

Another example of the granularity and the form is from Wu et al. [34], which is also a part of the motivation for this thesis. They captured the start page of the private mode for most browsers on the market. All the browsers provided explanations for the private mode on this page. Then they surveyed 460 participants to examine whether they correctly understood the meaning of privacy mode. Sadly, many participants incorrectly believed that their data is a hundred percent secured and not even the ISP can track their access. Despite the explanations given by the browsers, users were still not able to understand the private mode at a satisfactory level. This problem is caused because of the misuse of words such as "privacy" and "safe," the wrong granularity of explanations due to the over-estimation of users' knowledge, and the inappropriate form of presenting texts. Those reasons make the explanation to be misleading, incomplete, and hardly understandable.

Last but not least, the quality and helpfulness of the explanation should always be guaranteed. It does not make sense to explain something apparent. Users are only willing to spend time and effort perceiving useful and understandable information, as indicated by the Attention Investment Model [35]. Therefore, it is crucial to determine the users' needs from different target groups. The granularity and the form only matter if this condition is true.

5.3 Summary

Table 5.1 concludes Section 5.1 and Section 5.2 regarding the possible consequences which might be caused by providing or not providing explanations. If the explanation is provided and users' understanding was wrong, then the users' expectation is violated. The table assumes that the provided explanations are always correct. Otherwise, users' expectations could also be violated wrongly, even if they have understood the system correctly. This table could be considered as an outline for integrating explanations in software system. In the future, experiment could be conducted to verify and complete the table.

explanation asked	explanation provided	understood correctly	consequences	
			expectation violated by explanation	evaluation
F	F	F	no	Good, if the correct understanding is not critical to the use. Bad, if users' misinterpretation causes problems.
F	F	T	no	Very good. No need to explain.
F	T	F	yes	Good, if the granularity level of the explanation is appropriate. Bad, if no, or too much explanation is provided.
F	T	T	no	Bad. The explanation is redundant.
T	F	F	no	Very bad. Users' need is not met, and misunderstanding could cause problems.
T	F	T	no	Bad. Users need is not met.
T	T	F	yes	Good, if the granularity level of the explanation is appropriate. Bad, if no, or too much explanation is provided.
T	T	T	no	Very good. It meets the users' need and expectation.

1. This table assumes that the correctness of explanation is guaranteed.
2. T and F stand for true and false.

Table 5.1: The possible consequences of providing or not providing explanations

Chapter 6

Limitations and Threats to Validity

The experiment was conducted with a small sample of 20 university students, which may only represent a small percentage of the global population. The need for an explanation could vary a lot if the user has no technical background and is not familiar with similar navigation apps. The ideas of the app design and usability test design came from the author. Both implementation of the app and the usability test was done by the author alone. Thus, it cannot be guaranteed that all subjective ideas were ruled out, although a pilot test was done before the experiment with the supervisor of this work. Also, the codes were analyzed both by the author and the supervisor to avoid subjective bias as much as possible. The Cohen's kappa coefficient (κ) is a statistic that is used to measure inter-rater reliability (and also Intra-rater reliability) for qualitative (categorical) items. [36] It indicates the possibility of agreements between items. Between the original codes from the author and the codes after the agreement, κ is equal to 0.69. The codes presented in this thesis are after the agreement, though.

The users' personal characteristic could also affect their needs for explanations. Millecamp et al. [37] pointed out that people with a low need for cognition benefit more from the explanations. But for people with a high need for cognition, explanations lower their confidence. The participants in this study are not classed with these criteria, and thus, the possible effect of this factor may create a bias on the result analysis.

Due to the COVID-19 pandemic, the experiment was performed via sRUT as described in Section 3.4. However, the quality of the test results is equivalent to the laboratory test, which has been proven by Bastien [38] and Andreasen et al. [39]. It could be better if the experiment were performed outside to simulate the real navigation scenario.

Another threat to validity could be the difference in definitions. During the tests, some users mentioned terms like simplicity and conciseness. Yet,

some of them could not tell their differences, which threats the code result. For instance, after receiving unexpected negative feedback on the usability, the facilitator asked this participant to explain the reason for giving this feedback. The participant expressed that in her mind, simplicity is a rather relative concept, which can be evaluated as the ratio of the total amount of useful information to the space occupied. Thus, even if the third version was more complicated, the participant perceived more helpful information than the second version, and hence evaluate v3 is over v2 as simple. It is hard to unify users' definition without influencing their feedback or giving implications. Therefore, only when the participant asked explicitly for the meaning or made obvious mistakes, the facilitator intervened.

The result for RQ4 is analyzed using the Transparency SIG. This method is indirectly and provides only qualitative instead of quantitative evaluation. Therefore, it is hard to tell which soft goals contribute more to transparency.

Another possible threat is the learning effect. During the experiment, the order of the test versions is from v1 to v3. The order might have a potential influence on the users' feedback. Rey et al. [40] studied the effect of the primacy information order on subjects' decision making. In their experiment, the change of the information order affected users' preference for purchase. Therefore, the test results could diverge if the order of the test versions changed.

Chapter 7

Conclusion

To analyze the impact of explanation on other NFRs and UX, a navigation app was designed providing three different granularity levels of explanations: **v1 (no explanation)**, **v2 (brief explanation)** and **v3 (detailed explanation)**. A synchronous remote usability test was conducted to test the users' needs on explanations, and the effect that the granularity and form of explanations may have. Using the Transparency SIG as a basis and other NFRs as an intermediary, the influence of explanations on software transparency was derived qualitatively. The feedback of the experiment was coded with different soft goals and aspects of UX. The test results were both quantitative with statistical tests and qualitative analyses. The findings are as follows:

RQ1 According to the evaluation of the navigation apps on the market by participants, these products have more or less functions that some users cannot understand. Regarding the use problems, apps can support users learning the functionalities by providing guidance and tips. In addition, the problems related to users' confusion on the recommendation can be solved by providing explanations about system decisions. Since these problems exist, explanations are generally expected in a navigation app by users to answer their questions.

RQ2 Comparing the test results for v1 and v2, all participants trusted v2 over v1 and complained about the lack of information and clarity in v1 after testing both versions. Moreover, no participant has chosen v1 as their favorite version. Thus, explanation are perceived as important by the participants. The learning effect has also been noticed since most of the participants did not complain about the first version when they have not tested the second version. However, this effect should not be ignored, as users can also learn from other competitors' products.

RQ3 Most participants stated explicitly that providing explanations will increase their trust in the system. In the same way, the number of participants who trusted v3 over v2 is almost twice as the participants who trusted v2 the most. Thus, increasing the granularity of explanations could have a positive effect on users' trust. However, the participants seemed willing to trade trust for better simplicity and conciseness, as more participants chose v2 as their favorite version overall. Trust is perhaps not the first priority for users in a navigation app. It is usability that they may value more. The form of explanations should be well designed to lower the learning cost and improve usability.

RQ4 More explanations do not imply better transparency since the quality, form, and granularity of explanations also matter. If the explanation is not needed and the form is hard to understand, they can both impair the transparency. If the granularity including soundness and completeness does not meet the users' expectations, the explanation could also be negative to transparency. Besides that, increasing transparency with explanation does not always lead to positive feedback. It depends on whether users' expectations are violated. If they are violated, then providing more explanations could hurt trust. Thus explanation should be carefully provided to achieve the appropriate transparency level.

In summary, explanations usually are necessary for a complex system. Software engineers should consider the impact of explanations on NFRs and UX while developing different types of software. It is recommended to conduct experiments, to analyze the impact of explanations on software quality and how their form and granularity impact each kind of system. The gap in the mental models between users' and developers' mental models could be closed by analyzing users' feedback and improving the NFRs of a software system.

Appendix A

Script of the Remote Usability Test

Welcome to the BtMap (Bachelor thesis Map) Usability Test.

Interview language: English?

Please note that your voice and the screen of the emulator will be recorded. There are five phases in this test. During the test, you will be guided to use this app and answer questions.

The test uses the thinking aloud method, which means you should say your thoughts out loud while using the app. For example, if you are going to press a button, you should say this action out loud like "Now, I am going to press this button and . . . ", "I am not sure what will happen if I press this button," or "What should I do next?". Just remember to say whatever the thoughts in your mind out to let me know.

A.1 Phase 1 (Warm Up)

RQ1 Do end-users expect explanations in a navigation app?

Question 1 *1* Are you familiar with map applications, e.g. Google Maps, Apple Maps, etc? 0: not familiar at all 7: professional

Question 2 *2* How frequently do you use them? 0 – 7 day(s) a week

Question 3 *3* Have you ever had problems understanding the content or functions of other navigation apps, asked questions like what does this sign mean, or why does the app give me this recommendation, etc.? For example, you are curious about how the app recommends the top restaurants nearby, or the app suggests an unusual route instead of which you are familiar with.

Exploration Now, I will help you simply explore this app. As you can see, the current screen is called "Analysis" for test purposes. There are three versions of the design, you can switch them by tapping buttons in the middle of the screen. Each of these three versions stands for a level of explanation. Explanation means, for example, tips for the use of a specific function or reasons for a recommendation. They are typically answers to the things you don't understand and asked for.

At the bottom of the screen, there are other two tabs "Map" and "Settings". To use the map, you can tap the "Map" tab. On top of the map screen, there is a search bar allowing you to search for a location. Alternatively, you can tap the place on the map to show the detail.

A.2 Phase 2 (v1)

RQ2 How necessary is it to provide explanations to the end-user?

Task Now, assume you are a tourist here and not familiar with this area. Please choose different places nearby within a 1 km range of your current location and find routes to these places. Explore and interact with the screen first. Let me know if you are ready.

Task Search a city or region nearby, find an appropriate route to there. Explore and interact with the screen first. Let me know if you are ready.

Question 4 (in general) *4* Do you find anything that you don't understand or have any questions about any content on the screen? Any suggestions for improvement?

Question 5 (route recommendation) *5.1* For driving, if there is more than one route available, would you be interested in the reason that one of them is selected and recommended by default rather than the other routes? 0: not interested at all 7: extremely interested

5.2 And what is your assumption of the factors that may be taken into account for the route recommendation?

Question 6 (travel option) Did you notice that the recommended travel option is not always the same by default? For example, if you plan to travel to Bremen, the app suggests you drive by default. But if you want to go someplace nearby, let's say within a 1 km range, the app will suggest you go there on foot.

Do the recommended travel options conform to your anticipation? For example, the app suggests you drive/walk/ride a bike. Is this your preferred travel option?

6.1 If it is, are you interested in the mechanism, which the app uses to predict your anticipation? 6.2 And what is your presumption of the mechanism? 0: not interested at all 7: extremely interested

6.1 If it is not, do you need an explanation, why this app chooses this travel option for you rather than what you anticipate? 6.2 Reasons? 0: no need at all 7: extremely need

These are actually two different aspects. If the app's recommendation conforms to the anticipation, then the user may not even notice this feature, thus has no interest in the explanation.

Question 7 (explanation -> NFRs) *7.1* In this version, explanations are barely provided. What is your overall impression of this version?

7.2 Is this app easy to understand (0-7)

7.3 Is this app easy to use? (0-7)

7.4 Do you trust the app overall? (0-7)

7.5 Do you feel that you are in control while using the app? (0-7) ...

A.3 Phase 3 (v2)

RQ3 What is the appropriate level of the explanation granularity, and in what form should they be provided?

Now, phase 2 is finished. Please tap the "Analysis" tab on the bottom right. Please switch to the brief version of the explanation by tapping the "brief" button.

Task Same as before, choose some places nearby within a 1 km range and a nearby city. Find the route to those places. Explore and interact with the screen. Let me know if you become acquainted with the interface.

Question 8 (in general) *8* Have you noticed any differences between the last version and this version? And what are they?

Question 9 (route recommendation) Have you noticed that there are icons now above the routes for driving? (Do you know that they are tappable?)

Have you noticed that the routes for driving are sometimes filled with different colors? *9* Do you know what it means?

Question 10 (route recommendation) This question is exceptionally for RQ2.

You are using a brief version of the explanation now. *10.1* Again, for driving, if there is more than one route available, would you be interested in

the reason that one of them is selected and recommended by default rather than the other routes? 0: not interested at all 7: extremely interested

10.2 And what is your assumption of the factors that may be taken into account for the route recommendation?

Question 11 (travel option) Have you noticed that there is a star ($\star$) right after the travel time? *11.1* What does it means?

Have you noticed that the current weather is shown in this version? *11.2* Would the current weather affect your decision of travel option? *11.3* Why? *11.4* Do you think this feature is useful? *11.5* And why?

Task Which travel option do you normally prefer? Driving? Walking? Or cycling? Tap "Settings" on the bottom. Explore this screen for a while.

Tap "Preferences" and set the preferred travel option to your preference.

Now go back to the "map", choose some places again. Find the routes to these places. Let me know if you are ready to continue.

Question 12 (travel option – preference) Did you notice the text below the time and distance saying "Your preferred travel option."? [wait] What do you think this means? [wait] 12 Does this information affect your impression on this map?

Question 13 (travel option) *13* At this moment, how much are you curious about the mechanism for the recommendation? 0: not curious at all 7: extremely curious

Question 14 (guide) *14* Do you know that you can show the tips by tapping the information icon on the top left of the map? Tap it and read the tips thoroughly. Let me know if you have read them all.

Question 15 (guide) *15.1* Are these tips helpful or redundant? *15.2* And why?

Question 16 (explanation -> NFRs) *16.1* Again, what is your overall impression of this version of the explanation? *16.2/3/4/5* Please name at least two advantages and disadvantages of this version in comparison to the former version. *16.6* Understandable? *16.7* Easy to use? *16.8* Trust? *16.9* In control?

A.4 Phase 4 (v3)

Now, phase 3 is finished. Please go to "Analysis" and switch the version to detailed. Then go back to the "Map".

Task Same as before, find some places and the routes to them. Explore this new design for a moment. Let me know if you are ready.

Question 17 (guide) You must have noticed that this time, tips are shown automatically when you first time searching for a route. *17* Is it better to show tips in this way in comparison to the second brief version of the explanation? [wait] Do you think this function is just redundant or even annoying?

Question 18 (route recommendation & travel option) Have you noticed those tags between the weather and the bottom panel? *18* Are they being helpful and make this app better or on the opposite? And why?

Question 19 (travel option – weight panel) You must have noticed that there are now three additional indicators on the bottom. *19* Can you tell me your perception of these three indicators? What do the color and arrow mean? (If you don't understand them, tap tips button, read the tips, and try to explain them again.)

Question 20 (travel option – weight panel) *20* Do you agree or disagree that this weight panel helps you use this app? Have they answered your interest or are they just unnecessary and redundant?

Question 21 (explanation -> NFRs) *21.1* For this last version, what is your overall impression? *21.2/3/4/5* Please name at least two advantages and disadvantages of this version in comparison to the second brief version. *21.6* Understandable? *21.7* Easy to use? *21.8* Trust? *21.9* In control?

A.5 Phase 5 (Overall)

Now, you have finished the phase 4.

Question 22 (explanation -> NFRs) *22.1* Please tell me, which of these three versions do you like the most? *22.2* And why? *22.3/4/5* (if V1/2/3) What problems can you think of, if the app does not provide the explanation in the way that you anticipate?

Question 23 (explanation -> trust) *23.1* Which version do you trust the most? *23.2* Do the explanations affect your trust?

Question 24 (in general) At last, is there still something that confuses you or you are curious about?

The test reaches the end now. Thank you for taking part in this test.

List of Figures

List of Tables

Bibliography

[1] Shirley Gregor and Izak Benbasat. Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS quarterly*, pages 497–530, 1999.

[2] Mahsan Nourani, Donald R Honeycutt, Jeremy E Block, Chiradeep Roy, Tahrima Rahman, Eric D Ragan, and Vibhav Gogate. Investigating the importance of first impressions and explainable ai with interactive video analysis. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–8, 2020.

[3] Rahul Iyer, Yuezhang Li, Huao Li, Michael Lewis, Ramitha Sundar, and Katia Sycara. Transparency and explanation in deep reinforcement learning neural networks. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 144–150, 2018.

[4] Jessie YC Chen, Michael J Barnes, Julia L Wright, Kimberly Stowers, and Shan G Lakhmani. Situation awareness-based agent transparency for human-autonomy teaming effectiveness. In *Micro-and nanotechnology sensors, systems, and applications IX*, volume 10194, page 101941V. International Society for Optics and Photonics, 2017.

[5] Chun-Hua Tsai and Peter Brusilovsky. Explaining recommendations in an interactive hybrid social recommender. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 391–396, 2019.

[6] Consumer financial protection bureau, §1002.9(b)(2). `https://www.consumerfinance.gov/policy-compliance/rulemaking/regulations/1002/9/`. Accessed: 2020-07-06.

[7] Regulation (eu) 2016/679 of the european parliament and of the council 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). `https://eur-lex.europa.eu/eli/reg/2016/679/oj`. Accessed: 2020-07-06.

[8] Maximilian A Köhl, Kevin Baum, Markus Langer, Daniel Oster, Timo Speith, and Dimitri Bohlender. Explainability as a non-functional requirement. In *2019 IEEE 27th International Requirements Engineering Conference (RE)*, pages 363–368. IEEE, 2019.

[9] Larissa Chazette and Kurt Schneider. Explainability as a non-functional requirement: challenges and recommendations. *Requirements Engineering*, pages 1–22, 2020.

[10] Julio Cesar Sampaio do Prado Leite and Claudia Cappelli. Software transparency. *Business & Information Systems Engineering*, 2(3):127–139, Jun 2010.

[11] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[12] Richard Tomsett, Dave Braines, Dan Harborne, Alun Preece, and Supriyo Chakraborty. Interpretable to whom? a role-based model for analyzing interpretable machine learning systems. *arXiv preprint arXiv:1806.07552*, 2018.

[13] Donald A. Norman. *The Design of Everyday Things*. Basic Books, Inc., USA, 2002.

[14] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[15] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

[16] Jörg Hoffmann and Daniele Magazzeni. Explainable ai planning (xaip): Overview and the case of contrastive explanation. In *Reasoning Web. Explainable Artificial Intelligence*, pages 277–282. Springer, 2019.

[17] Frank Kaptein, Joost Broekens, Koen Hindriks, and Mark Neerincx. The role of emotion in self-explanations by cognitive agents. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 88–93. IEEE, 2017.

[18] Francisco J Chiyah Garcia, David A Robb, Xingkun Liu, Atanas Laskov, Pedro Patron, and Helen Hastie. Explain yourself: A natural language interface for scrutable autonomous robots. *arXiv preprint arXiv:1803.02088*, 2018.

[19] Sylvain Bromberger. *On what we know we don't know: Explanation, theory, linguistics, and how questions shape them.* University of Chicago Press, 1992.

[20] P. N. Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness.* Harvard University Press, USA, 1986.

[21] René F Kizilcec. How much information? effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2390–2395, 2016.

[22] Weiquan Wang and Izak Benbasat. Recommendation agents for electronic commerce: Effects of explanation facilities on trusting beliefs. *Journal of Management Information Systems*, 23(4):217–246, 2007.

[23] Pearl Pu and Li Chen. Trust-inspiring explanation interfaces for recommender systems. *Knowledge-Based Systems*, 20(6):542–556, 2007.

[24] Wolter Pieters. Explanation and trust: what to tell the user in security and ai? *Ethics and information technology*, 13(1):53–64, 2011.

[25] Adrian Bussone, Simone Stumpf, and Dympna O'Sullivan. The role of explanations on trust and reliance in clinical decision support systems. In *2015 International Conference on Healthcare Informatics*, pages 160–169. IEEE, 2015.

[26] Lawrence Chung and Julio Cesar Sampaio do Prado Leite. On non-functional requirements in software engineering. In *Conceptual modeling: Foundations and applications*, pages 363–379. Springer, 2009.

[27] Olena Zinovatna and Luiz Marcio Cysneiros. Reusing knowledge on delivering privacy and transparency together. In *2015 IEEE Fifth International Workshop on Requirements Patterns (RePa)*, pages 17–24. IEEE, 2015.

[28] Luiz Marcio Cysneiros and Julio Cesar Sampaio do Prado Leite. Non-functional requirements orienting the development of socially responsible software. In *Enterprise, Business-Process and Information Systems Modeling*, pages 335–342. Springer, 2020.

[29] Lance J Hoffman, Kim Lawson-Jenkins, and Jeremy Blum. Trust beyond security: an expanded trust model. *Communications of the ACM*, 49(7):94–101, 2006.

[30] Michalis Pavlidis. Designing for trust. In *CAiSE (Doctoral Consortium)*, pages 3–14, 2011.

[31] Eric Yu and Lin Liu. Modelling trust for system design using the i* strategic actors framework. In *Trust in Cyber-societies*, pages 175–194. Springer, 2001.

[32] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[33] Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. Too much, too little, or just right? ways explanations impact end users' mental models. In *2013 IEEE Symposium on Visual Languages and Human Centric Computing*, pages 3–10. IEEE, 2013.

[34] Yuxi Wu, Panya Gupta, Miranda Wei, Yasemin Acar, Sascha Fahl, and Blase Ur. Your secrets are safe: How browsers' explanations impact misconceptions about private browsing mode. In *Proceedings of the 2018 World Wide Web Conference*, pages 217–226, 2018.

[35] Alan F Blackwell. First steps in programming: A rationale for attention investment models. In *Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments*, pages 2–10. IEEE, 2002.

[36] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282, 2012.

[37] Martijn Millecamp, Nyi Nyi Htun, Cristina Conati, and Katrien Verbert. To explain or not to explain: The effects of personal characteristics when explaining music recommendations. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, pages 397–407, New York, NY, USA, 2019. Association for Computing Machinery.

[38] JM Christian Bastien. Usability testing: a review of some methodological and technical aspects of the method. *International journal of medical informatics*, 79(4):e18–e23, 2010.

[39] Morten Sieker Andreasen, Henrik Villemann Nielsen, Simon Ormholt Schrøder, and Jan Stage. What happened to remote usability testing? an empirical study of three methods. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1405–1414, 2007.

[40] Arnaud Rey, Kévin Le Goff, Marlène Abadie, and Pierre Courrieu. The primacy order effect in complex decision making. *Psychological Research*, pages 1–10, 2019.