FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Design und Implementierung einer zukunftsfreundlichen Benutzeroberfläche für Online-Wahlhilfen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Medieninformatik

eingereicht von

Alexandra Geier, BSc.
Matrikelnummer 1125752

an der Fakultät für Informatik
der Technischen Universität Wien
Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr. Peter Purgathofer

Wien, 10. Oktober 2017

Alexandra Geier

Peter Purgathofer

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Design and Implementation of a Future-Friendly User Interface for Voting Advice Applications

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Media Informatics

by

Alexandra Geier, BSc.
Registration Number 1125752

to the Faculty of Informatics

at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr. Peter Purgathofer

Vienna, 10th October, 2017

Alexandra Geier

Peter Purgathofer

Erklärung zur Verfassung der Arbeit

Alexandra Geier, BSc.
Storkgasse 6/11, 1050 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 10. Oktober 2017

Alexandra Geier

Danksagung

Ich möchte dem Team des Instituts für neue Kulturtechnologien danken, die ihr Vertrauen in uns gesetzt haben, für sie diese neue Applikation zu implementieren.

Darüberhinaus möchte ich meiner Familie danken, die mir das Studium überhaupt erst ermöglicht hat, und meinen Freunden, die mich während dem Studium und insbesondere beim Verfassen dieser Masterarbeit unterstützt haben. Ein besonderer Dank gilt Andreas Taranetz mit dessen Zusammenarbeit die wahlkabine.at Platform implementiert wurde.

Acknowledgements

I would like to thank the members of the Institute for New Culture Technologies for placing their trust in us by letting us implement the platform.

Above all, I want to express my gratitude to my family, especially my parents who made it possible for me to study and all my friends for supporting me during my studies and especially during writing this thesis. Special thanks to Andreas Taranetz for his cooperation in developing the new wahlkabine.at platform.

Kurzfassung

Online-Wahlhilfen erfüllen einen wichtigen bildungserzieherischen Zweck, da sie unentschlossene Wähler dazu animieren sich mit politischen Themen auseinanderzusetzen. Die österreichische Online-Wahlhilfe *wahlkabine.at* wurde im Zuge der Nationalratswahl 2002 erstmals eingesetzt. In diesen mittlerweile 15 Jahren hat die Applikation massiv unter Softwarealterung gelitten, was eine Neuimplementierung unumgänglich machte. Ein zukunftsweisender Designansatz musste gewählt werden, um die Software-Lebensdauer zu maximieren.

Diese Masterarbeit diskutiert Design und Implementierung einer zukunftsfreundlichen Benutzeroberfläche für Online-Wahlhilfen am Beispiel von *wahlkabine.at*. Um vorzeitiger Softwarealterung entgegenzuwirken wurde eine Liste an Kriterien für zukunftsweisende Applikationen definiert: Barrierefreiheit, Anpassungsfähigkeit, Auffindbarkeit, Wartbarkeit, Performanz, Nachhaltigkeit und Usability. Diese Prinzipien dienten als Entwicklungsgrundlage; besonders Barrierefreiheit war ein essenzieller Aspekt, da die Applikation für alle Wähler gleichermaßen verwendbar sein sollte. Existierende Online-Wahlhilfen wurden untersucht, um von unterschiedlichen Lösungsansätzen zu lernen und Best Practices zu identifizieren.

Die Applikation wird in dieser Arbeit durch den gesamten Entwicklungsprozess begleitet, von der Problemdefinition zum Plattformrelaunch, wobei Zukunftsfreundlichkeit in jedem Entwicklungsschritt ein Hauptaugenmerk war. Das Resultat ist eine adaptive Benutzeroberfläche, die für alle Wähler eine gleichermaßen großartige User Experience bietet, unabhängig von verwendeten Enduser-Geräten.

Empirische Anwendungstests wurden durchgeführt, um die Leistung der Applikation zu beurteilen. Darüber hinaus wurden formale Evaluierungen durchgeführt, um die Einhaltung der etablierten Prinzipien sicherzustellen. Diese durchgeführten Analysen sichern die Anforderung, dass die implementierte Wahlhilfe auch in den kommenden Jahren eingesetzt werden kann.

Abstract

Voting Advice Applications (VAAs) help voters to familiarize themselves with highly-debated political topics, and therefore serve an important educational purpose; Austria's VAA *wahlkabine.at* has been in use since fall 2002. During the last 15 years the application suffered from severe software aging. Consequently, a new platform needed to be developed using a future-friendly design approach.

This master thesis describes design and implementation of a forward-thinking user interface for VAAs. To avoid premature software aging, first a list of criteria of future-friendly software was defined: accessibility, adaptability, findability, maintainability, performance, sustainability and usability. These principles served as the basis for the development of the new platform; especially accessibility was an important aspect as the educational application should be usable for all voters. Existing VAAs were investigated to learn about different solutions and identify best practices. This thesis delineates the development process from problem definition to platform relaunch focusing on implementing future-friendliness in every step of the way. In the end, an adaptive user interface emerged providing a great user experience for all voters, regardless of the devices used to access the application.

Empirical user tests were conducted in order to evaluate the application's performance. Moreover, formal evaluations were carried out to ensure compliance with the established principles and support the idea that the platform and will withstand the test of time.

Contents

Kurzfassung **xi**

Abstract **xiii**

Contents **xv**

1 Introduction **1**
1.1 Overview . . . 1
1.2 Problem Statement . . . 2
1.3 Aim of the Work . . . 2
1.4 Methodological Approach . . . 3
1.5 Structure of the Thesis . . . 3

2 Motivation **5**
2.1 Importance of VAAs . . . 5
2.2 Context . . . 7
2.3 Future-Friendliness . . . 7

3 Voting Advice Platforms **9**
3.1 Overview . . . 9
3.2 Classification . . . 10
3.3 Existing Applications . . . 11
3.4 Conclusions . . . 19

4 Future-Friendly User Interfaces **21**
4.1 Overview . . . 21
4.2 Adaptability . . . 22
4.3 Accessibility . . . 24
4.4 Findability . . . 26
4.5 Maintainability . . . 27
4.6 Performance . . . 29
4.7 Sustainability . . . 30
4.8 Usability . . . 31

5 Relaunch Wahlkabine.at **35**
5.1 Overview . . . 35
5.2 Requirements . . . 36
5.3 Research . . . 37
5.4 Content Analysis . . . 38
5.5 Design . . . 39
5.6 Mockups . . . 44
5.7 Mockup Evaluation . . . 48
5.8 Implementation . . . 51
5.9 Group Usability Testing 1 . . . 53
5.10 Group Usability Testing 2 . . . 56
5.11 Final Result . . . 60

6 Evaluation **63**
6.1 Adaptability . . . 63
6.2 Accessibility . . . 65
6.3 Findability . . . 67
6.4 Performance . . . 68
6.5 Maintainability . . . 69
6.6 Usability . . . 70
6.7 Sustainability . . . 70
6.8 Conclusion . . . 70

7 Conclusion **73**
7.1 Lessons Learned . . . 73
7.2 Research Directions . . . 74

List of Figures **75**

List of Tables **77**

Appendix A **79**
List of Voting Advice Applications . . . 79

Appendix B **81**
Accessibility Report . . . 81

Acronyms **91**

Bibliography **93**

CHAPTER 1

Introduction

1.1 Overview

Elections are essential instruments of modern democracy [103]. Voters' choices seem overwhelming; they have many possibilities to obtain information on the nominated parties. For voters who do not want to read entire election programs, information can be obtained from traditional media like television or print. Although television seemed to be the major source of information in 2009, according to Schulz [114], in recent years the Internet is also an increasingly important source to gain further knowledge on parties' positions [6]. However, comparing different parties' opinions on certain issues can sometimes be cumbersome.

Voting Advice Applications (VAAs) are web applications that can overcome this problem by providing a set of questions about various topics. These questions are sent to and answered by the nominated parties, which usually provide additional information to explain their reasoning. Users can then answer these questions themselves and thereby compare their views with the parties' positions [7].

Austria's VAA *wahlkabine.at* has been in use since fall 2002 [20]. Since its establishment, more than 100 million questions have been answered [20] and thus helped voters to become familiar with important topics of the elections and the parties' positions.

As the platform is outdated in terms of technologies used and user interaction, this thesis discusses design and implementation of a user interface for a relaunch of Austria's VAA in cooperation with the Institute for New Culture Technologies/t0, which operates the platform.

The first part of this thesis deals with research on existing VAAs in other countries, as well as a literature review on future-friendly design, which is an essential characteristic of the new platform. The second part of this thesis describes the actual steps involved in the relaunch as well as the presentation of the final result.

The platform is developed in collaboration with Andreas Taranetz, who focuses his research on the identification of a future-friendly software stack, as well as performance optimizations for the new platform [122].

1.2 Problem Statement

wahlkabine.at has existed for more than 15 years, during which many things have changed in terms of technology. First of all, the technology used is outdated, as parts of the old development stack were not brought up to modern day standards.

Secondly, the user interface also does not meet current standards. New technological devices emerged for which the application was not optimized in the first place. Since the invention of the iPhone in 2007 [53], more and more people use smartphones to access web applications; *wahlkabine.at* was never optimized for mobile usage. Studies show that non-mobile optimized websites are very cumbersome to use, as they force users to scroll extensively, which makes navigation complicated [118]. In addition to mobile-optimization, accessibility for people with special needs is missing.

The changing context in which the application has been used has required several adjustments. In order to make it operable on smartphones, for example, the questionnaire was separated from the website in 2010. Changing demands led to a lot of patches within the past 15 years. Thus, the codebase has reached a stage where it is not easily maintainable anymore and lots of refactoring efforts would be necessary to continue maintaining the current system. Therefore, it is more cost effective to scrap the old platform and build a new application from scratch [23].

1.3 Aim of the Work

The aim of this thesis is to design and implement a future-friendly user interface for VAAs on the example of *wahlkabine.at*. It should be forward-thinking in two ways: firstly, it should be easily maintainable with a small budget for the non-tech-savvy staff; secondly, the user interaction design should work without problems for the next ten years by taking future developments into account. Obviously it is not possible to account for every scenario as Dawson [28] points out, because software aging will always be an issue [96]. However, an effort should be made to implement the application in a way that will work with new devices and platforms with the highest probability [28].

Therefore, criteria need to be defined that make an application future-friendly. Afterwards, the design for a VAA can be created, taking into account the previously-defined criteria. VAAs should be freely accessible to all people. Thus, it must be ensured that the application meets the web accessibility guidelines.

To summarize, the Austrian VAA should be completely redesigned in a future-friendly way and successfully relaunched before the next big Austrian general election, which takes place on the 15th of October, 2017.

1.4 Methodological Approach

1.4.1 Literature Research

As mentioned in the previous section, a literature review on future-friendly web development was conducted to consider important characteristics for the relaunch of the platform. Different common online databases were used to gather information, namely: ACM Digital Library, Google Scholar, Mendeley research library, IEEExplore as well as CatalogPlus, the online library of Vienna University of Technology.

1.4.2 User-Centered Design

Besides a literature review, User Centered Design (UCD) is the central methodological approach in this thesis. UCD aims to put the user in the center of focus when designing a platform, as proposed by Norman and Draper [93]. There are many different methods to empirically evaluate usability [90]. In this thesis we will focus on Group Usability Testing (GUT) as presented by Downey [36], which is predestinated to find usability problems of simple tasks. This involves several participants simultaneously, but individually, performing a predefined task. Meanwhile one or more testers are observing and interacting with the participants. As many users perform the evaluation at once, testers' time is minimized and usability issues can be identified fast. A focused discussion with the participants at the end of the process identifies future features to consider. GUT will be carried out at several stages of the projects to identify as many usability issues as early as possible.

1.5 Structure of the Thesis

The motivation behind VAAs and future-friendly software development are summarized in Chapter 2. In order to learn about best practices, research on VAAs was conducted and will be presented in Chapter 3. Chapter 4 discusses properties of future-friendly user interfaces, which will serve as background information for Chapter 5. Chapter 5 delineates the development process from problem definition to platform relaunch. Evaluation of the platform, against criteria defined in Chapter 4, will be conducted in Chapter 6. Finally, in Chapter 7 the results are summarized and additionally future research topics and pending issues are outlined.

CHAPTER 2

Motivation

After outlining the aim of this thesis in the previous section, this chapter gives an overview on the motivation of VAAs and further describes the history and context of the Austrian VAA *wahlkabine.at*. Moreover, it highlights reasons why software developers should program in a forward-thinking manner.

2.1 Importance of VAAs

VAAs are web applications that enable users to compare their political views with the positions of political parties. This is achieved by answering usually between 20 and 50 questions about heavily-debated political topics such as "Should homosexual couples be allowed to get married?" or "Should there be compulsory vaccination for children?". The users' answers are then compared to the responses given by the political parties. This results in a ranked list of parties in descending order of agreement [83] and thereby helps users with their decision-making abilities by making political parties more easily comparable [6]. Rosema et al. [107] state that 30 to 40% of the voters in countries like Switzerland and Finland consult a VAA before an election.

Although VAAs differ in some aspects as Garzia et al. [47] point out, they share the following common benefits:

2.1.1 Clarity of Political Programs

A major benefit of VAAs is, that they force political parties to actually commit to a clear opinion [7][84]. Because VAAs make use of closed questions, parties are restricted to yes or no answers, though often with the possibility to provide further clarification and reasoning. Politicians always seem to provide evasive answers to simple questions [57, p. 76], so this functionality can really support voters figuring out what politicians mean.

In 2017, when the Austrian People's Party decided to publish its election program only one month before the election, Austria's VAA made sure to provide their positions even earlier [31].

2.1.2 Source of Information

VAAs are a source of knowledge and provide a good overview of the political landscape [46], thereby making it easier for people to stay informed [46]. In the Finnish elections of 2007, the majority of VAA users stated that VAAs are the most relevant source of information, even more than television or print media [112].

VAAs usually cover a broad range of topics, some of which might not be included in election programs [84]. Editorial teams make an effort to collect, analyze and evaluate information [46], thereby saving voters a lot of time, because citizens do not need to carry out these steps themselves [84]. This is a huge benefit to voters in countries with a high degree of party fragmentation.

In addition to the previously mentioned questions, further information is often provided on these platforms, such as detailed statements on why a political party opted for a certain answer. This is often complimented by a glossary that explains uncommon political terms, such as "press council" or "civilian service", in an understandable way [47].

2.1.3 Objective Results

Political candidates' personalities have become an increasingly important factor when determining whom to vote for [86]. A more personalized campaign could be seen as a threat, as politicians could manipulate the public by faking certain desirable characteristics [46].

To separate parties' believes from those of their representatives, VAAs usually provide full anonymity, as questions are answered without insights about the parties' statements while filling out the questionnaire [84]. This anonymity leads to objective results, which often surprises users, as Enyedi's study on the Hungarian VAA points out [37].

2.1.4 Civic Education

VAAs help voters to distinguish between different political parties using the same information source [49] and therefore serve an educational purpose [7][125]. Andreadis et al. [7] show in their research that VAAs not only motivate voters to learn more about political parties, but also help citizens to learn about currently debated topics.

It is also an important tool for schools, as students learn how to differentiate parties through gamification [43]. Some of the VAAs even make an effort to provide course materials for teachers to use in class with their students.

2.1.5 Motivating Voters

VAAs motivate people to learn more about politics in general and the different parties' positions [7]. According to a 2005 study conducted by Marschall [80], half the respondents stated they are going to collect further political information for the upcoming elections after having consulted a VAA. Therefore, VAAs are an important mechanism to encourage voters to learn about politics. In addition, voters are not only encouraged to gather further information, but are also more likely to cast a ballot in elections [50][125].

2.2 Context

As elaborated on in the previous section, VAAs have a lot of benefits, which is why many VAAs in Europe and all over the world [20] are becoming increasingly popular in recent years [47]. Most of them have been around for decades, the earliest of which was the Dutch *StemWijzer*, which was introduced in 1989 [107].

wahlkabine.at was founded in Austria in 2002 and is privately operated by the Institute for New Culture Technologies/t0 in cooperation with the Association for Political Enlightenment (German: Gesellschaft für Politische Aufklärung) [20]. The questions are drawn up by an editorial team consisting of journalists from respected Austrian newspapers; a new team is formed for each election.

Although studies clearly show the usefulness of VAAs [6], there is little support from the government for the Austrian platform [84]. *wahlkabine.at* attaches great importance to data security and privacy, which is also why no personal data is stored, and results are not used or sold in any way [64]. In order to finance the project, the institute needs to apply for public grants before elections to be able to compensate journalists for their editorial work in drawing up the questions and to pay for hosting. Major online traffic usually occurs four to six weeks prior to general elections and increases in popularity as soon as the platform is presented by the media. Between elections web traffic is insignificant. The institute does not have the financial means to employ dedicated engineers and therefore relies entirely on volunteers.

Since its design and implementation in 2002, there has been only one major update of the platform in 2008, when a patch using JQuery Mobile was developed to make the platform usable on smartphones. The vast amount of patches resulted in highly unreadable code which, when combined with old technology and an outdated user interface, demands the implementation of a new platform built from scratch.

Even the Institute for New Culture Technologies stated it could not be reused and is not worth the effort to fix it [64]. The old platform is described in detail in Section 3.3.1.

2.3 Future-Friendliness

As mentioned in the previous section, there are only very limited funds to maintain *wahlkabine.at*. Furthermore, the new implementation must be easily maintainable by

people with limited technical knowledge. Taking these two considerations into account, the forward-thinking design approach of the new platform will mitigate the risk of premature software aging.

Nowadays, little effort is made to make applications or products future-friendly. Frequently, software development happens under tight constraints. Speed is often the top priority in software development, which leaves little time to consider software aging [96]. What is more, many software engineers may not have the right educational background to program in a sustainable way [96].

However, a trend towards future-friendliness has been emerging in many different areas around information technology and design. Research is being conducted on future-friendly algorithms [51], robotics [98] and information systems [10]. There is even research on how to make laws for big data protection as long-lasting as possible [87]. Kerr [71] has worked closely on future-proof product design. He points out reasons for aiming to develop sustainable products. In addition, he defines what future-proof means by examining everyday items such as coffee cups, tables and silverware [71]. Besides research on design of tangible products, Dawson [28] has worked on future-friendliness in the field of web design by elaborating on devices and browsers that are currently being used and ensuring that a website is compatible with them.

There is also a lot of research on software aging. Researchers agree on the fact that software is going through a similar aging process as are biological systems [96]. The difference, however, is that software aging can often be reverted by understanding the root cause [2]. Previous studies focus on measuring software aging [142], depicting influence factors for software aging [96], identifying types of age-related bugs [3], prolonging the software aging process [23] and rejuvenating software [26]. However, all these studies consider already finished applications that are in use rather than software that is yet to be developed.

There is little-to-no research on how to develop new software in a future-friendly way. In particular studies on future-friendly user interfaces in regards to VAAs are non-existent; this thesis aims to fill this gap.

After reading this section it should be evident why it is desirable to have a VAA as an information tool for elections and why there is the need to develop a new application from scratch and doing so in a future-friendly way. This knowledge will be useful to analyze other countries' VAAs in the following chapter in order to develop best practices.

CHAPTER 3

Voting Advice Platforms

After an elaboration on the desirability of future-friendly VAAs in Chapter 2, this chapter gives an overview on different user interfaces of VAAs across Europe in order to identify similarities and uncommon practices.

3.1 Overview

The Internet becomes more and more popular in everyday life [102]; it also found its way into politics [38], where it facilitates different political activities: parties provide information about their programs on their websites, donations can be made via online payment methods and political discussions are held on social media platforms. Some countries like Brazil [106] and Estonia [126], amongst others, even apply e-voting, hoping it will increase voter participation.

Before voters cast their ballot, they obviously need to decide which party represents their interests in the best way. In order to help them with their voting decisions, an increasing number of people use VAAs. They are a valuable source of information to facilitate their choice [6]. VAAs are spread all over the world; they are especially popular in Western European countries and North America. Figure 3.1 highlights countries with national or transnational VAAs, which were identified in the course of the VAA Global Census 2016 [48, p. 379]. Some of the countries were added on the map in the course of the research for this thesis.

In this section we survey state-of-the-art VAAs in order to highlight similarities and differences in their approaches, starting with a simple classification.

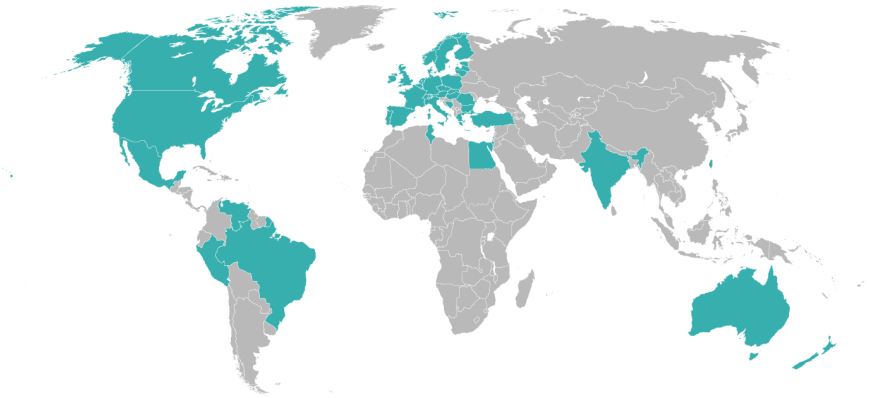


Figure 3.1: Countries with national or transnational VAAs.

3.2 Classification

Basically, VAAs can be categorized in three pillars depending on how users can interact with the provided information.

Platforms listing standpoints

This most basic type of platforms focuses on summarizing standpoints of politicians and parties without establishing relations to another. For instance, the German public-broadcasting institution Westdeutscher Rundfunk (WDR) recorded 771 nominated politicians answering currently discussed political questions for the federal elections in 2017 [133].

Platforms comparing politicians' standpoints

More advanced platforms take the process one step further. Instead of merely listing information they offer the opportunity to compare different standpoints and help users to relate information. *Pollenize* [8] is an example of such a platform comparing viewpoints of candidates of the United States Presidential Elections 2016. In 2017, VAAs also found their way into social media. Facebook introduced the possibility to add statements on political issues on political parties' pages [136]. As different parties comment on the same issues, it is possible to directly compare different parties' viewpoints.

Platforms comparing the user standpoint

Some applications go even further and do not only compare positions of different political parties, but also take users' viewpoints into account. Such applications are constructed as a questionnaire, which let users answer political statements in a quiz-like manner before they receive a ranked list of political parties or politicians [111] in descending order of

political agreement based on their answers. *wahlkabine.at* [64] is an example of such an application.

As this thesis is about the third category of applications, we will analyze several user interfaces falling into this category.

3.3 Existing Applications

This section will examine various VAAs across different countries, as well as the current *wahlkabine.at* platform in order to learn about best practices.

To gain a preferably diverse set of insights, applications from different countries were chosen. As there are countless applications with huge differences, we decided to evaluate the VAAs of countries where they are used most [19], as well as where VAAs are a well-established concept of democracy. Specifically, we chose to analyze the previous Austrian *wahlkabine.at* [64], the German *wahl-o-mat* [13] and the applications of Switzerland [41], the Netherlands [105] and Belgium [74], all of which are discussed in Garcia and Cedroni's report on European state-of-the-art VAAs [20]. Although Finland is a country where VAAs are very popular, its VAA was not further analyzed due to the language barrier. Instead, the Canadian application *Vote Compass* [128] was examined, as it appeared to have pursued a forward-thinking approach.

3.3.1 wahlkabine

The Austrian VAA *wahlkabine.at* [64] was first published in fall 2002 and is operated by the Institute for New Culture Technologies/t0 [20]. Questionnaires are published for federal and regional elections, as well as for European elections and elections of the Austrian Student Union (German: Österreichische Hochschülerschaft (ÖH)).

On the landing page there is a comprehensive overview of all the past questionnaires. However, only the questionnaires from 2010 onwards can be executed due to technical changes of the platform. Starting a questionnaire results in a separate window being opened which impedes accessibility [129]. The reason is that website and questionnaire application were separate into two applications to allow for mobile usage.

The actual questionnaire is in contrast to the rest of the website, optimized for mobile devices, which makes it a little cumbersome to operate on desktop computers. Buttons and display elements are rather small as can be seen in Figure 3.2. Users are shown approximately 25 statements and need to express whether they agree, disagree or abstain. Uncommon political terms, which are highlighted, are explained in a glossary. When an answer is given another input field fades in where users need to state on a scale from one to nine how important they consider the statement. Only after they have decided on their weight, a "continue" button is enabled which needs to be pressed to proceed to the next question. Overall, this means in total three clicks are needed to answer one

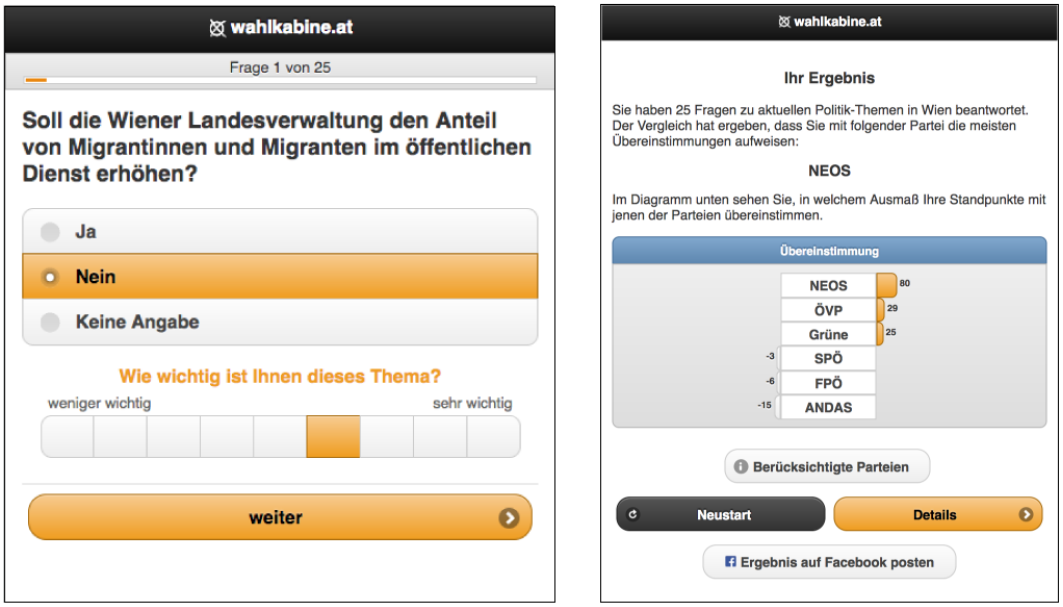


Figure 3.2: wahlkabine.at. Question and result screens of the old application.

question, which results in a total of 75 clicks for a 25-question-test, which indicates room for improvement.

In case users want to change their mind about a question, it is possible to go back questions one by one. However, it is not possible to jump forth over the already answered questions again, which means all jumped over questions need to be answered again. There are no keyboard shortcuts. After each question the questionnaire is completely rebuild, which results in loading time. Especially at locations with bad Internet connection, like in public transportation, this results in a considerate amount of waiting time.

After having answered the questions, a note explains that the result is not a voting recommendation. More influence factors need to be considered than a couple of questions could cover. After accepting this statement, the user can navigate to the result page.

The results are displayed as bar charts, as can be seen on the right screen of Figure 3.2. Values can be positive or negative absolute integers. To gain further information on the parties' positions the user can click through a list of statements showing the parties answers complemented by comments exemplifying their opinion. There is also the option to share the result on social media.

In summary, *wahlkabine.at* is definitely not state-of-the-art anymore. Although there are many promising concepts behind the platform, like the glossary, lacking responsiveness and accessibility make it cumbersome to use as of 2017.

3.3.2 wahl-o-mat

Germany's *wahl-o-mat* [13] was launched with the federal elections in 2002 and is managed by the German Federal Agency for Civic Education (FACE)[81].

Every questionnaire consists of 38 so-called assumptions. The user can decide whether he or she agrees, disagrees or is undecided. There is also the possibility to skip statements. The question screen on desktop devices can be seen in Figure 3.3. The application is fully responsive and also works well on mobile devices. It is very effortless to go through the questionnaire as only one click per question is needed, which keeps the process quick and easy. In case a user wants to change already given answers, it is possible to go back to any answered question, as on the bottom of the questionnaire a list of points, one for each question, shows the progress and facilitates navigation.

Social media links are not just embedded at the result screen, but available for every question. This means political issues can be shared on twitter or via mail during the process, which fosters discussion on these political topics. The page is rather slow, as it is completely reloaded after each question, which takes around three seconds for every answer depending on the connection speed.

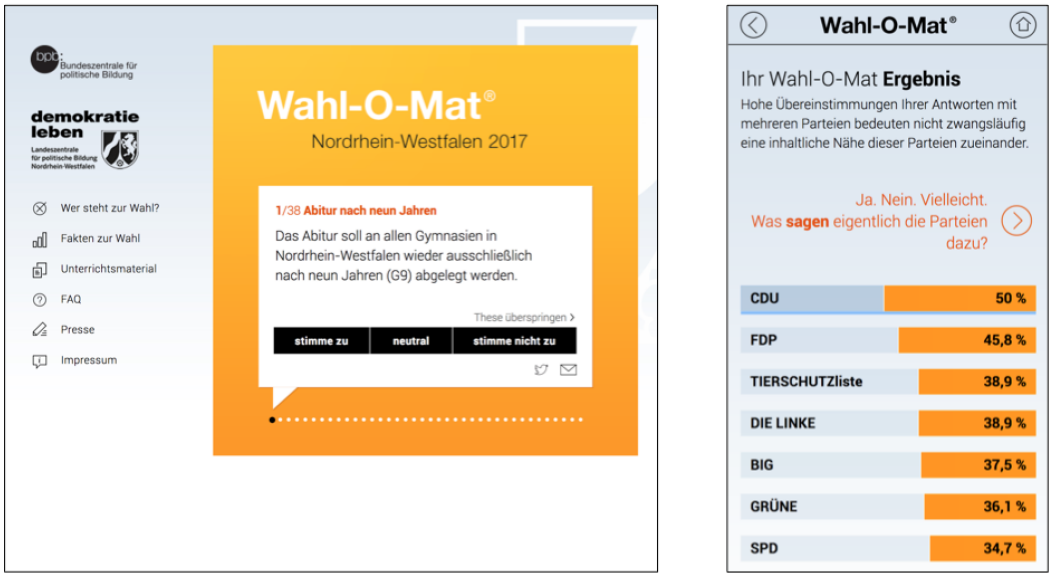


Figure 3.3: wahl-o-mat. Question and result screens.

After having answered all questions, users reach a summary screen where they have the opportunity to give some questions an additional weight by highlighting them as important. Such questions have twice the impact towards the result. This step is optional and can be skipped. As all the questions are listed on the same screen, it provides a good opportunity to compare questions and save time in case a user does not care about

differentiating questions in regards of weight.

As a last step, users need to decide with which parties they want to compare their positions. As there are at times more than thirty parties eligible for election [14], up to eight parties can be chosen for comparison. Thus, the result graphic is kept as concise as possible.

The results are shown as bar charts, where next to each party a bar is depicted, which reflects the degree of proximity in percent to the previously chosen parties, as can be seen in Figure 3.3. By clicking one of these bars, it is possible to obtain further information on the particular party. In order to understand the result better, it is possible to go through all the questions again and look at all the parties' answers and their explanatory statements and compare it to one's answer. It is also possible to download an overview of this as a PDF document.

wahl-o-mat is a great application providing a good user-experience. The application adapts to different screen sizes and avoids unnecessary clicks. The result is easily comprehendible and encourages users to be further engaged in politics. However, further efforts need to be made to support accessibility.

3.3.3 Smartvote

Introduced in 2003, *smartvote* [41] is the most popular VAA in Switzerland and follows, compared to the *wahl-o-mat*, a slightly different notion [75]. Unlike *wahlkabine.at* and *wahl-o-mat*, where parties are contacted by an editorial team to answer questions, politicians can register on the platform and enter their responses themselves [40]. Unlike other countries, where VAAs are mostly used for federal elections, *smartvote* is also applied at regional and municipal level. Switzerland has more than one official languages; thus, the test can be conducted in five different languages: German, English, French, Italian and Rhaeto-Romanic. Concerning responsiveness, the user interface is not responsive but offers a separate version for mobile devices.

Compared to other VAAs it has more than 70 questions which is more than twice as many as questionnaires of other VAAs [75] have. Thus, more topics can be covered in greater detail. Therefore, the outcome can be considered as more reliable [75]. However, in order to fasten the procedure there is also the possibility to conduct a rapid test with a subset of 30 questions.

The questions are grouped by categories like "school & education" or "finance & taxes". Users can decide whether they prefer answering question by question or category by category. There are four possible answers ranging from "yes" to "rather yes", "rather no" and "no". All of the questions are preanswered with "no answer". Users can skip as many questions as they want, which speeds the whole process up, but also reduces reliability of the result [75]. Additionally, an optional question weight can be defined which is a number between one and five and has a default medium value of 3.

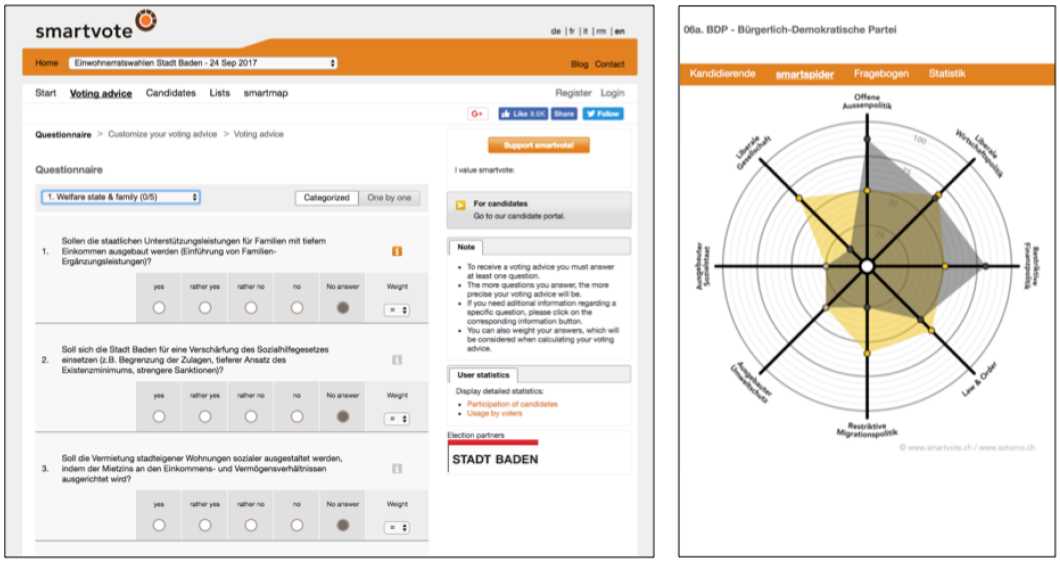


Figure 3.4: smartvote. Question and result screens.

Some of the statements provide additional information explaining the statement further and also listing its advantages and disadvantages. After answering the questions the user can decide whether he or she wants to be compared to a politician or a whole electoral list.

The result is visualized using bar charts with positive percentage values. What makes *smartvote* special, is, that the result can also be visualized on a political spectrum; the x-axis ranges from left to right and the y-axis ranges from conservative to liberal. Another visalization is a so-called smartspider, which is an eight-dimensional radar chart illustrated in Figure 3.4.

All in all, *smartvote* allows users many degrees of freedom, from choosing how many questions to answer and how many per screen, who they want their result to be compared to and even the method of visualization, which can be either a bar chart or a radar chart. However, accessibility for all users needs to be better supported by the application.

3.3.4 StemWijzer

The Dutch StemWijzer [105] is considered to be one of the oldest VAAs [29]. Initially it started in 1989 as a book to educate students. The first advanced digital version of the *StemWijzer* was published in 1994. Although the user rates were low at the beginning, they kept on producing new versions for the following elections [29].

The website is fully responsive for mobile and desktop devices. What is special is that *StemWijzer* provides a separate, visually simplified version of its questionnaire for people

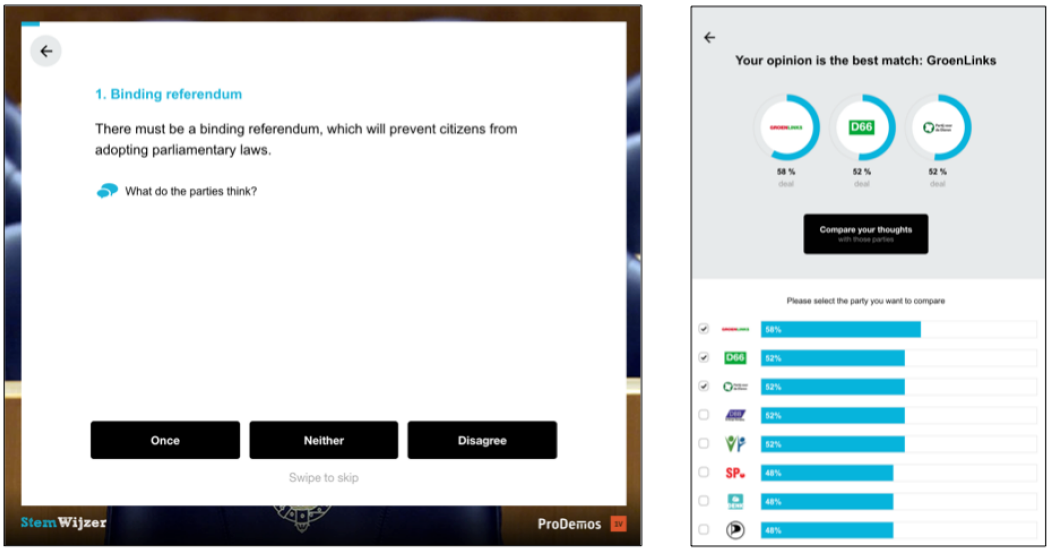


Figure 3.5: StemWijzer. Question and result screens.

with visual impairment. Regardless of the visualization each of the statements can be marked with either "agree", "disagree" or "neither of them". An additional feature of the conventional version is the possibility to see directly which party supports which position by clicking the "What party?" button. This might influence people in their decision and reduces the impartiality.

After answering approximately 30 questions the user can mark topics as particularly important in order to improve their weight in the overall result. Unfortunately, it is not possible to see a user's given responses when highlighting topics.

Afterwards, the user needs to select, similar to the *wahl-o-mat*, all the parties he or she wants to compare their opinion to, whereby at least three parties need to be selected. Compared to the *wahl-o-mat* it is possible to select all the parties using checkboxes. There are shortcuts to select all parties or all incumbent parties at once.

After this step the results are displayed as bar charts with positive percentage values similar to *wahl-o-mat*. Moreover, the top three parties are highlighted and visualized as donut charts, as can be seen in Figure 3.5. Afterwards, it is also possible to compare the results with the top three parties or all parties by clicking through the questions again.

The accessible version abdicates pie charts and bar charts. Only the parties' logos and the degree of agreement in percent is shown. Interestingly, the accessible version does not support to review all questions again and compare user and parties' opinions for each statement individually.

To sum it up, the Dutch *StemWijzer* provides a good user experience by pursuing a

minimalistic approach. Although it would be more attainable to have a questionnaire that targets all voters and avoids stigmatization, *StemWijzer* at least provides an accessible version. However, the accessible version does not provide all functionalities.

3.3.5 Kieskompas

With applications in more than 50 countries the *Kieskompas* [74] is one of the most successful European VAAs. Until 2007 it was the basis of the Belgian VAA. *Kieskompas* offers the possibility to embed the VAA into other websites by providing two differently-sized iFrames [74]. What makes the *Kieskompas* outstanding is that it not only asks about agreement on current polticial topics, but also considers the previous legislature's performance. Moreover, assessements on contestants potential performance as prime ministers are interrogated [130].

Figure 3.6 shows the *Kieskompas*' responsive and minimalistic user interface. Users need to answer 30 questions with answers ranging from strongly agree to strongly disagree. There is also the option to skip questions. Generally, only one click per question is required. During the process it is possible to hide and show a progress bar. This facilitates users to go back and forth questions and review given answers. Some questions offer further details, which can be shown by clicking a button.

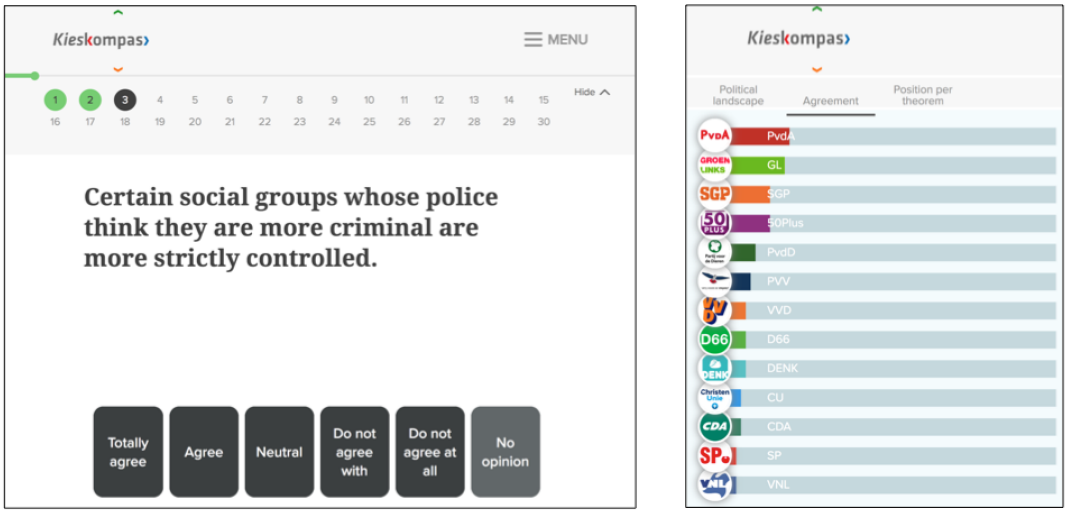


Figure 3.6: Kieskompas. Question and result screens.

Similar to *smartvote*, there are different ways to visualize the result. Firstly, there is a graph mapping the parties on a 2D political spectrum; the x-axis ranges from conservative to progressive and the y-axis ranges from left to right. On the other hand, there is a bar chart ranking in descending order of compliance. However, the labeling is missing for this visualization, not even numbers are given, as presented in Figure 3.6. The only way of interpret the result is to compare the bars with each other. What makes the *Kieskompas*

special is that the result can be refined by selection and deselection of topics that should influence the result.

All in all, the *Kieskompas* offers a minimalistic and clean user interface with multiple different result visualizations, which help users find the optimal visualization for them. What is more is that the *Kieskompas* can be easily integrated in other websites, which certainly helped spread it in many different countries [131].

3.3.6 Vote Compass

The last investigated VAA in the scope of this thesis is the Canadian *Vote Compass* [128]. Having *Vote Compass* Initiatives all over the world, from New Zealand to Germany, the application has gained a lot of recognition since its launch in 2010 [78].

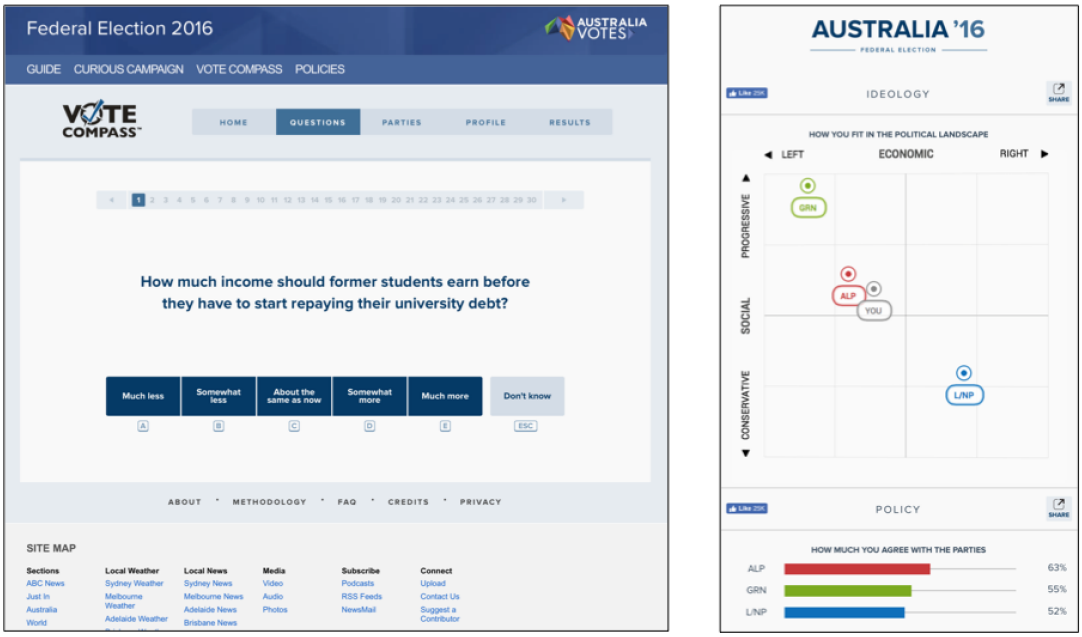


Figure 3.7: Vote Compass. Question and result screens.

Before starting the answering process, the user first needs to state his or her electoral district. Figure 3.7 shows the question screen of *Vote Compass* on a desktop computer. The question is written in bold letters and framed by a progress bar at the top and the list of possible answers at the bottom. A user has five possibilities to comment on current political issues ranging from "Much less" to "Much more", respectively "Strongly Disagree" to "Strongly Disagree" depending on how the question is phrased. There is also the possibility to abstain from statements. The application clearly stands out in terms of operability. Highlighted keyboard shortcuts are positioned directly below each answer button. However, the chosen shortcuts are amendable; instead of using the letters A to E, it would be better to use neighboring keys.

After providing answers to the questions, the application demands further information: firstly, the likeliness to vote for each of the parties; secondly, perceived competence and trustworthiness of nominated candidates using a scale from one to ten. This is followed by two dozen optional demographic questions, before the user is shown the calculated results. Figure 3.7 presents parts of the result screen. As can be seen, *Vote Compass* not only maps the result on a two-dimensional political map, but also shows the degree of agreement as a percentage value below, using bar charts. Colors help to distinguish parties at first sight. However, the visualization does not rely on color alone. Responsive arrangement of the result tiles allows this screen to work on both, mobile and desktop devices.

On this screen the user has the possibility to either mail, print or share his result, whereby the latter allows users to decide whether they want to share the application in general or their particular result. The application offers some additional benefits like mailing the result or modifying the result based on issues individual voters care about the most. Moreover, users can compare parties by issue and therefore find out why results might not be what they expected.

To sum up, *Vote Compass* provides a responsive and accessible, yet aesthetical pleasing experience that requires a lot of data from its users compared to the other applications. However, users are rewarded with a more detailed analysis of their results.

3.4 Conclusions

VAAs have vast differences in terms of visualization, user interaction and methodologies. In this study we analyzed differences in procedure and user experience. Many different applications were examined. See Appendix A for a comprehensive list of discovered VAAs; some of them were examined in greater detail in this section.

Throughout the research it turned out that the most important feature is to be able to start the questionnaire right away. With some applications, especially the ones with a language barrier, it was quite hard to actually find the questionnaire and start it. The questionnaire is the most essential part of the application. Thus, it must be placed at a prominent spot. The Belgian application *wecitizens* exceled in displaying important information: the button linking to the VAA was spotlighted in red. Moreover, at first glance the election date stuck out as well as a simple countdown in days until that day.

Moreover, the length of the questionnaires was quite different. Most questionnaires provided a fixed length of around 30 questions. Exceptions to this rule are, for example, *smartvote*, which has 70 questions in its deluxe questionnaire and the British *Who shall I vote for* with up to 160 questions [59].

Most VAAs have the approach to show users one question at a time, which lets the user focus. Exceptions for this are the American *isidewith* that operates in multiple countries and *smartvote*, which lets users choose whether they want to see one or more questions at once. Moreover, the accessible version of the Dutch *StemWijzer* lists all questions on

a single screen. Only several examined VAAs are responsive, probably due to the fact that they have all been in use for years. Some of them, like the Austrian *wahlkabine.at*, have limited financial means for regular updates. Only a few of the applications were operable using keyboard shortcuts, one of which is the Canadian *Vote Compass*, which does a particular good job displaying shortcut icons below the respective buttons [128].

Almost all the platforms provided additional information about the questions, sometimes in form of a glossary, sometimes just with more information explaining questions more detailed. *smartvote* even provided a pros-and-cons list on every question to give users more insights on logical consequences. Although this is very valuable information, it also causes a lot of editorial overhead [41].

Besides stating whether a user agrees or disagrees with a statement some applications allow to give certain questions additional weight in the result. On *wahlkabine.at* one needs to select a weight for every question. The German *wahl-o-mat* has a separate screen to perform highlighting of questions, which fastens the process. Compared to the Austrian application this can be skipped if users perceive all the topics as equally important. Moreover, letting users assign importance to all questions at once makes comparison easier.

After all questions have been answered, the result is rendered for the user to interpret their matching result. There are different kinds of visualizations. In the most common case, it is just a simple bar chart with values between 0 and 100%, describing the degree of agreement. The German *wahl-o-mat* follows this approach, which can be seen in Figure 3.3.

Instead of having just positive percentage values, the British *Who shall I vote for* and the former *wahlkabine.at* page use positive and negative integer values to highlight disagreement in opinions.

The primary result indicates which party matches your opinion best or which candidate would represent your opinion best, according to the questions answered. However, it is also possible to look at the previous questions answered. *Vote et Vous*, the French VAA, lists all the questions directly below the result. In order to gain more insight, additional information can be revealed by clicking on one of the questions. This will display the answers of the different candidates as well as arguments to defend their positions.

This research revealed instructive approaches of VAAs, which will be used in the redesign process of *wahlkabine.at* in Chapter 5. However, preceding the implementation, a literature review on future-friendly design will be presented in the following chapter.

CHAPTER 4

Future-Friendly User Interfaces

After describing Voting Advice Applications in depth in Chapter 3, this chapter pins down future-friendly user interface characteristics by conducting a literature review, which lays the theoretical background for Chapter 5.

4.1 Overview

The aim of this thesis is to implement a future-friendly user interface, but what does this actually mean? The term "future-friendly" was especially coined by a group of developers and designers in 2011 [140]. They released a manifesto describing strategies how to cope with the increasingly accelerating speed of technological development. They argue that the Internet is at a turning point and started discussion to find ways to anticipate future developments and how to make sure to provide long-term value for users. Their solution encompasses five main ideas:

Simplicity With more and more devices it is important to focus functionality to the essentials. Rather than overwhelming people with too much possibilities, services should be simplified to supply different devices more easily.

Data Structure Defining and storing data in a flexible way is important to remain flexible in case of changes of access. This can be achieved by using standards. Moreover, data should be accessible and modifiable. Data should be the central unit; it should be stored in a way that different entry points can access it seamlessly in order to avoid redundancy.

Adaptability An application should be programmed in a way to be support on devices with different capabilities.

Standard Compliance It is cumbersome to tailor an application to every single device. A better approach is to find a common set of standards that are broadly supported and stick to them.

Device Support People use different devices for certain activities. This should also be considered when developing an application. Thus, not all functionality needs to be supported on every device type.

The overall idea of future-friendly implementations is to facilitate applications to be displayed on devices that have not been invented yet [140].

In comparison to surveys on prolonging software aging [23] or methods to rejuvenate applications [143][26], future-friendly design aims to consider software aging even before starting the development process and not after the software is already in use for a couple of years.

The ultimate goal of future-friendliness could be described as future-proof. However, as Dawson [28] points out, it is impossible to achieve complete future-proofness. Nothing can protect from everything the future might hold under any circumstances, because every application has to deal with software aging [96]. However, what can be done is to consider currently available information to make websites as future-friendly as possible.

This chapter lists and describes these principles. Moreover, strategies will be introduced showing how to implement these criteria. They will shape the result of the final implementation. Some of them are highly interrelated, others might even be contradictory, but all of them need to be carefully considered to find a forward-thinking solution.

4.2 Adaptability

At the beginning of the Internet, websites were created just like magazines in a pixel-perfect manner [28]. As devices have become increasingly diverse, this is no longer feasible. However, devices do not only differ in terms of screen size. There are other factors to incorporate:

Different Resolutions Besides screen sizes, also the resolution can differ a lot and gets better with every year. As a consequence, media of higher quality is needed to provide a good customer experience [77].

Different Operating Systems There are many different operating systems (OSs). Writing separate applications for each OS would result in multiple codebases to be maintained, yet not all devices would be served.

Different Standard Support Due to an increasing number of different vendors, new features and standards are developed with rapid pace. Next to mobile versions of desktop browsers, e.g. Firefox or Safari, there are customized browsers of device vendors [97].

Other devices start visualizing websites as well, but often with lacking standard support [69]. In-app browsers face similar challenges.

Different Input and Output Devices There are different input and output devices available to users. Not all people use mouses and keyboards to operate a computer and have the capabilities to handle these input devices; some people rely on head pointers, single switch devices or Braille keyboards. What is more, touch gestures were added to the list of input methods, at first especially on mobile and tablet devices. Nowadays, these type of interaction is also supported by computer displays.

The same issue is encountered with output devices. A screen being able to display vivid colors cannot be considered the standard user output device. There are eBook readers with e-ink displays that only display black and white [69]. There is the Internet of things (IoT) [44], a network of physical devices embedded with electronics. A website might be accessed in cars, on fridges or other objects; there are endless possibilities. Output devices are especially a problem for visually impaired people who are dependent on screen readers [104].

Different Languages The world becomes an increasingly connected place. Therefore, we also need to ensure that we provide support for different languages, in order to be prepared for a more globalized world.

After all, there is only one Internet and we need to make sure applications are functional across different devices and different capabilities of users [97][104].

4.2.1 How to achieve Adaptability?

In order to achieve adaptability for devices with different resolutions and different device capabilities, a combination of progressive enhancement [95] and responsive design is a promising approach [69]:

Progressive Enhancement

Progressive Enhancement ensures that web technologies are applied in layers in order to provide the right level of experiences depending on a device's capabilities [55]. The basis is a website with good structured, semantic Hypertext Markup Lanuage (HTML), which is supported on most devices. Cascading Style Sheets (CSS) are applied to specify styles for certain elements. Finally, a layer of JavaScript complements the application by introducing behavior. The basic concept of progressive enhancement is to ensure universal usability and accessibility, by making sure that basic HTML structure is usable right from the beginning [55], rather than creating complex structures and trying to fulfill accessibility criteria afterwards.

The opposite of progressive enhancement is graceful degradation; with this approach a fully functional website is developed first; functionality and content is gradually removed from devices with reduced capabilities [18].

The main principle behind progressive enhancement is fault tolerance. HTML and CSS are backwards compatible. They support all past features and ignore unknown features [55], thus ensuring good results on all devices.

Responsive Design

Coined by Marcotte [79], responsive web design makes sure that a website adapts to the context of use. Rather than developing a website for every devices, responsive design makes sure that one design fits all the screen sizes and resolutions [110]. This is achieved through the following three concepts:

Flexible Layout Traditionally, websites were based on fixed-width layouts, just like magazines and newspapers. However, as more different devices with varying screens sizes emerged the need for more flexible layout came up as well [141].

Flexible images Besides layout, images need to be adaptive as well. Setting fixed sizes on images result in the image being really small on big screens and too big on small screens. Flexible images adapt to the context [141] and therefore also save bandwidth, as smaller images are loaded on smaller devices [44].

Media Queries Media Queries enable the definition of separate CSS for different devices and screen sizes. This helps to display the structure defined in the HTML in different ways, according to the device capabilities [141].

Although there is a trend towards abandoning native applications in favor of responsive web applications [68], Podjarny [101] argues that only 11-12% of websites make use of responsive design. However, it is a first important step towards future-friendly serving different devices that are about to come [97].

4.3 Accessibility

Similarly to how we ensure that our physical environment is accessible to people with disabilities we also need to make sure the Internet is. It is good practice to ensure web accessibility and it might even be required by law in the future [89].

The World Wide Web Consortium (W3C) issued the Web Content Accessibility Guidelines (WCAG) to help implement accessible websites. This set of guidelines invite to build accessible websites and thereby ensure that all people can access the vast resources of the Internet equally [104]. The Web Accessibility Initiative (WAI) published the first draft of WCAG 1.0 in 1999. Being around for almost a decade it was replaced by WCAG 2.0 in 2008 [104]. This should be a more future-proof set of accessibility guidelines, as

it removed the technical aspects. The adoption rates of these guidelines are still rather low. As Goncalves et al. [52] point out, all websites of the largest 250 Forbes enterprises he assessed have a considerable number of accessibility errors, although they should set a good example for other, smaller companies. WCAG 2.0 was designed around the following four principles [129]:

Perceivable A user interface must be presentable in such a way that all users can perceive it, even if they have special needs. This principle includes guidelines that all non-text content must have a description in order that blind people can perceive its content with screen readers. Another example is the use of appropriate colors; enough color contrast needs to be ensured. Furthermore, information should never be conveyed using color encoding only.

Operable Besides perceivability an accessible user interface needs to be operable irrespective of the input devices users might use and avoid actions they cannot perform. This also includes giving the user enough time as fulfilling a task might take longer on one device than on another.

Understandable Content and user interaction need to be designed in an understandable way. On the one hand, this refers to the way the content is written. On the other hand, the user interface should behave as the user would except it to. Elements should behave and appear in a predictable way for the user and support the user to avoid mistakes.

Robust Robustness refers to the compatibility with current and future devices that are used to access a websites. This also includes assistive technologies like screen readers or switches, amongst others, but also more trivial things like the browser used.

4.3.1 How is Accessibility achieved?

By following these principles and guidelines a website will be more likely accessible. The guidelines have certain success criteria that differ in levels of importance. Depending on how many of these criteria are fulfilled the following conformance levels are distinguished [129]:

Conformance level A Websites with this conformance level comply with all success criteria for level A; this is the baseline that important information is accessible to all. It is the lowest conformance level ensuring a minimal level of accessibility.

Conformance level AA This conformance level is achieved by complying to all success criteria for level A und AA, thus ensuring no user experiences difficulties accessing content.

Conformance level AAA This conformance level ensures that all information is accessible to everyone. It is the most difficult level to achieve and demands compliance to all success criteria for level A, AA and AAA, even success criteria that are optional.

The higher the conformance level the more design restrictions need to be faced. It is not necessarily important to comply with the highest level of conformance, as this is impossible for some types of content. However, at least a minimum effort should be made to ensure that the platform is accessible to everyone. The entire list of accessibility guidelines can be seen on the website of the W3C [129] and should be read before implementation in order to consider important aspects beforehand.

Accessibility is an important concept to make the Internet equally available to all users. Although some criteria are often difficult to achieve and conflict with aesthetics, it is important in order to guarantee equal user treatment.

4.4 Findability

According to Morville [89] findability can be defined as "The quality of being locatable or navigable or the degree to which a particular object is easy to discover or locate." The main goal of this property is that users find what they are searching for. This principle needs to be considered in two fundamentally different ways:

External Findability Search engines have become an access point for websites [97]. The Internet is increasingly flooded with more content; thus, it is essential to be found by potential users. Search Engine Optimization (SEO) is not just a plain marketing tool anymore [44], but helpful to users. Therefore, SEO is crucial when it comes to implement a future-proof application.

Internal Findability However, external findability alone is not sufficient. It is equally essential that users find what they are looking for on the website itself. Therefore, it needs to be ensured that a web page is structured in a way that information can be found where a user would expect it to be. This also relates to a principle of accessibility.

Finding something faster goes along with a reduced energy consumption while searching [44]. This is also in line with sustainability, which is discussed in Section 4.7.

4.4.1 How is findability achieved?

As mentioned above findability of a website can be achieved by optimizing a website for search engines. Various guidelines exist about this topic, some of them even published by Google itself. The essentials include defining a "robot.txt" file for a web crawler to facilitate harvesting a website's information [115]. Additionally, a sitemap with all for search engines important Uniform Resource Locators (URLs) should be defined [115].

Moreover, Open Graph Meta Tags, a set of meta tags added as a W3C standard to HTML5 help to define semantics of a webpage [97]. They support spreading content on social media platforms because they integrate a website as an object on social media [5]. When they are not explicitly defined, these platforms use internal heuristics to define what the content of your website is [108], which often induces unpleasant results.

Additionally, content needs to be structured in a meaningful way for the user to find it [44]. Rethinking prioritization of content on mobile devices is also an important measure to help a user find what he or she is looking for. As there is limited space on smaller devices, more effort is required to structure the content to make navigation intuitive [72]. Also, not all the content needs to be displayed right from the beginning. Revealing more content, if users request it, is a better approach [72]. If everything seems to be relevant, jump functionalities can be provided to let users jump to the page section of interest, rather than letting them scroll too long [72]. In order to get an overview of the available content and to structure it in meaningful way, a content audit is a helpful tool [44].

Adding an internal search is another way to facilitate findability [44], if there is a lot of content available.

4.5 Maintainability

According to Parnas [96] there are two causes for software aging: firstly, software is not modified according to the changing environment; secondly, bugs often result from performing a change in the source code without sufficient knowledge. To diminish the second risk, software needs to be easily maintainable.

Of course an effort should be made to implement the application as age-prolonging as possible; the optimal result would be that maintenance is not even necessary. However, some adaptions will be inevitable because content presumably needs to be altered or added. Thus, we need to make sure that the code is easily modifiable; a web developer with sufficient experience should be able to incorporate changes without problems.

4.5.1 How is maintainability achieved?

According to ISO 9126 [66] one notion of software quality is maintainability. It is divided in the following subcharacteristics:

Analyzability This characteristic describes how easy it is to find a problem of a system and how easy it is to identify the piece of code that needs to be changed. The less code there is and the better it is structured the easier it is to find it.

Changeability Changeabilty refers to the ease a problem can be fixed. Separating code layers [55] is, for example, a good practice to make parts of the software easily interchangeable.

Stability This refers to how stable the system is running while changes are performed.

Testability This criterion is fundamental to verify, after an alteration of the system, if it is still working as intended.

Based on these criteria, Heitlager et al. [60] present a practical model to measure maintainability. There are several general strategies one can pursue to keep the maintainability as low as possible:

Clean Code

Clean Code is a practice to program in an efficient yet readable way such that a third party can alter a program without being part of the initial development process. This can be achieved by naming functions, variables, files in a meaningful way and encapsulate behavior in meaningful, reusable functions. Comments should only communicate the purpose of the code, not the functionality, because this can be read from the code anyway. Proper error handling and testing, together with appropriate formatting according to coding conventions, complete a maintainable coding base [82]. Writing bad code might be initially faster, but in the long run this will slow programming down considerably [82].

Reduce Codebase

An application needs to be developed for every platform that must be supported [68]. Through responsive web design one only needs to maintain one codebase [72]. Changes need to be carried out only once. Therefore, it is also easier to verify that all devices have an up-to-date version of the platform [18]. Moreover, a smaller codebase is also easier maintainable. The smaller the codebase is, the easier it is to find specific code segments. If content changes frequently or needs to be added regularly, one codebase can help to reduce maintenance costs considerably [141].

Separation of Concerns

It is common practice to separate code in layers [55]. Rather than putting all CSS inline in HTML code, it is better to keep it separate and put it in an extra file. This facilitates the exchange of layers. Instead of looking through the entire codebase for occurrences, this makes sure that all code is at one place in case one layer is to be exchanged. This will also ensure that changes are only necessary for the replaced layer and not throughout the code.

Every piece of software is going to deal with age-related bugs. If software is developed in a maintainable way, risks of software modifications can be mitigated. The ultimate goal of maintainability is prolonging the software lifecycle [23].

4.6 Performance

Web performance can be described as the amount of time it takes from the moment a URL of a website is entered until the page is fully rendered [9]. The fastest website is the one with the leanest code and the fewest assets to load [44].

A website's performance is crucial and in the best case the user would not notice any delay at all [9]. If a website performs poorly, users might not be willing to wait, might opt for a competitor's solution and will not come back again [44].

In 2009 Google experimented with the performance of their search functionality [72]. They slowed their service down and found out the weaker their search was performing the fewer search requests a user was conducting. This also influenced the users on a long-term basis as their did not went back to their previous usage level, even after the experiment was over.

Performance is especially an issue for responsive websites [100], as the same version of a website is displayed on all devices, from a desktop computers to a mobile phone. Web performance researcher Guy Podjarny [100] found out that 86% of 347 responsive websites, he had tested, had no or only little performance savings when they were loaded on the smallest supported screen size [72]. Additionally, also latency is worse on mobile devices, because they are not as fast loading and processing information from servers, which slows down applications even further [72].

4.6.1 How can we improve performance?

Barker has summarized the quantitive indicators of performance as follows [9]:

- Page load time
- Page file size
- Number of Hypertext Transfer Protocol (HTTP) requests
- Page render time

In order to optimize performance these indicators need to be minimized as much as possible. Page load can be decreased, for example, by minimizing CSS and JavaScript code [44]. Number of HTTP requests is reduced by concatenating different JavaScript and CSS files into joint files.

Moreover, websites should not only be designed mobile-first, but also programmed in a mobile-first manner. Devices with a smaller screen should only load data with the required file size, which is especially relevant for images [72]. In general, only content that is necessary should be loaded.

Google, Yahoo and many others have published common recommendations to improve performance, which are neatly summarized by Frick [44]. Another approach to enhance performance even further is conditional loading described by Podjarny [101].

In the end, it is often a trade-off between performance, aesthetics and maintainability. A Content Management System (CMS) might be really helpful and future-friendly, but it can also slow down the performance considerably. Similar issue arise with frameworks like Angular: they might help build a website very quickly, but if only a small fraction of its functionality is needed, it might cause unnecessary overhead [44].

A further approach to optimize a website's speed is the use of caching as Gardner [45] points out.

4.7 Sustainability

"Humanity has the ability to make development sustainable to ensure that it meets the needs of the present without compromising the ability of future generations to meet their own needs." [137]

This definition by the World Commission on Environment and Development (WCED) from 1987 can also be applied to the Internet. According to a report issued by Greenpeace USA [24], the Internet has far more environmental impact than one might think:

> "If the Internet were a country, it would be 6th largest user of electricity behind China, the US, Japan, India and Russia." [24]

As the data Greenpeace is referring to is already from 2011, the current situation might even be worse, considering how many hours of Netflix are streamed daily, how many videos are uploaded on YouTube and how much time people are spending on social media. There is even the thought that the Internet might not be more sustainable or environmentally friendlier than books, its printed counterpart, considering the energy consumption [44]. Up to 40% of the Internet's energy usage might be caused by applications' front-ends. For this reason it is important to make especially user interfaces more sustainable.

Frick [44] claims that a sustainable Internet offers many benefits. Besides decreasing the environmental impact, it also improves performance, usability, search engine results and accessibility.

4.7.1 How can sustainability be achieved?

So far, we discussed why sustainability is desirable. To actually implement a more sustainable web, Frick [44] categorized the activities in four categories:

Sustainable Components One approach would be to use green hosting and offer sustainable alternatives for your user, like sustainable shipping.

Sustainable Content This refers to the ease with which users can find what they are looking for, on a search engine as well as one the website itself. This is described in more detail in Section 4.4.

Sustainable Design User Experience is about efficiency, effectiveness and satisfaction and will be covered in more detail in Section 4.8.

Web Performance Optimization (WPO) Performance refers to the speed of a website and is further discussed in Section 4.6.

All these issue might seem insignificant when being evaluated for a single device. However, considering how many devices access web applications on a daily basis, sustainable web applications can make a huge difference overall.

As one can already see, all criteria and measures discussed in this chapter are highly interrelated. As the Internet has such an high energy consumption, sustainability is an important issue when developing future-friendly software.

4.8 Usability

To have a future-friendly user interface, usability needs to be ensured in the present. Abdullah et al. [2] identified that one of the reason for software aging is a lack of user-friendliness.

According to ISO-Standard 9241-11 [65], usability can be described as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use".

This three measurable elements are the core of usability:

Effectiveness Effectiveness describes the completeness with which a certain task can be executed by an application [65] [58]. This is usually measured by observing test users.

Efficiency Efficiency is how good and fast a task can be achieved. This can be evaluated in different ways, like the number of clicks needed to fulfill a task or the time needed to complete an activity.

Satisfaction Satisfaction is the subjective perception of the user and other people affected by the use of the application. This also includes aspects like fault tolerance. A user should be able to navigate freely through website without fearing that an action might undo everything he or she has done so far.

Other approaches include additional attributes in their research, like learnability and usefulness [109]. Ease of use is an essential point when talking about future-friendliness of a platform because great usability today is the must-have usability of tomorrow [70].

4.8.1 How is usability ensured?

In 2005 Nielsen [91] presented a set of ten usability heuristics that ensure software quality. This should be kept in mind while designing and implementing a usable application. Usability evaluation should be conducted from the very beginning in order to avoid costly usability problems.

There are two different approaches to identify usability defects: usability testing and usability inspections.

4.8.2 Usability Inspection

Usability inspections are important tools to find usability defects. They are conducted by usability experts that use their knowledge to identify usability issues. Several different methods to inspect were proposed amongst others [61]:

Heuristic Evaluation

Heuristic Evaluation is conducted by usability experts examining a user interface based on a set of usability guidelines [92]. It is a rather cheap way of identifying usability problems. However, opinions diverge how effective this way of gathering feedback really is.

Cognitive Walkthrough

A Cognitive Walkthrough (CW) [134] involves usability experts clicking through a user interface having a concrete user task in mind. While exploring the application, the expert sets a goal for every step of the way, evaluates the currently available actions and determines which of these helps them reach their goals. After performing the walkthrough, feedback is collected and thereby determined whether the system is intuitive. There are many variations of the CW; Spencer (2000) introduced a Streamlined Cognitive Walkthrough (SCW) to overcome lengthy design discussions and design defensiveness [120].

4.8.3 Usability Testing

In contrast to usability inspections where experts are brought in to identify problems, usability testing is conducted with actual users. In its initial form it was conducted with many indiviual participants. However, in recent years many group testing approaches emerged to maximize efficiency. We want to especially spotlight GUT because this a central methodological approach of this thesis.

Group Usability Testing

GUT [36] includes several participants simultaneously, but individually, going through a predefined set of tasks. One or several testers are observing the group and taking notes. The session is followed by a focused discussion involving users in problem identification and collaboratively finding solutions. GUT produces lots of useful input within a relative short time and therefore helps identify major usability issues.

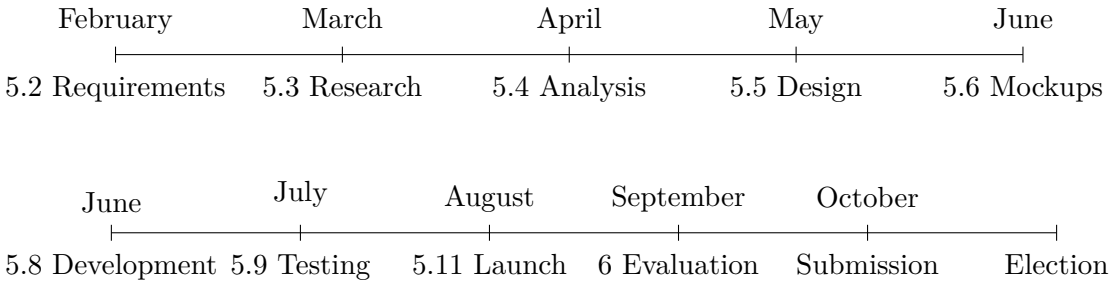
This chapter proposed a list of future-friendly software properties that need to be paid attention to when developing a new platform. Some of them are highly interrelated, as performance and usability are also subcategories of sustainability. Some of these criteria even are contradictory; more high-performance code often means reduced accessibility. All in all, an adequate balance needs to be struck to ensure good user experience for everyone. The next chapter will describe the development process of *wahlkabine.at*, where all these proposed principles are incorporated.

CHAPTER 5

Relaunch Wahlkabine.at

In the previous chapter a literature review on future-friendly user interfaces was conducted in order to serve as a background for the implementation process. The aim of this chapter is to outline the development process of *wahlkabine.at* from the first meeting in February 2017, when requirements were defined, until the platform's final launch in late August 2017. A timeline is used to visualize the development process before each milestone will be described in more detail.

5.1 Overview



In February, after the first meeting with all stakeholders, a list of requirements was drawn up as described in Section 5.2. Section 5.3 talks about research that was necessary to be conducted before the analysis of the current platform, as can be seen in Section 5.4. After adapting the visual design in Section 5.5 to reflect a new image of the platform, mockups, as explained in Section 5.6, were created to get some input on usability issues, before implementation. In June the programming started and was going on until the end of August, as described in Section 5.8, while feedback was continuously collected from

the Institute for New Culture Technologies as well as actual users using a UCD approach. As a last step the platform was evaluated against identified criteria. This section can be found in Chapter 6. The finished platform was launched on the 24th of August, seven weeks before the general elections on the 15th of October, 2017.

5.2 Requirements

In the first meeting in February the primary requirements for a relaunch of *wahlkabine.at* were discussed. First and most important of all, the basic functionality should remain the same. It is a quite unique situation to have an entire feature list of the final product beforehand because the platform has already existed for more than 15 years. The user should be able answer around 25 questions either with "yes", "no" or "no answer". The last option can only be chosen a maximum number of five times to avoid distortion of the result. The questions should shuffle randomly, so that no narratives are built.

The questions also need to be weighted with a numerical value from one to nine. The application's algorithm should remain identical; thus, the same user input data is needed as before. The algorithm is a scientific topic and relies on extensive testing by political scientists before implementation, which is why it was not modified. More details on how the algorithm works can be found in Stefan Fenz's thesis [39] about online VAAs.

After answering all questions, users get a list of political parties sorted in descending order of agreement. The result can be shared on social media such as Twitter and Facebook.

The look-and-feel of the new user interface should not change much, so that users do not need to learn an entirely new platform. However, the new platform should be - in contrast to the old platform - accessible as there were complains in the past by people with special needs; they were not able to use the website.

The application should be usable without problems on all kinds of devices: mobile phones, tablets, mobile watches with browser, televisions and ordinary desktop computers. It should be as simple as possible in order to let the user focus on the task at hand.

An essential requirement is color neutrality. As each political party is usually associated with a special color, it should be made sure that none of these colors is used excessively on the website to not influence voters.

Another important demand of the new application is good performance. As there are certain peak times where particularly many people access the application, it needs to be programmed in a way that it can withstand such peaks.

Also maintainability is an important aspect. The institute wants to take over the platform and run it without our help as soon as the master thesis is finished; this should be as unobstructed as possible.

All important content from the previous application should remain, like the statements pages, news, literature and press releases. However, redundant content can be removed in accordance with the Institute for New Culture Technologies.

Other requirements include security, metrics, and the possibility for the administrators to adapt and create new content; these aspects are part of Andreas Taranetz' [122] research.

The Institute for New Culture Technologies stated, that they are very open to suggestions. In order to agree on upcoming issues, regular meetings are held.

At the beginning of the thesis the date of the election was expected to be in spring of 2018; thus, initially there was no time pressure to finish the platform. Only in May 2017 it was announced that the federal elections would be preponed to the 15th of October [12].

5.3 Research

After all the requirements were put on record, the first step was to conduct research. First of all, a review on existing VAAs was carried out, as described in Chapter 3. Aftwards, a literature review about future-friendly user interface design was executed, which resulted in a list of principles as presented in Chapter 4.

These properties include in alphabetical order:

Accessibility
The platform needs to be equally usable for people on different devices and with different capabilities.

Adaptability
wahlkabine.at should not only work with new devices and new functionality but also with current and future technology.

Findability
Content and functionality should be structured in a way that people will find something without searching for it.

Maintainability
The application itself should be developed in a way that future developers can modify it without problems.

Performance
The application needs to endure the load at peak times before elections.

Sustainability
The application should use as little resources as possible to function properly.

Usability
The application should help the user in an efficient and effective way [65] to compare their opinion with the views of political parties.

Sometimes during the development process important decisions need to be made. The listed criteria do always have a stronger negotiating power than other properties of the platform, such as aesthetics for instance. Therefore, these criteria are fundamentally helpful in the decision making process. After finishing this research process, the analysis of the current version of the VAA and its content will be dealt with in the next section.

5.4 Content Analysis

The content is the heart of a website. For this reason design considerations often start with identifying and structuring content. As only information should be shown that provides value, it should be considered carefully what to publish [72]. In order to get an overview of the available content and its underlying structure it is helpful to conduct a content audit, as Frick discusses in his book on sustainable design [44]. A content audit can be compared to an inventory and helps summarizing content structure and currentness [69].

For structuring the content of *wahlkabine.at*, a combination of Veen's [127] and Spencer's [119] audit approaches were applied. This resulted in the following set of fields:

Page ID
A unique identifier to use as reference

Page Name
A title of the primary content of a particular page

Page Link
A page's URL

Content Hierarchy
Structure of a page's content

Content Description
A description what the content of a page is about

Comments
Further remarks that are relevant for implementation

The previous *wahlkabine.at* platform was carefully examined. For each page and each subpage, a row in a spreadsheet was created until all pages were recorded. Afterwards, outdated or duplicate information was discarded in collaboration with the Institute for New Culture Technologies. Related information on separate pages was consolidated on joint pages. In the end this audit process helped coming up with a sitemap for the new platform.

5.5 Design

After having analyzed the existing content, as described in the previous section, the next step was to define key design elements of the new platform which include colors, fonts, graphics, buttons and navigation. In the end, a style tiles was created, summarizing the design elements at a glance.

5.5.1 Colors

Colors play an essential role in our lives. It is one of the first things a user notices about a web platform. Besides contributing to a website's aesthetics, there is much research about colors and their psychological factors. Colors also evoke certain emotions and facilitate recognizing and understanding of complex relationships.

Figure 5.1: Initially, Turquoise was chosen to be the primary color.

A new color scheme needs to be developed for the relaunch of *wahlkabine.at*. Not only to provide a modernized look, but also to comply with common accessibility standards, which were not supported by the old color scheme, due to a lack of color contrast. Figure 5.1 presents the final color (at this point of the thesis). A vivid turquoise is chosen as it provides sufficient color contrast, emits trustworthiness and is also considered an aesthetical pleasing choice. The following sections describe how accessibility, party association of colors and color perception influence this choice. Moreover, a color scheme will be presented at the end of this section.

Figure 5.2: Colors auf Austria's political landscape.

Party association

VAAs compare positions of political parties. Each of them is usually associated with a color, which not only helps voters to easily recognize a party, but also to distinguish parties from one another; thus, colors reduce the visual complexity of the political landscape [113].

Political parties' colors need to be taken into account when designing a VAA, because no particular party's color must be overly exposed on the platform; a VAA should be a neutral ground. When choosing a main color for *wahlkabine.at* it is important not to choose one that is too similar to one of an Austrian party. Figure 5.2 shows political parties' colors of Austria's previous federal elections in 2013. As of 2017, the Alliance for the Future of Austria (German: Bündnis Zukunft Österreich (BZÖ)) and Team Stronach (FRANK) do not exist anymore. Moreover, the Communist Party of Austria (KPÖ) claimed the yellow color of FRANK. At first glance it might seem that not many neutral color options remain. To better analyze the situation, Figure 5.3 pinpoints the political parties' colors onto the circular color spectrum. It is apparent from the visualization that turquoise, orange and purple are possible color choices.

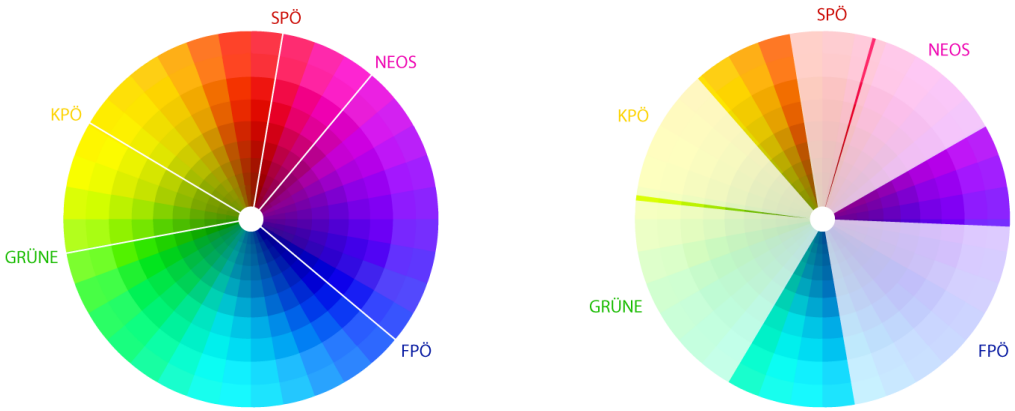


Figure 5.3: Political parties' colors mapped onto a color wheel.

Accessibility

As accessibility is an important aspect for the new platform, this is an important consideration reflected in the color scheme. Sufficient contrast is crucial for text to be perceivable for people with reduced visual acuity. According to WCAG 2.0 level AA a minimum contrast of 3:1 is required for large text. Level AAA even requires a ratio of 4.5:1[129]. What is more, approximately 8% of humans suffer from visual impairments [94], which means, that they are not able to distinguish colors properly. Thus, it is important that colors do not only differ in hue, but also in brightness. In order to make sure people can still perceive all given data, information should never be conveyed using color only.

Color Perception

Colors are not only related to political parties, but with meaning in general. Each individual color is associated with unique properties and emotions; this does not hold to be true around the world because different cultures perceive colors differently [16].

wahlkabine.at is developed for Austria, which is why we focus on color perception of Western Culture. What colors mean and which emotions they evoke is studied intensively and cannot be neglected when designing a user interface. Kim and Moon [73], for instance, examined interfaces of a banking website and came to the conclusion that color has a main impact on a website's perceived trustworthiness. Alberts et al.[4] came to the conclusion that blue and green are considered more trustworthy than red and black, for example. Besides trustworthiness blue is associated with reliability and safety [16]. Green is considered to promote environmental topics and well-being, while it also stands for innovation and growth. Purple is associated with luxury products and sometimes perceived to emit mystery. Pink is utilized for feminine topics and connected to romance [16]. Warnings and dangers are often visualized using shades of red because it is the color that attracts most attention. It is also frequently used for food related matters. Orange is a friendlier version of red, giving an impression of energy and cheerfulness [17]. Yellow promotes happiness and enthusiasm, while darker shades signalize antiquity [15].

As a general rule studies show that people prefer short wavelength colors (blue, green) to long wavelength colors (yellow, orange, red) [56]. A study on color preferences by Guilford [54] in 1957 showed that colors are preferred in this particular order: blue, green, purple, red, orange and yellow. This is also confirmed by an evaluation in the course of this thesis: A hundred people were shown a newly designed *wahlkabine.at* graphic in three different colors, namely orange, purple and turquoise. Almost 100% of the interrogated people preferred the turquoise option. Only a minority preferred either the purple or the orange options; thus, turquoise was in the end the color of choice, while orange remained an additional color for highlighting. In order to have complimentary colors to convey more complex information, a color scheme was looked for.

Figure 5.4: Color Scheme of the new platform.

Color Scheme

There are different methods to generate a color scheme. We opt for a compound complementary color scheme [17], which is a combination of two pairs of complimentary colors, therefore, having warm and cold color elements to visualize information. Figure 5.4 illustrates the final color scheme. Turquoise is chosen as primary color, complemented by its complementary color yellow. Purple and orange are additional colors in case more complex information needs to be visualized requiring more than two colors.

5.5.2 Fonts

There are thousands of fonts to choose from. One can opt for variants with or without serifs, monospace, handwriting or decorative fonts; the typographical choices are endless. In terms of integrating fonts into a web application, there are basically three different options [44]:

System fonts

System fonts are already installed on a device. Therefore, no extra files need to be downloaded. However, as a consequence, this restricts the font choices to select from.

Embedded fonts

Embedded fonts can be included in a website's CSS. Although the advantage is endless font possibilities, it also increases the page size, making it a less optimal choice in terms of performance and sustainability.

Web fonts

Web fonts are hosted on external servers and therefore increase a page's number of HTTP requests. To increase performance they are often not loaded on mobile devices [69]. The obvious advantage is that fonts can be displayed, even if they are not installed on a device.

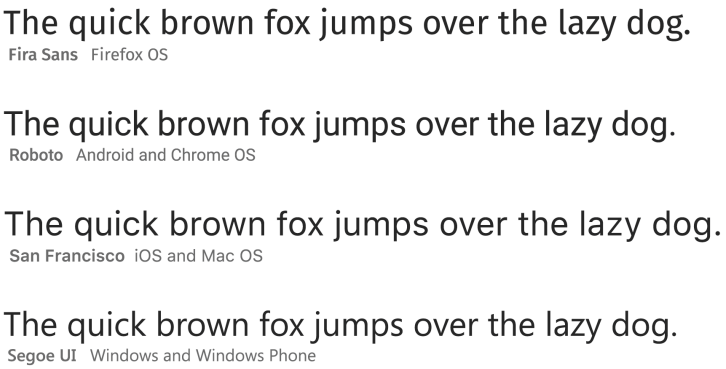


Figure 5.5: System fonts on different devices [135].

For *wahlkabine.at*, we decide to opt for system fonts because it is the most sustainable approach [44], as additional HTTP requests are avoided and the page size is therefore decreased [44]. This is especially important when a huge amount of users accesses the platform at the same time. Although system fonts make it quite unpredictable to see

how a platform will look on every single device, sustainability and performance are more important for this project than appearance.

By using a system font the application might also look more intuitive to users because they are already familiar with the font on a daily basis. In terms of readability there is no significant difference between serif and sans-serif fonts [88][11]. We opted for sans-serif fonts because it is in line with the minimalistic design approach [25]. Figure 5.5 illustrates pangrams of system fonts of popular operating systems.

Generally, it is a good approach to try to reduce the number of fonts used on a site. Thus, we choose to use the same system font throughout the application with different font weights [44]. From an aesthetic viewpoint it is also better to reduce the number of fonts.

Concerning font size, we opt for relative font sizes because it is a more sustainable choice than sizing fonts in pixels [69]. Different browsers handle zooming of fixed font sizes differently; Internet Explorer does not support it at all [69].

The default font size for running text on most browsers is 16 pixels [69]. To overcome an Internet Explorer bug, the default font size needs to be defined as 100%; the remaining text elements, such as headings, are styled using em units. Running text therefore is 1em and headings are increased appropriately, using these em units. To increase readability, not only color contrast was important, but also line height was increased and a minimal font size of 16px was chosen. The finally chosen font sizes can be seen in Figure 5.7, which presents the style tiles.

5.5.3 Logo

In the course of the *wahlkabine.at* relaunch also the logo was modernized. Figure 5.6a shows the old visual identity of *wahlkabine.at*. It is a combination of icon and bold sans-serif text as well as a slogan remarking it is Austria's most successful VAA. The background shows a 3D polling booth.

In the course of 15 years *wahlkabine.at* has become quite popular in Austria. This is why, it is not necessary anymore to incorporate an application description in the logo. Moreover, also the background does not convey any additional information, which is why it can be considered as visual noise.

(a) Old logo of wahlkabine.at

(b) New logo of wahlkabine.at

Figure 5.6: Comparison of old and new wahlkabine.at logo.

Figure 5.6b presents the new logo of *wahlkabine.at*. It was designed following a minimalistic approach. Icon and font sizes were adjusted to be more similar, and the font weight was also chosen as to match the icon as good as possible. Background and description were removed in order to achieve a clearer appearance.

5.5.4 Style Tiles

At the end logo, colors and fonts were combined into so-called style tiles. Style tiles are an early deliverable to convey a design's message [44]. As it is independent from the actual layout, it was an important step to gain feedback on the design elements from the Institute for New Culture Technologies, which has the final decision on such matters. As described by Kadlec [69], the style guide was directly created using HTML and CSS in order to make it reusable for the developing process. Figure 5.7 presents the style tiles at this point of the thesis.

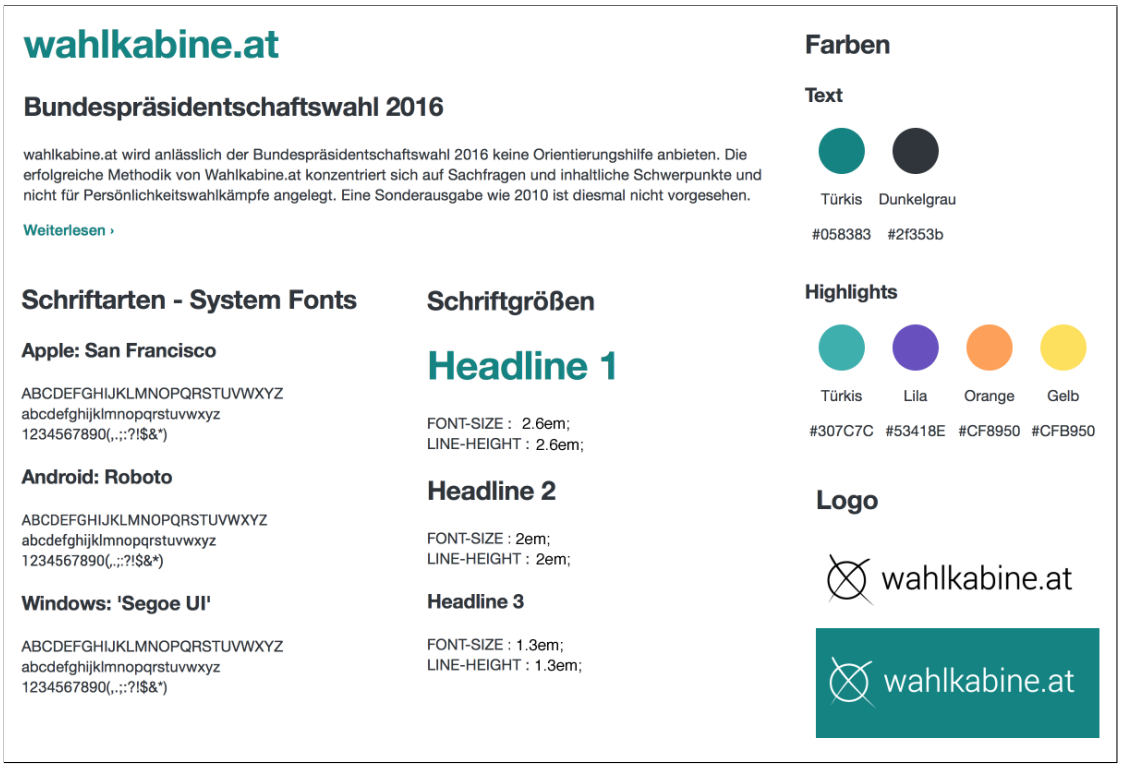


Figure 5.7: Style tiles for wahlkabine.at

5.6 Mockups

After having defined colors and basic design elements, mockups were created to gather feedback on user interface design from the Institute for New Culture Technologies before

the actual implementation. The core functionality of the platform is giving voting advice. Although mockups were created for all pages, in this section we will concentrate on the mockups for the question, ranking and result screens of the application. First, there are some general remarks on the mockups before each of the screens is discussed in greater detail.

5.6.1 General Remarks

Studies show that more and more people are accessing the Internet solely via smartphones [69, p. 162]. For this reason the mockups were created in a mobile first approach. With this approach the design for mobile devices is created first, before modifications are made to enhance the experience on larger devices [76, p. 105]. An effort was made to make it as minimalistic as possible to emphasize important content and avoid user distraction.

Hall [56] pointed out that user interfaces using black and white as main colors communicate professionalism and are, as a consequence, very suitable for educational purposes. Moreover, this color combination also facilitates readability due to a high contrast ratio. This is why especially running text was visualized in this way. In the same study Hall [56] identified that the use of colors make a website more visually appealing. Thus, color was used in the user interface as well. However, an effort was made to use color only for interface elements with a particular function, which means it can be clicked, for example. By doing this the user can perceive at a glance which items have a function and can be interacted with.

Talking about navigation, a hamburger icon has become the prevalent metaphor for hidden menus [27]. In order to assimilate mobile and desktop screens this established standard was applied for all screen sizes, hiding a full-screen menu.

5.6.2 Question Screen

The question screen basically consists of three sections, as can be seen in Figure 5.8. On the top there is the progress section where users are informed about their current status. As Nielsen [91] points out, it is essential to keep the user informed about their current progress. On both sides of the question counter there are arrows helping a user to navigate back and forth between the questions. These are positioned on the appropriate sides, so the natural order is followed. On the mockup for larger devices all the question numbers are displayed, whereby only the ones already answered are highlighted. This should fasten navigation, as users can directly click on the number they want to jump to.

The major part of the screen is captured by the current question. Being the most important piece of information on the screen the font size needs to be especially big to be easily legible. It scales with screen size and is always centered vertically and horizontally within the available space. Glossary words are shown in turquoise to emphasize their clickability.

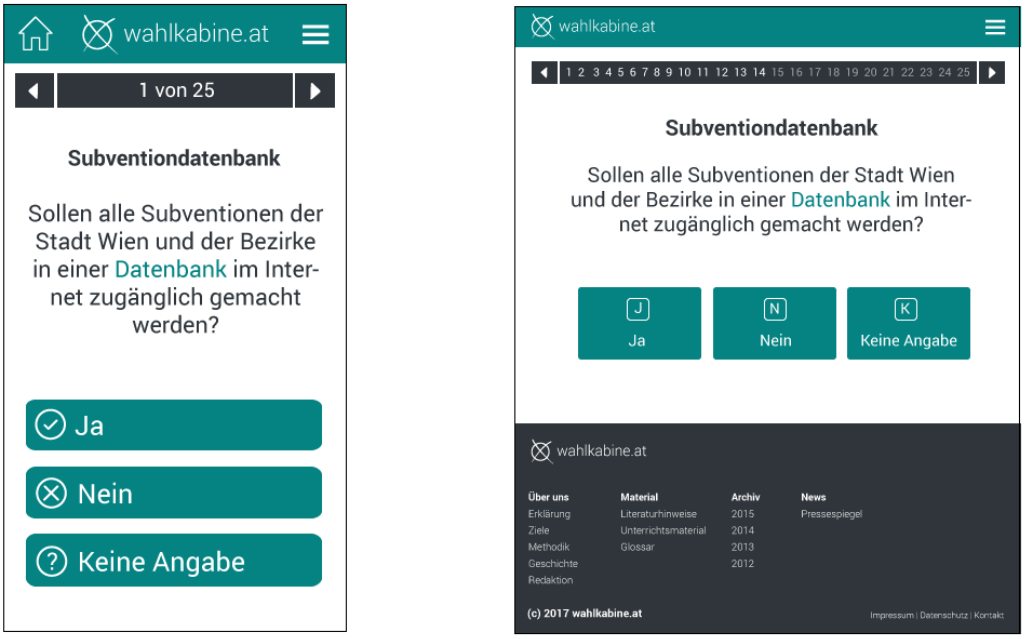


Figure 5.8: Mockups for the question screen for mobile and desktop.

At the bottom are three buttons, one for each possibility to answer the questions. Depending on the screen size, they are either aligned horizontally or vertically. On larger screens these buttons are bigger to make them comfortable clickable on touchscreen devices and to use the available space. In addition to operate the application with a mouse, answers should be selectable using keyboard inputs as well. Keyboard shortcuts are indicated inside the respective buttons they replace. By having a frame with a border radius they emulate the appearance of keyboard keys. These shortcuts are only indicated on desktop devices.

5.6.3 Ranking Screen

Besides stating whether they agree or disagree with around 25 questions, users need to weight the issues using a value from 1 (low priority) to 9 (high priority). On the previous *wahlkabine.at* platform this was done directly on the question screen. However, we argue it would be better to perform all the weighting at once, rather than with every single question, thereby following the example of the *wahl-o-mat*. Firstly, it is easier to relate importance of different questions when having all of them at a glance. Secondly, users can skip ranking if they consider that all questions are of equal importance.

In order not to force users to read the questions a second time for the weighting process, each of them comes with a short name, which also serves as the header for a collapsible section holding the entire question text. On the left of this collapsible section an icon,

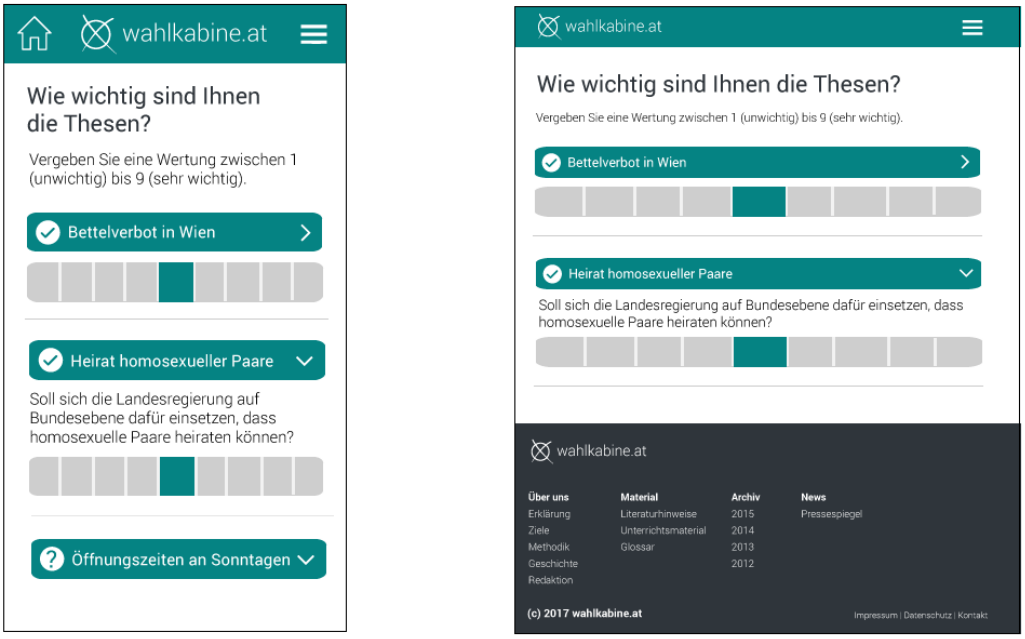


Figure 5.9: Mockups for the ranking screen for mobile and desktop.

showing the answer a user has given to a particular issue, provides an overview, reducing users cognitive load and foster Nielsen's "recognition rather than recall" [91]. These icons match the ones used on the answer buttons on the question screen in order to provide some kind of consistency, which also complies with one of Nielsen's usability heuristics [91]. The question name is already displayed on the question screen.

Below, there is a weight bar where users indicate how important they consider the political issue. As stated in Section 5.2 about requirements, the user can choose an importance level from a nine-step scale. Five, the average value, is preselected for all the questions, thereby weighting all questions equally if not selected differently by the user.

5.6.4 Result Screen

After having answered and ranked all questions, the user is forwarded to the result screen. Figure 5.10 presents them, as for the previous mockups, on mobile and desktop devices.

At the top, explanatory text summarizes the process. The party with the most agreement is written in bold letters. Below, there is a list of all political parties having provided their statements to the questions sorted in descending order of accordance, displaying the party with which the user shares the least agreement at the very bottom of the list. There are positive and negative values depending on whether the user agreed or disagreed on more statements and depending on the weighting allocated by the user. Next to the

bars is a value representing the length of the bar, which is not a percentage value, but an absolute number. Clicking on these bars navigates users to the parties' websites.

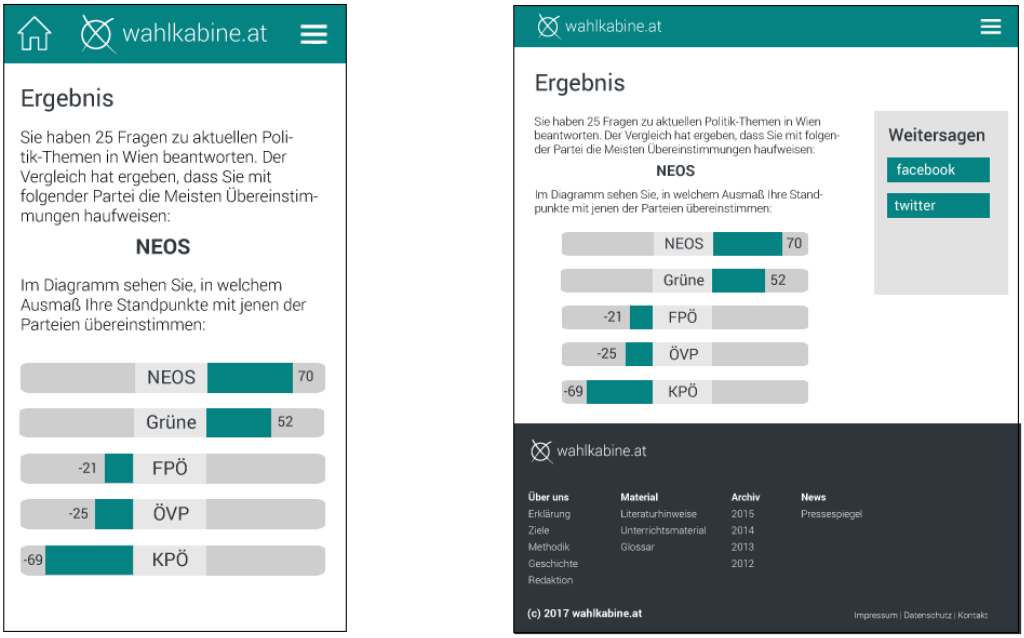


Figure 5.10: Mockups for the result screen for mobile and desktop.

Sharing results on social media platforms is another integral functionality of the new platform, as this yields higher user rates. In order to facilitate the sharing process buttons for the prevalent social media platforms are placed prominently on the right side of the screen. On mobile devices these buttons are placed below the result chart.

Moreover, after answering all questions, it is possible to compare one's responses with the answers of the politcal parties. A link to this subpage is provided below the social media buttons. There is also the possibility to redo the questionnaire.

Before transforming these mockups into actual code, a group usability testing session took place in order to gain feedback from the Institute for New Culture Technologies. This process is described in the following section.

5.7 Mockup Evaluation

After finalizing the mockups, feedback was collected in form of group usability testing with the members of the Institute for New Culture Technologies. As they have a very diverse background from User Interface (UI) Design, Journalism, Data Privacy, Public Relations and Marketing, insights from many different areas could be brought into the

conversation. After identifying usability problems, possible solutions were worked out together.

5.7.1 Procedure

As it is easier and cheaper to identify usability problems in earlier phases [116] a group usability testing approach was chosen. This was proposed by Downey (2007) [36] and was conducted after the mockups finalization. The evaluation procedure consists of the following steps, which was conducted with eight participants from interdisciplinary backgrounds.

(1) Introduction

After explaining the mockup evaluation process, the goal for the analysis is presented, which is to detect as many usability issues as possible to prevent usability issues before implementation.

(2) Exploration

Afterwards the mockups, as presented in the previous section, are presented one after the other to the participants to familiarize them with the new user interface. Feedback is already appreciated at this step and participants are encouraged to comment on encountered usability issues.

(3) Walkthrough

The next step is a walkthrough through the application from start screen to result screen in chronological order. At each screen participants are encouraged to comment on usability issues and possible errors or problems. On some issues vivid discussions are encountered as participants with different backgrounds have different experiences and opinions.

(4) Summary

As notes are taken throughout the procedure a list of usability issues emerges at the end of the evaluation process. The evaluation process is followed by a discussion about how to overcome these usability problems. Additionally, ways to further improve the application are defined together. In the end a document emerges which is sent to all the participants and is the basis of improvements of the mockups.

5.7.2 Results

Resulting from the evaluation process the following eight usability issues were detected: The home icon in the header, visualized as a house icon, is considered to be redundant. This was only visible on the mobile version of the mockup. Instead clicking on the

wahlkabine.at logo on the top left of the screen was seen as sufficient, as it is a de-facto standard these days that clicking on a logo leads to the start page.

Talking about the question screen, the progress bar with numbers from 1 to 25 was considered as gratuitous. Participants argued that usually nobody wants to go through questions more than once and even going back on question is usually the exception. Moreover, the numbering does not convey any information about the question and make it difficult for users to find a particular question anyway as they might not remember a question's number.

They are also remarks on the glossary popups including the suggestion to provide more intuitive ways to close them. This includes, on the one hand, a close icon on the upper right corner of the popover dialog. On the other hand, clicking on the background should also lead to closing the popover; thus, behaving like a modal dialog. In order to indicate this behavior the screen should be enhanced by an unobtrusive overlay.

Participants were also discordant what the framed capital letters inside the buttons stand for. After an explanation that these should indicate keyboard shortcuts, GUT participants recommended to think about a different way of visualization. In order to not distract users too much during the process, one remark was also to remove the footer on question and ranking screens. Another outcome was that the textual result should be further emphasized to stand out from remaining text on the page. The agreed upon solution was to increase font size. Figure 5.11 shows the improved question screen after the mockup evaluation session.

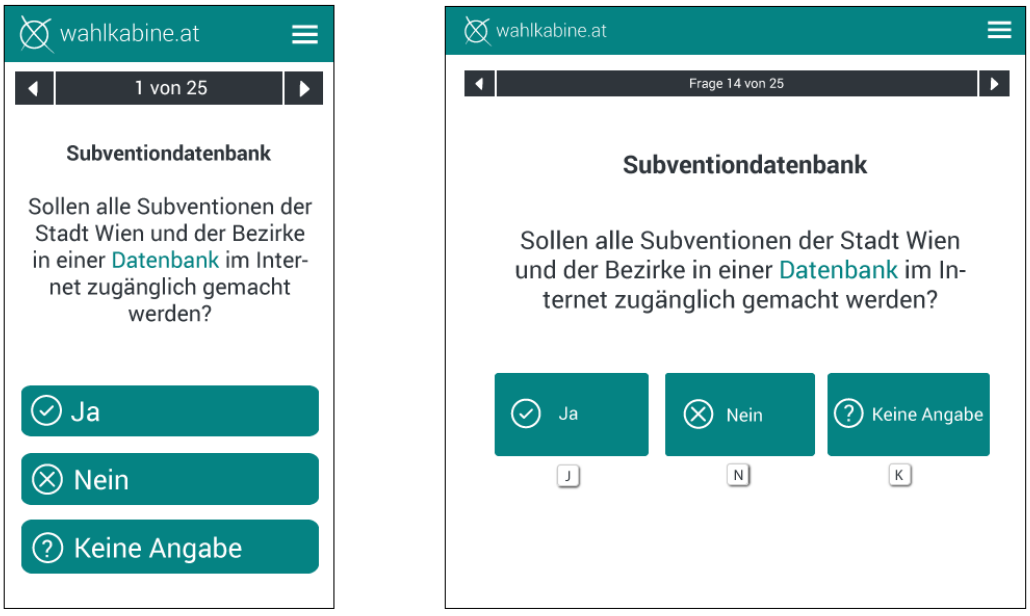


Figure 5.11: Question screen after mockup evaluation.

5.8 Implementation

After the UI design was finished and some improvements were made resulting from the design evaluation, the implementation of the platform took place.

Many different considerations were taken into account when deciding on a software stack for the new platform. An important goal was to supply as many devices as possible while maintaining only a single codebase [18]. For this reason native applications were not further considered [28]. A completely static page is not sufficient as the team behind *wahlkabine.at* does need to make occasional modifications [28]. This means a dynamic yet lightweight and high-performance solution is sought-after. In the end the MongoDB, Express.js, Angular and Node.js (MEAN) stack is chosen which consists of the following components:

MongoDB leading open source NoSQL Database [62]

Express.js web framework running on Node.js

Angular front-end framework for Single Page Applications (SPAs)

Node.js event-drive web server based on JavaScript

By using Angular it is possible to build the new platform as a SPA. A well-known benefit of this kind of applications is that all page resources are fully loaded after the initial request and page components are dynamically replaced, as the user navigates through the page [67]. In case of *wahlkabine.at* this is especially beneficial to overcome network loss, which can occur when using websites on public transport.

All in all, it is also a good choice for maintainability, as a software developer maintaining the applications only needs to know one programming language to modify front-end and back-end of the application [62]. In his thesis, Andreas Taranetz [122] conducted a detailed analysis on why this software stack was the optimal choice for this case.

5.8.1 Adaptability

To ensure support on most devices, despite their technical capabilities, a progressive enhancement approach is chosen [55][95]. This means starting with a very basic prototype and adding more web techniques in a layered fashion on top of this [42]. By creating a platform made of plain HTML first, the platform functions on almost every device with a functioning browser [55]. This should help that the application is already usable without CSS and avoid later modifications for screen reader compatability. When using a button element as button, we avoid the necessity to manually add Accessible Rich Internet Applications (ARIA)-role attributes on modified elements. Afterwards, it is coated with a layer of CSS, which means design, fonts and colors are added. HTML and

CSS are both designed in a forward compatible way which means code that is written nowadays will still work in the coming years, although the environment, which the page is accessed from, changes [55]. At last it was completed by adding a JavaScript layer in form of Angular components for each page.

In order to give further consideration to adaptability, the front-end was developed in a responsive way. Instead of giving elements fixed width and heights, flexible values are chosen. Also different graphics for different screen sizes are loaded using media queries [101]. Device appropriate images help that only the resolution absolutely necessary is loaded which should also improve performance [9]. In general, media queries were used a lot in order to use the available screen size as efficient as possible. Therefore, four different break points were created for mobile devices, big mobile devices, tablets and desktop devices.

5.8.2 Maintainability

Generally a clean code approach was pursued, following standard naming conventions, reuse of code and separation of code layers. For every subpage a separate Angular component is created. Moreover, certain sections appearing on more than one page were sourced out in separate components in order to avoid duplicate code.

In order to keep CSS maintainable, Sassy CSS (SCSS) was used as the style sheet language. By doing so variables and mixins can be created avoiding duplicate code. CSS code was written using the Block Element Modifier (BEM) methodology, which was created for long-term maintainability, codebase reduction and reusability of components [1].

5.8.3 Performance

In terms of performance, an effort was made not only to keep the codebase as small as possible by using the above mentioned SCSS, the compression and minification of front-end code was done. Especially graphics are often responsible for the largest part of a page's size. Therefore, many efforts are made to use as few images as possible and to keep them small. There are three different sizes for each image to reduce page load on devices with lower resolution. To reduce the number of HTTP request, HTML and JavaScript were compressed and CSS code inlined. Whenever possible images were built as Scalable Vector Graphicss (SVGs), minified, encoded to base64 and inlined as well. To further boost performance a caching mechanism is used.

5.8.4 Accessibility

Accessibility was always kept in mind while developing. The general rule was that function is the first priority and aesthetics is there to support it [123]. Special efforts were made that the application is usable without a mouse. To make text perceivable it was added on top of graphics using CSS, instead of writing it into graphics. ARIA

attributes were used to indicate areas that are changing, and augment HTML elements with more semantic meaning to facilitate screen reader usage.

5.8.5 Findability

From research in Chapter 3 we learned that it was difficult to find the questionnaire on some VAA websites; most people visit a VAA to conduct this questionnaire. To ensure findability on the new platform the start button of the *wahlkabine.at* questionnaire was spotlighted in the center of the landing page using a bright color. In order to improve external findability and support sharing on social platforms, meta tags are used; these are adapted depending on the subpage, which is not a standard functionality of SPAs.

5.8.6 Usability

Nielsen's ten heuristics [91] were always kept in mind throughout the development process. One of the major changes is that the questionnaire does not open in a separate window, which is a big improvement compared to the previous platform. To continuously evaluate the platforms usability, several usability testings were performed which are described in the following sections.

5.9 Group Usability Testing 1

After having a first functional version of the new platform, group usability testing with the Institute for new Culture Technologies was conducted. Different people were invited in order to gain as much valuable feedback as possible; including a User Experience (UX) expert, a system administrator, a privacy expert, two marketing experts, the project manager for this years' election questionnaire and the director of the institute.

5.9.1 Procedure

The evaluation procedure involved the following steps:

(1) Introduction

The *wahlkabine.at* platform is presented as well as the aim detecting usability problems as possible. It is stressed that there is no such thing as a wrong answer and they should just comment on issues they feel should be improved.

(2) Walkthrough

Afterwards the whole group performs the predefined task together. This task includes accessing the new platform, finding this years' questionnaire, answering all the questions and comment on the result. Each participant does this simultaneously, but independently, on his or her brought devices. Both desktop and mobile devices, different operating

systems and browsers are tested during this session. While clicking through the platform, they are encouraged to discuss about usability issues and possible errors or problems.

(3) Summary

During the process notes are taken. After all the participants conclude the task, a group discussion is held. Possible solutions to the problems are discussed and a list of improvements emerges. This list is sent to all the participants in order to rely on it during the next meeting.

5.9.2 Results

As a result of the evaluation the following ten issues were pointed out:

Starting with a general remark, participants had different opinions on the text alignment. Some of them are irritated by justified text, others prefer it. According to Trollip and Sales [124] justfied text accounts for 10% longer reading time, especially due to uneven spaces between words. For this reason justified text is only used on larger devices as whitespace is better balanced due to more available screen space.

Figure 5.12: Color comparison of wahlkabine.at and Österreichische Volkspartei (ÖVP)

Besides text alignment, a change of color was necessary. The Austrian People's Party (German: ÖVP) changed their primary color from black to turquoise [121] in the course of this work. Thus, it was necessary for *wahlkabine.at* to change its color scheme as well because the platform needs to remain 5.5.1 its neutral appearance and avoid influencing users towards a political party. Figure 5.12 illustrates the color similarity of *wahlkabine.at* and ÖVP. Screens with poor color calibration distort color representation and might therefore display the colors similarly. Although not only brightness, but also hue differed in seven degrees, the Institute for New Culture Technologies requested a color change. This is why the color scheme was switched from turquoise to orange. Thankfully, this change was easy due to SCSS variables.

Question titles endanger a question's neutrality on the question screen, was another outcome of the evaluation. By putting a question title on top of the actual question, people might be tempted to only read the question title instead of the entire question in order to save time. As some questions are formulated using the negated form, this would lead to questions being answered against a user's intention.

Removing parties' logos was another result from the focus group. Some of the participants perceived certain graphics to be more eye-catching than others due to bright colors or vastly varying aspect ratios. Therefore, the logos are dropped entirely and replaced by text to avoid favoring of certain parties. Moreover, some electoral lists might not even have a visual identity and would be disadvantaged in the first place.

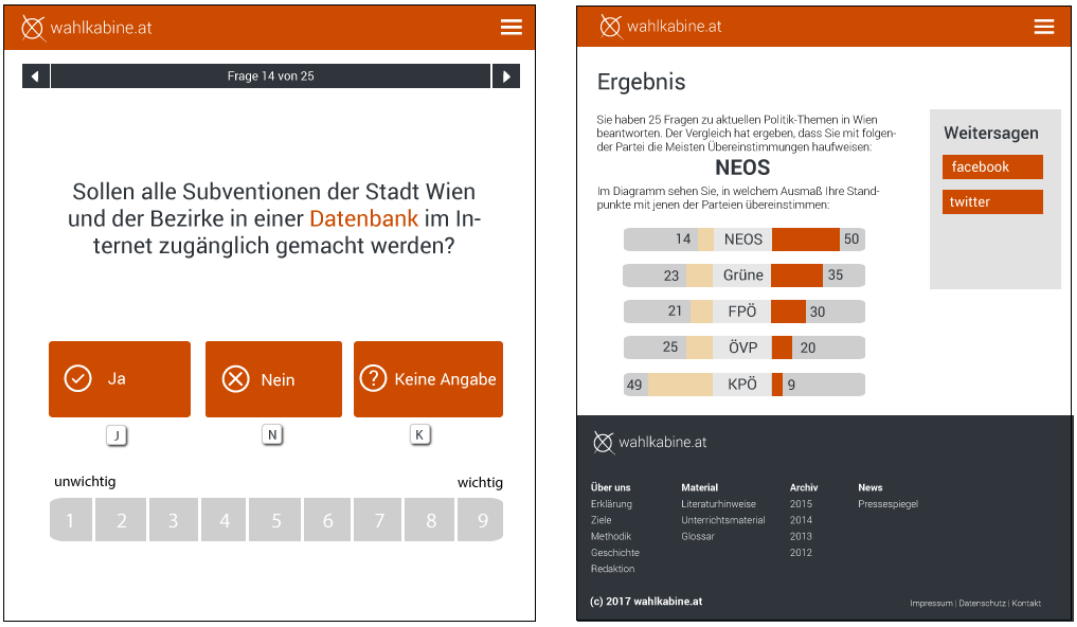


Figure 5.13: Question and results screens after the first evaluation.

The focus group agreed upon that there was still too much noise on the question screen and an even more minimalistic approach is the way to go. Therefore, the menu is removed as it does not provide any particular user benefit at this point as there is no help functionality for filling out the questionnaire.

A feature proposed by a usability expert was to add a visual progress bar in addition to writing the progress in numbers at the top of the question screen. Huotari [63] argues this helps "invoking progress-related psychological biases" and is therefore a good tool to enhance gamification of the application.

Instead of answering questions and rating importance on separate screens the usability expert claims it is better to consolidate these functions on one screen. He argues that it

is essential to avoid that users have to read each question twice. Although we argued in Section 5.6 that it is better put this step on a separate screen, the Institute for New Culture Technologies requested the change; they have the upper hand on such decisions.

Before showing the result, it was suggested to add another indication that the application's outcome is not actual the party they should vote for; many other factors need to be considered when deciding for a party. This indication was already prominently placed on the landing page. However, a participant argued that it needs to be added in the questionnaire process as well, to avoid prosecution for any reasons. Only after the user accepts this advice, he or she is navigated to the result page.

On the result page the focus group agreed upon adjustment of the result bars. The result for each party was one bar summarzing values for agreement and disagreement. As this removes information from the result it was proposed to display it as two distinct numbers and still sort it in descending order of the difference. Moreover, in order to distinguish positive and negative values better different shades of the same color should be used for positive and negative bars.

Figure 5.13 shows the page amended by the evaluation results of this section.

5.10 Group Usability Testing 2

After having collected and implemented the evaluation results from the Institute for New Culture Technologies, a second round of group usability testing was conducted with a group of users.

5.10.1 Procedure

In the course of the evaluation a second GUT [36] session is conducted with a total number of ten users. The users are from different age groups with varying technological background.

(1) Introduction

The *wahlkabine.at* platform was presented as well as the aim of the evaluation, which is the detection of usability problems. It is stressed that there are no wrong answers; they are encouraged to freely comment on what struck them as odd.

(2) Task Explanation

Afterwards the task is explained to them. They have to access the *wahlkabine.at* platform and to conduct the questionnaire on their own device. After finishing the platform they have to interpret the result of the questionnaire.

(3) Walkthrough

The users perform the predefined task. While doing so, they are encouraged to talk about usability issues and possible errors they encounter.

(4) Summary

While users are performing the task, notes are taken. Together with the users solutions are discussed to overcome the discovered usability problems. As with the previous evaluations at the end of the evaluation process a list of usability problems and possible solutions emerges.

5.10.2 Result

The 15 usability issues are categorized in specific comments about question and result screens, as well as general remarks on colors, menu and structure.

General Remarks

Generally the responses were quite positive. One of the participants stated that to him it seems like the previous application. After showing him the old version he admitted that it is completely different. This is exactly the kind of effect we wanted to achieve: the user should feel familiar with the platform, yet it should provide improved user experience.

What was confusing for some users was the distinction between welcome page and the sub pages, as both have similar structure and the same header with a call to action to start the questionnaire.

Some people tried to click the icons on the subpages, which is not possible. Making them clickable was the decision, that emerged to overcome this problem. However, the same link was used as the link description because otherwise these links would be listed twice on screen readers.

Concerning the menu, several people stated they do not like the menu's orange background color as they considered it to be too penetrative, which is why it was replaced by a dark grey background. The menu header was changed for the same reason. This was a relict of the color change from turquoise to orange; as turquoise is not equally penetrative, it works well as color for the navigation in contrast to orange.

Some users felt a bit lost on the website when they were asked to navigate to a certain section. As the menu was at the top of the website and not scrolling with the user; they had to navigate to the top of the website to change to a different subpage. A user suggested introducing a sticky menu to overcome this problem. Denney [30] did some research on this kind of menus and found out that they do not only fasten the navigation, but also subconsciously please the user without them being able to tell why.

Besides general feedback test users also had some remarks on the particular application screens.

Question Screen

Users were irritated that the question switched immediately when answering a question without any indication which answer was chosen. Presumably the application was simply too fast. An artificial delay had to be introduced in order for the user to grasp what was happening. More precisely, after 200ms the question was fading out, giving the user enough time to see the selected answer. The question is changed before it slowly fades in again. This effect is important; without the timeout inattentive users fail to notice that an exchange occurred when they were focused on the answer buttons.

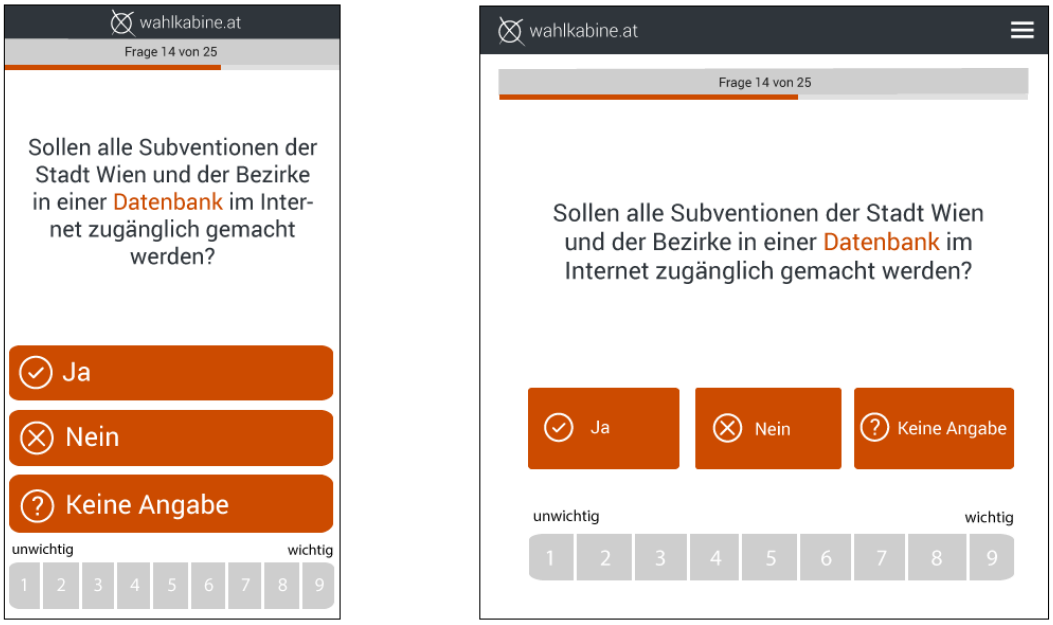


Figure 5.14: Question screen after the second group usability testing.

Some users are also confused that after selecting "Yes", "No" or "No opinion", they were not instantly navigated to the next question. It seemed not obvious to weight the questions. In order to make this more obvious, the weight buttons labeled from one to nine are fading from grey to orange in order to indicate a call-to-action. This change of focus suggests that an action needs to be taken. This is also in line with the color concept on highlighting user interface elements that can be clicked.

Keyboard shortcuts were shown using small indicators looking like buttons of a keyboard, visualized below the respective answer buttons on large devices. However, as people did not understand what they were for, they were removed to simplify the design even further. Moreover, the button size were increased to better use the available space.

Another feedback concerning the buttons was, that after an answer was chosen and the question was changed, buttons sometimes kept the selected color. Later on it was

discovered that this resulted due to the fact that on mobile devices hover and focus are used interchangeably. As setting a hover color led to more problems than benefits hovering color effects were removed from the questionnaire.

Some participants used Firefox during the group usability test. As it turned out there were some issues, as inlined CSS icons were not displayed. Although these icons never particularly transmitted additional information, it is important that it looks similar across devices.

Figure 5.14 illustrates the question screen after the second evaluation. Further efforts were made to increase the size of the user interface elements even further. For this reason, the progress bar on smaller devices is shrunk considerably.

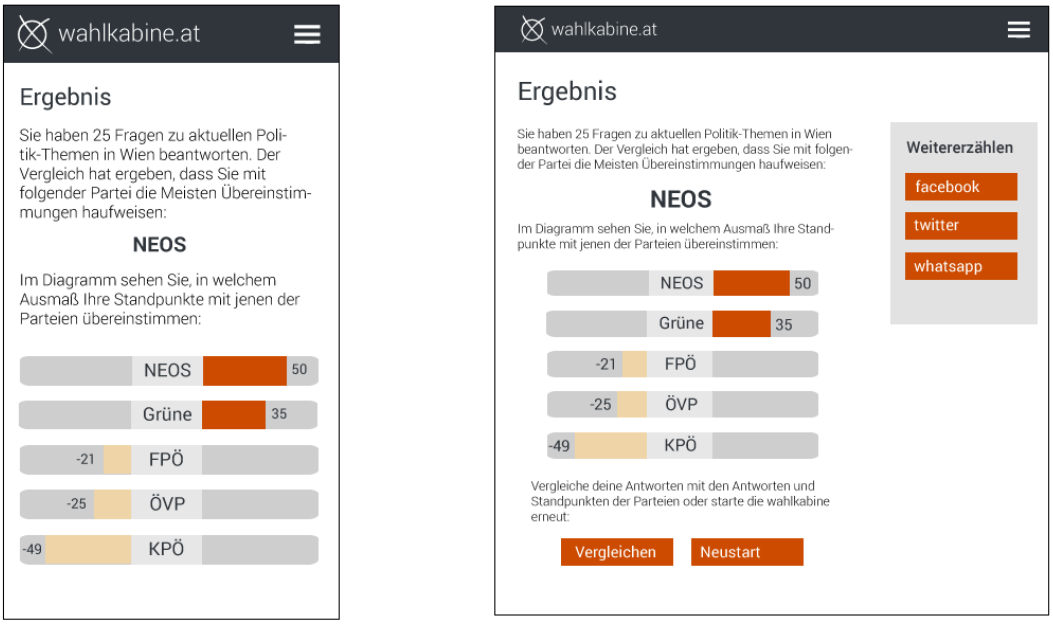


Figure 5.15: Result screen after the second group usabilty testing.

Result Screen

Evaluating the result screen, users had troubles understanding what positive and negative bars of the result meant. They stated that it is more cognitive load to calculate the difference of these two numbers themselves. A summarized value is preferred, which is why it was switched back to the previous implementation. Different colors for positive and negative result bars were perceived as a good way to emphasize a difference. Another piece of advice was to emphasize the party with most agreement more by displaying it in an even bigger and bolder way.

The sharing sidebar was supplemented by another button in order to share the result directly on WhatsApp as well.

Some of the users were confused by the result values; they were unsure what it meant to score 50 with a certain party. However, as this is part of the underlying algorithm and not of the visualization per se, this feedback was discarded for this thesis, but was passed on as to the Institute for New Culture Technologies that is responsible for the VAA's methodology.

Figure 5.15 shows the result screen amended by the issues that came up in this evaluation phase.

5.11 Final Result

Figure 5.16: Welcome page of the final application.

Having rectified usability problems that emerged from the evaluations, the final application was deployed timely on the 24 of August, 2017.

Figure 5.16 illustrates the final welcome page of the new Austrian VAA. The button to conduct the questionnaire is sticking out intentionally; this is the function most users visit the platform for.

When clicking on this button the user is directly navigated to the most current questionnaire, as displayed in Figure 5.17. The application is fully responsive serving desktop and mobile devices equally. Even existing smartwatches with a working browser can operate the application.

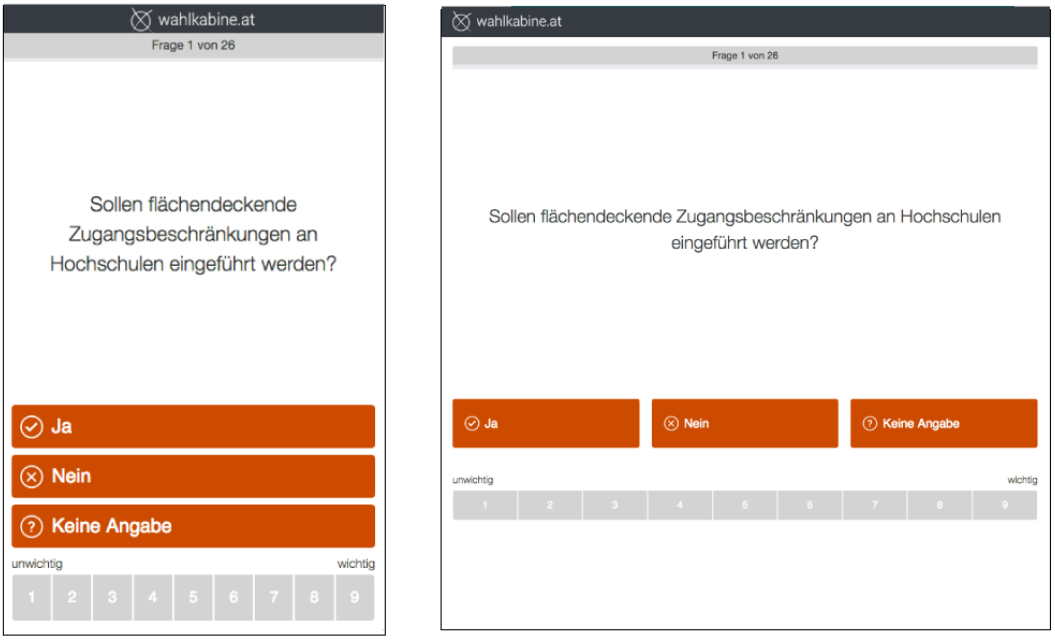


Figure 5.17: Question screen of the final application.

Since its deployment more than 900,000 people (as at 8th of October) conducted the questionnaire. Especially in the first couple of days the page gained a lot of media attention as journalists and newspapers reported about the platform's relaunch [32][33]. Despite this high user amount no usability problem was reported, which can be seen as a good sign and shows that the group usability testing of the platform turned out to find a great part of the relevant usability problems.

After successful deployment of the application only one last step in the process remains, namely evaluation of the platform. Chapter 6 completes the development process of *wahlkabine.at* by providing an evaluation against the prior defined criteria of future-friendly user interfaces.

CHAPTER 6

Evaluation

The previous chapter dealt with the software development process from requirement engineering to deployment of the final application. This chapter focuses on evaluation of the implemented platform against in Chapter 4 defined criteria. We will go through these criteria one after the other listing benefits and drawbacks of the solution before summarizing outcomes at the end of this chapter.

6.1 Adaptability

In order to make *wahlkabine.at* adaptive to different screen sizes and resolutions, a responsive design approach was applied. Media queries were used to use available space in an optimal way. The previous *wahlkabine.at* implementation was not responsive at all. Thus, the new application is an improvement. To assess the application's adaptability we decided to verify lower and upper bounds of the user interface's responsiveness.

6.1.1 Lower Bound

The left screen in Figure 6.1 shows the application still works on screens with a resolution of 160 x 230 pixels. Although not the entire question can be displayed on such a screen, the question container is scrollable to read the entire question. At this dimension the application is certainly not user-friendly to its full extend anymore. The nine weight buttons, which are displayed next to each other, are very narrow and it might be challenging to hit the intended one. Moreover, answer buttons are only distinguished using small icons. Also, questions with a higher word count include a lot of scrolling at these dimensions.

Only from a screen width greater than 280 pixels the answer button icons are complemented by a description, as can be seen in Figure 6.1. In order to make it work on

smaller more quadratic screens, buttons are placed next to each other, instead of on top of each other.

Figure 6.1: Question screen at different resolutions.

The bottleneck is certainly screen height. As answer buttons and weight buttons together have a fixed minimum height of 125 pixels, screen height needs to be at least 230 pixels in order to ensure enough space for the answer buttons. As questions' text lengths differ, a scroll bar helps to be able to read the entire questions. An effort could be made to support it on even smaller devices. This could be achieved by putting answer and weight buttons on separate screens. However, as screen size and resolution decrease it is also continually more difficult to perceive text and push buttons. Thus, a smaller implementation does presumably not make sense.

Moreover, there is a trend towards higher screen resolutions. In 2014, smartwatch screen resolutions, as analyzed by Cha et al. [22], had already a resolution of 320 x 320 pixels, which is supported by our implementation. The second screenshot in Figure 6.1 shows the user interface at these dimensions. Apart from that, most smartwatches these days do not have a browser yet. However, there are already smartwatch browsers out there [34].

6.1.2 Upper Bound

The lower bound was quite easily estimated, the upper bound is more difficult to define. Technically, there is no real upper bound; the questionnaire as well as the entire platform has a maximum content width of 960px. On larger screens the questionnaire will simply keep this width and remain centered. Figure 6.2 illustrates this behavior taking a screen width of 2000 pixels as an example.

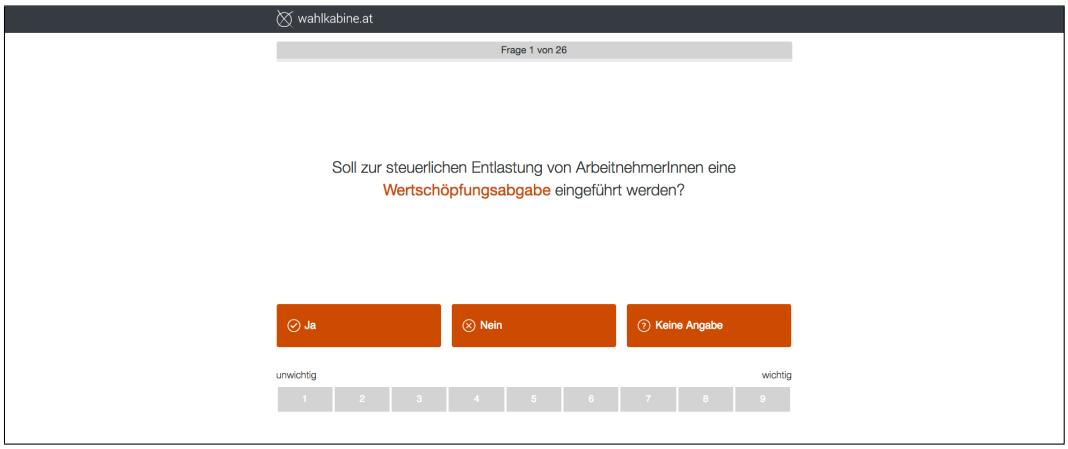


Figure 6.2: Question screen at 2000px screen width.

Although obviously the available space could be used in a better way, mouse distances are reduced, as everything is already perceivable. Compared to the old platform, where the desktop mode was a stretched version of a mobile application, this is definitely an improvement.

Concerning adapatability in term of input devices, *wahlkabine.at* can be used with touch devices, mouse and keyboard input, which is also further analyzed in the next section.

6.2 Accessibility

In order to identify and correct accessibility errors, previous and new platform were evaluated using the Web Accessibility Evaluation (WAVE) tool [132]. Afterwards accessibility was formally evaluated through testing the platform against WCAG 2.0. The key learnings are summarized in the following sections on the primary principles of WCAG 2.0. The whole evaluation list can be found in Appendix B.

6.2.1 Perceivable

This principle states that a website needs to be presented in a way that it can be well perceived by all its users. The platform performed very well at this first principle, achieving an AA rating, which is mainly owed to the platform's simplicity. The VAA

does neither include time-based media nor user input forms, which is why a lot of the guidelines are not applicable for *wahlkabine.at*. There is not much non-textual content either; thus, providing text alternatives is trivial. Due to progressive enhancement, where developers are encouraged to create semantic HTML, usage by assistive technology is naturally supported.

An effort was made to achieve the required color contrast ratio of 4.5:1, which is quite trivial for ordinary black text on white background. However, due to the incorporation of colors to highlight certain components, color brightness had to be reduced drastically to fulfill this success criterion. Figure 6.3 shows the color contrast evaluation of the final primary color using the WebAIM color contrast checker [21].

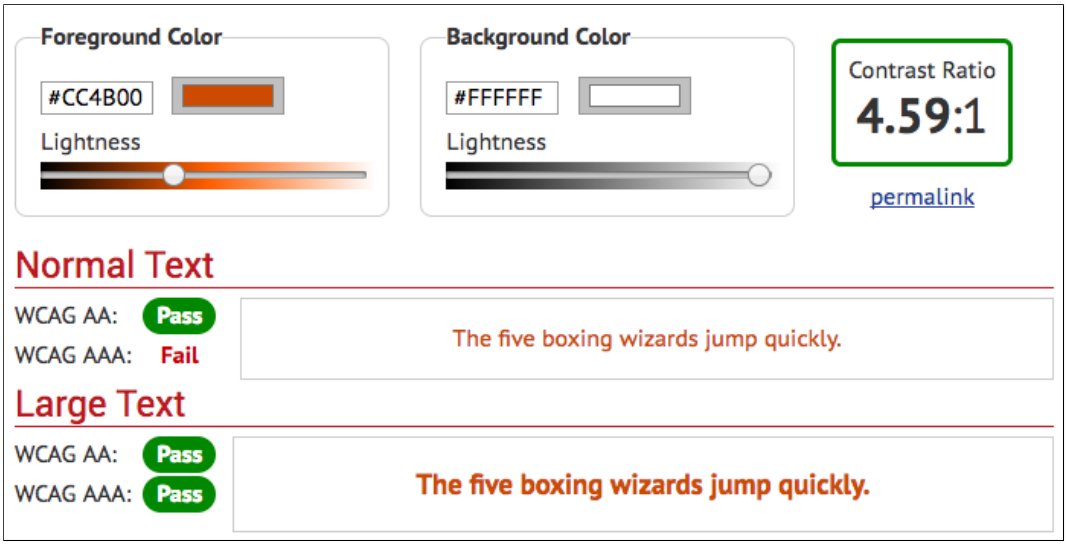


Figure 6.3: Color contrast evaluation of the primary color.

What did impede the AAA rating in the end was success criterion "1.4.8 Visual Representation". Although text is not fully justified on mobile devices, it is on desktop devices. Moreover, although the color scheme provides sufficient contrast and changing colors might be possible through browser functionality, colors cannot be directly changed on the platform.

6.2.2 Operable

This principle ensures that a website can be used by its users and avoids operations that users cannot perform. Effort have been made that the questionnaire is also operable using both keyboard navigation and keyboard shortcuts. Content that could cause seizures is avoided.

In a first version of the platform the visible focus was disabled due to aesthetic reasons. This was later revoked for the benefit of accessibility and usability. This spotlights once again the ongoing conflict between aesthetics and accessibility.

Naming and descriptions were created to be very easily understandable. To avoid redundant information some descriptions are only visible to screen readers for aesthetic purposes and to not impair reading fluency.

Although one could argue that further possibilities to navigate through the content could be offered, like a sitemap for example, all the success criteria were fulfilled for this principle.

6.2.3 Understandable

Understandability refers one the one hand to the page's content. On the other hand it examines the functionality and how predictable possible operations are. There is no form on the platform; thus, a lot of the success criteria were not applicable. In order that a screen reader knows in which language to read, the page language was defined. Complicated words are described in a glossary, which is to be found on a separate page. Link descriptions, names and navigation were created in an easily understandable and consistent way. Although one could argue that content can always be expressed in a simpler way, the requirements of this principle were fulfilled.

6.2.4 Robust

This guideline helps maximizing compatibility with users' current und future tools. HTML validator [139] and CSS validator [138] checks were carried out to ensure valid usage. Besides some Angular specific attributes, that were not recognized by the validator, the platform passed the test. Considering markup accessibility aria-attributes were used to enhance meaning of custom written markup.

Summarizing the WCAG accessibility check, the platform complies with WCAG 2.0 level AA. Improvements could be made, removing justified text alignment and the missing feature to change background and foreground colors. The whole list on th criteria assessed can be found in Appendix B.

6.3 Findability

Most important features should be findable in an intuitive way. On *wahlkabine.at* the most relevant functionality is the questionnaire a user can use to learn about his political preferences. Figure 6.4, shows the applications welcome page undergoing the so-called blur test. This examination shows that functionality is still perceivable with visual impairment [16]. Moreover, this helps to visualize high contrast areas in a user interface [144].

Also the other pages were undergone by this kind of evaluation. The performance of the platform is very satisfactory. Color was only sparely used to set focus on clickable areas. This approach shows its benefits.

Figure 6.4: Blur test highlights high contrast areas of the landing page.

Concerning external findability a SEO checkup [117] was conducted with old and new website and showed that the overall SEO score improved considerably from 59/100 of the old website to 85/100 on the new platform. Besides verifying usage of meta tags, this validator also checked correct usage of headings and performance indicators, as these also influence a page's search engine rank.

To further improve SEO, the platform suggests using even more key words in the meta description, as well as creating a sitemap.xml to help crawlers indexing faster. Another suggestion was to include Google Analytics. As one of the requirements was to not host analytics on foreign servers, this is not an option for *wahlkabine.at*.

6.4 Performance

Performance was also a critical aspect of the thesis. The application experiences extreme peak times, as soon as it is exposed on national media. As stated in the Chapter 4, Barker [9] introduced four key performance indicators: page load time, page render time, number of HTTP requests and page file size. To evaluate the platform a performance audit was conducted using a performance evaluation tool [99]. Table 6.1 contrasts these key performance indicators for old and new *wahlkabine.at* implementations.

At first sight one already notices that page file size of the new application is higher; the main reason for this is that the new application is implemented as a SPA. This means, in comparison to conventional web applications, where additional resources need to be requested when the user is navigating, SPAs load all page resources with the initial

	new application	old application
Page loaded	957 ms	1,560 ms
Page file size	1.1 MB	546 kB
Page rendered	3,620 ms	2,660 ms
HTTP Requests	16	56

Table 6.1: Performance comparison

request, therefore explaining the higher page load of the new platform. Google identified in a 2016 report [35] that 53% of all users would abandon a website that takes longer than three seconds to load; our implementation is clearly below this threshold. Figure 6.5 illustrates content breakdown by media type. Although images are responsible for the majority of HTTP requests, they only account for approximately a quarter of the page load. On the other hand, JavaScript amounts to one third of the page size split up into three files.

The number of HTTP requests is significantly smaller for the new application. This was achieved by consolidating source files and inlining as many SVGs as possible. Obviously this was not achievable for some graphics, because they can be changed by the administrator using a file upload in the back-end. However, compared to the previous system, where 56 HTTP requests were necessary on initial loading, the number has reduced considerably.

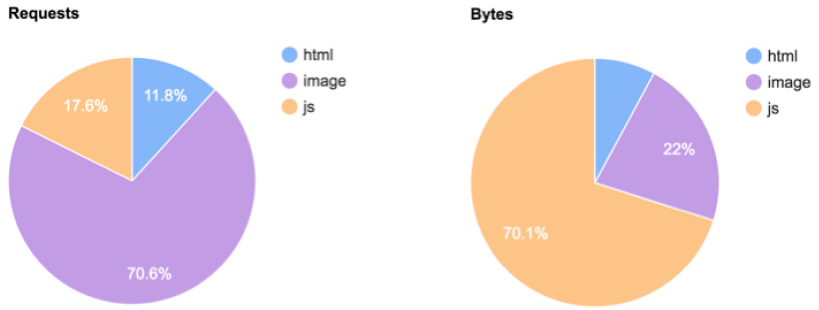


Figure 6.5: Content breakdown by media type.

See Andreas Taranetz's thesis [122] for a thorough anaylsis of the platform's performance.

6.5 Maintainability

The ISO standard on software quality [66] identifies four subcharacteristics of maintainabily: analyzability, changeability, stability and testability. To analyze maintainability of the new platform, these four criteria were looked into:

Talking about analyzability, the code was split, having separate HTML, CSS and JavaScript files for each component. This should facilitate identifying the line of code to adapt. Changeability was ensured through choosing a software stack based on one programming language. In order to be able to make modifications, a developer needs to know only Typescript. Stability is ensured as changes can be performed in the source code and deployed afterwards, which keeps the application stable. Lastly, in terms of testability, the underlying algorithm calculating a user's party preferences was tested extensively because it is of general interest that the algorithm performs the calculation in a neutral way without preferring a particular party. Besides that, test coverage is not as comprehensive as we want it to be.

What is more, a software stack with a dedicated community was chosen, thus making sure that the framework will be further improved in the years to come. How maintainable the new application actually is will turn out within the next months and years.

6.6 Usability

Several sessions of group usability testing (GUT) were conducted throughout the development process and a total number of 33 usability problems were identified. Feedback on mockups helped to prevent usability problems being implicated in the development phase. Further usability sessions, after the development process was finished, helped to improve the application even further. Further details on the carried out evaluations can be found in Section 5.7, Section 5.9 and Section 5.10

In the end more than 900,000 people (as at 8th of October) used the application and no single usability concern was reported, which shows that our usability evaluation found the most relevant usability issues.

6.7 Sustainability

This principle was already tested in the course of performance, usability and findability evaluations. The remaining subcharacteristic of sustainability is operation of sustainable components. In this regard, sustainable hosting would be a valid approach. However, the platform is hosted on a privately held server in Austria. The financial means to switch to sustainable hosting are restricted.

6.8 Conclusion

Overall, *wahlkabine.at* performed well on different criteria. However, there is still potential for improvement.

Accessibility evaluation showed that the application complies with WCAG 2.0 level AA. However, not many improvements would be necessary to receive an AAA rating. Concerning maintainability, the test coverage should be increased to better detect

inconsistencies after performing a maintainance operation. In order to evaluate usability, several usabiliy testings were conducted. One could argue that another formal usability analysis should be made after the three evaluation sessions' feedback was implemented in the application, in order to evaluate the final platform. In terms of adaptability further efforts could be made to support even smaller devices. However, as devices' resolutions gradually increase this might not be necessary.

All in all, our project partner and end-users were satisfied with the relaunch. We received a lot of informal feedback after the platform's launch, which showed that the application achieved its objective. Readers are encouraged to continue reading the final chapter, which discusses lessons learned and gives an outlook on future research.

CHAPTER 7

Conclusion

The aim of this thesis was to design and implement a future-friendly user interface of a VAA on the example of *wahlkabine.at*. After evaluating similar applications, a literature review on future-friendly user interfaces was conducted, which led to the identification of seven key principles of future-friendly user interfaces: adaptability, accessibility, maintainability, performance, usability and sustainability. They served as an important framework, under which the development of the platform took place. Throughout the development process, informal evaluations were conducted to improve the application along the way. Although some challenges occurred in the course of this work, like the moving up of the federal elections, the platform was launched on schedule in August 2017. The new application supports usage from different devices, ranging from smartwatches to television. Moreover, it is fully usable for people with special needs, as the platform complies with the WCAG 2.0 level AA. Empiric observation showed that the application is fulfilling current software quality standards. A formal evaluation was conducted and showed that the platform complies with the established principles and supports the idea that the application will work in the next couple of years.

Only time can tell, how future-friendly the platform actually is.

7.1 Lessons Learned

Many different aspects need to be considered when building a future-friendly system from scratch, which is why criteria were defined before implementation was carried out. Certain principles were found to be in conflict with each other. Interestingly enough, aesthetics often conflicted with accessibility, making it challenging to discard something aesthetically pleasing in favor of accessibility, as this was the approach agreed upon in the first place. Thorlacius [123] argues that aesthetics play an essential role on websites. However, functionality for all users should always outweigh aesthetics, which is not always an easy step for a designer after putting much effort into a piece of work. On the other

hand, Mayhew [85] argued user interface design is always a matter of compromise. Slowly a trend towards functionality over aesthetics is emerging, but it is still an important lesson to be learned by many designers. Most websites are still not even close to being accessible.

Similarly, another discovery was the conflict between customers and developers. Customers usually know their product best and have a clear mental model about the final version. Because *wahlkabine.at* has been around for such a long time, customers have certain expectations on how things are supposed to work, making the introduction of new ideas very difficult. Using scientific evidence does not always help to convince a customer of new approaches. One finding emerging from this project is that it is the most important goal to satisfy a customer's needs.

In addition, many lessons were learned in terms of communication. Sometimes it took several attempts to get a feature the way the customer wanted it, even tough one was sure having understood the goal of a feature before implementation. Our mistake in this instance was relying on the old platform. Under the premise that the application worked as intended by the customer or was correctly understood by the customer, this might have worked out. However, customers are not always well-versed in their application knowledge. Moreover, what they want and what they need often diverges. Therefore, requirements should be defined without looking too much at previous implementations in order to avoid transferring mistakes onto a new platform.

7.2 Research Directions

Future-friendliness is an emergent topic in computer science. In times where an effort is made to prolong the software lifecycle and rejuvenate applications [143], there is a lot of potential for future research on considering this aging process from the start.

This thesis was a first attempt to list criteria that help prolong the lifecycle of user interfaces from the start of implementation. However, the list of principles might not be complete.

There has been research on how to prolong this software aging process [23], and there might be many overlapping areas. However, future studies could investigate further how to prolong the software lifecycle before development. Software aging is a process that certainly cannot be prevented, but efforts can be made to prolong software usage as much as possible [3].

We hope this research will serve as a basis for future studies on future-friendly user interfaces. The prospect of being able to prolong the software lifecycle serves as a continuous incentive for future research.

List of Figures

3.1 Countries with national or transnational VAAs. 10
3.2 wahlkabine.at. Question and result screens of the old application. 12
3.3 wahl-o-mat. Question and result screens. 13
3.4 smartvote. Question and result screens. 15
3.5 StemWijzer. Question and result screens. 16
3.6 Kieskompas. Question and result screens. 17
3.7 Vote Compass. Question and result screens. 18

5.1 Initially, Turquoise was chosen to be the primary color. 39
5.2 Colors auf Austria's political landscape. 39
5.3 Political parties' colors mapped onto a color wheel. 40
5.4 Color Scheme of the new platform. 41
5.5 System fonts on different devices [135]. 42
5.6 Comparison of old and new wahlkabine.at logo. 43
5.7 Style tiles for wahlkabine.at 44
5.8 Mockups for the question screen for mobile and desktop. 46
5.9 Mockups for the ranking screen for mobile and desktop. 47
5.10 Mockups for the result screen for mobile and desktop. 48
5.11 Question screen after mockup evaluation. 50
5.12 Color comparison of wahlkabine.at and ÖVP 54
5.13 Question and results screens after the first evaluation. 55
5.14 Question screen after the second group usability testing. 58
5.15 Result screen after the second group usabilty testing. 59
5.16 Welcome page of the final application. 60
5.17 Question screen of the final application. 61

6.1 Question screen at different resolutions. 64
6.2 Question screen at 2000px screen width. 65
6.3 Color contrast evaluation of the primary color. 66
6.4 Blur test highlights high contrast areas of the landing page. 68
6.5 Content breakdown by media type. 69

List of Tables

6.1 Performance comparison 69

Appendix A

List of Voting Advice Applications

The following list summarizes VAAs, which were discovered in the course of this thesis. A detailed analysis on a subset of these can be found in Chapter 3.

Country	Name	Link
AT	wahlkabine	http://wahlkabine.at
AU	Vote Compass	https://votecompass.abc.net.au/home
BA	Glasometar	http://glasometar.ba
BE	WECitizens	http://vote.wecitizens.be/
BG	Glasovoditel	http://www.glasovoditel.eu
BR	I Side With	http://brasil.isidewith.com
CA	Vote Compass	https://votecompass.cbc.ca/federal/
CH	smartvote	https://www.smartvote.ch
CH	vimentis	https://www.vimentis.ch/wahlen/
CZ	volebnikalkulacka	https://volebnikalkulacka.cz
DE	plakos	http://parteien-test.plakos.at
DE	wahl-o-mat	http://wahl-o-mat.de
DK	Kend din kandidat	http://www.altinget.dk/kandidater/
ES	Brujula Electoral	https://brujulaelectoral.es
EU	EU Profiler	http://www.euprofiler.eu
FI	Voting Aid	http://votingaid.com/en/home/
FR	vote & vous	http://www.vote-et-vous.fr
GR	votematch.gr	http://www.votematch.gr
HU	vokskabin	http://www.vokskabin.hu/de/
HU	voksmonitor	http://voksmonitor.hu
IE	Which Candidate	http://www.whichcandidate.ie
IT	Vote Match	http://it.votematch.eu
LI	wahlhilfe.li	http://www.wahlhilfe.li/wahlhilfe/
LV	providus	http://providus.lv/page/pielaiko-partiju
NL	stemwijzer	http://www.stemwijzer.nl/
PE	GPS Electoral	http://gpselectoral.pe
PL	Latarnik Wyborczy	http://www.latarnikwyborczy.pl

Country	Name	Link
PT	Bussola Eleitoral	http://www.bussolaeleitoral.pt
SK	volebni kalkulacka	https://volebnikalkulacka.sk
TR	Oy pusulasi	http://www.oypusulasi.org
UK	The Political Compass	https://www.politicalcompass.org
UK	Who shall i vote for?	http://www.whoshallivotefor.com
US	electoral compass	http://www.electoralcompass.com

Appendix B

Accessibility Report

Guideline	Level	Yes	No	n.a.
1. Principle: Perceivable				
1.1. Text Alternatives				
1.1.1. Non-text Content				
There is a meaningful and equivalent alternative for all non-text content, such as images, graphics, objects, graphic controls in forms and hotspots in image maps.	A	x		
If the alternative text is not sufficient for the text alternative, a long description is prepared and is referred to in the alternative text.	A	x		
Decorative graphics or layout graphics have empty alt attributes or they are concealed from assistive technologies (e.g., screen readers) in some other way.	A	x		
There are no graphic CAPTCHAs or an alternative is present.	A	x		
1.2. Time-based Media: Provide alternatives for time-based media				
1.2.1. Audio-only and Video-only (Prerecorded)				
If audio or video media are not an alternative to the content, the following applies:				
There are text transcripts for prerecorded audio media.	A			x
There are text transcripts for prerecorded video media or	A			x
Prerecorded video media have text transcriptions or audio descriptions.	A			x
1.2.2. Captions (Prerecorded)				
Prerecorded video content has simultaneous subtitles.	A			x
1.2.3. Audio Description or Media Alternative (Prerecorded)				
Prerecorded audio media (e.g., podcasts) have written text transcripts.	A			x

Guideline	Level	Yes	No	n.a.
Prerecorded video media have written text descriptions or audio descriptions.	A			x
1.2.4. Captions (Live)				
Live audio media have simultaneous subtitles.	AA			x
1.2.5. Audio Description (Prerecorded)				
Prerecorded video media have audio descriptions of visual content that is not described in the standard audio description (SC 1.2.1).	AA			x
1.2.6. Sign Language (Prerecorded)				
Sign language videos are provided for all prerecorded audio content.	AAA			x
1.2.7. Extended Audio Description (Prerecorded)				
An extended audio description is prepared for video media if the pauses are not sufficient to communicate the meaning of the video content.	AAA			x
1.2.8. Media Alternative (Prerecorded)				
There is an alternative for all prerecorded video media, containing information that is spoken and/or shown simultaneously.	AAA			x
1.2.9. Audio-only (Live)				
An alternative is provided for all live audio media.	AAA			x
1.3. Adaptable: Create content that can be presented in different ways (for example simpler layout) without losing information or structure.				
1.3.1. Info and Relationship				
A. Headings				
Headings make the structure of the document clear.	A	x		
Headings are marked up using the heading element (h1, h2, ... , h6).	A	x		
B. Lists				
Listed information is formatted as a list (ul, ol, dl).	A	x		
C. Forms				
In forms with multiple parts, the parts are grouped by content into information blocks.	A			x
Labels and related form input fields are logically linked.	A			x
D. Data tables				
Data tables are formatted with the necessary markup, e.g., headings for columns; rows and tables are clearly labeled, and headings and summaries are present.	A	x		
Data tables can be read serially and are not used for layout purposes.	A	x		

Guideline	Level	Yes	No	n.a.
E. Use of symbols				
Special text is correctly formatted, e.g., citations with cite and long quotations with blockquote	A	x		
1.3.2. Meaningful Sequence				
The logical order is retained for screen readers and when CSS is turned off.	A	x		
Contents in tables are correctly linearized and no empty cells are used to create space in the layout.	A	x		
No character spaces are used to create space in the layout; CSS is used instead.	A	x		
There is no contextual confusion caused by content positioned with CSS.	A	x		
1.3.3. Sensory Characteristics				
There are no instructions that are solely optical or acoustic, e.g., "Press the green button on the left".	A	x		
1.4. Distinguishable: Make it easier for users to see and hear content including separating foreground from background.				
1.4.1. Use of Color				
Information is not communicated solely based on color.	A	x		
If color alone is used for differentiation, e.g., for links in a text, the links have a contrast ratio to the surrounding running text of at least 3:1.	A	x		
1.4.2. Audio Control				
If audio plays automatically for more than 3 seconds, a stop button is provided.	A			x
1.4.3. Contrast (Minimum)				
The contrast ratio of the font color to the background color is at least 4.5:1.	AA	x		
The contrast ratio of the font color of large fonts (at least 18 pt or 14 pt for bold text) to the background color is at least 3:1.	AA	x		
This applies to all text and tips, as well as to the borders around input fields and texts in information graphics. Does not necessarily apply to logos, logotypes or purely decorative graphics.	AA	x		
1.4.4. Resize Text				
The font size is defined in the CSS in terms of % or em.	AA	x		
It is possible to enlarge either the contents of the entire page or the text alone using the browser's "zoom" function.	AA	x		
1.4.5. Images of Text				

Guideline	Level	Yes	No	n.a.
Text is used instead of text graphics for content. Exceptions:				
The display size of content can be scaled, and the content can be read without CSS.	AA	x		
The content is necessary, such as a logo or brand name (e.g., if a particular graphical form is required). They can be described either with alt attributes or title attributes.	AA	x		
1.4.6. Contrast (Enhanced)				
The contrast ratio of the font color to the background color is at least 7:1.	AAA	x		
The contrast ratio of the font color of large fonts (at least 18 pt or at least 14 pt for bold text) to the background color is at least 4.5:1.	AAA	x		
This applies to all texts and tips, as well as to the borders around input fields and texts in information graphics but does not necessarily apply to logos, logotypes or purely decorative graphics.	AAA	x		
1.4.7. Low or No Background Audio				
Prerecorded speech has almost imperceptible background noise or none at all, or the audio can be switched off.	AAA			x
1.4.8. Visual Presentation				
For visual presentation of blocks of text, the following is possible:				
The user can select foreground and background colors.	AAA		x	
The width is not more than 80 characters.	AAA		x	
Text is not justified (it is either left-aligned or right-aligned).	AAA		x	
The line spacing is at least 1.5 within paragraphs and the paragraph spacing is at least 1.5 times as large as the line spacing.	AAA	x		
The text size can be scaled up to 200 percent without assisting technology (without scrolling).	AAA	x		
1.4.9. Images of Text (No Exception)				
Text is used instead of text graphics for content. Exceptions:				
The contents are essential for a piece of information that cannot be communicated without text graphics (e.g., logo).	AAA	x		
2. Principle: Operable				
2.1. Keyboard Accessible: Make all functionality available from a keyboard				
2.1.1. Keyboard				

Guideline	Level	Yes	No	n.a.
The following can be navigated and operated using the keyboard (tab key):				
All page functions and elements.	A	x		
All form input fields, controls and switches.	A	x		
No particular timing of individual keystrokes is needed for operation.	A	x		
2.1.2. No Keyboard Trap				
The keyboard focus is not blocked for any element of the website.	A	x		
The user can move focus to and from every element using the keyboard.	A	x		
The user is advised if keyboard keys other than the conventional ones are used (tab key, arrow keys).	A	x		
2.1.3. Keyboard (No Exception)				
All functionalities can be operated using the keyboard, without exception.	AAA	x		
No particular timing of individual keystrokes is necessary for operation.	AAA	x		
2.2. Enough Time: Provide users enough time to read and use content				
2.1.2 Timing Adjustable				
There is no time limit for pages. Exceptions:				
The user can turn off the time limit before encountering it.	A			x
The user can adjust the time limit before encountering it.	A			x
2.2.2. Pause, Stop, Hide				
The following applies to any auto-updating, moving or flashing information that starts automatically and is presented in parallel with other content for longer than 5 seconds:				
The user can use some mechanism to stop, close or hide the information.	A			x
A mechanism is provided for automatic updates, so that the user can stop or hide the update or control its frequency.	A			x
2.2.3. No Timing				
No time constraint is necessary when processing the content.	AAA	x		
2.2.4. Interruptions				

Guideline	Level	Yes	No	n.a.
Interruptions such as advisories can be postponed or suppressed by the user, except interruptions involving an emergency.	AAA			x
2.2.5. Re-Authenticating				
If an authenticated session is in progress, the user can continue an action after re-authenticating without loss of data.	AAA			x
2.3. Seizures: Do not design content in a way that is known to cause seizures				
2.3.1. Three Flashes or Below Threshold				
Websites contain nothing that flashes more than three times a second on an ongoing basis, or the flash is below a defined limit for flashes.	A	x		
2.3.2. Three Flashes				
Websites contain nothing that flashes more than three times per second on an ongoing basis.	AAA	x		
2.4. Navigable: Provide ways to help users navigate, find content and determine where they are				
2.4.1. Bypass Blocks				
Skip links are made available to avoid repeated blocks of information	A	x		
Repeated blocks of information are grouped or labeled using headings.	A	x		
2.4.2. Page Titled				
Web pages have a title (title tag in the meta area) that describes the topic or purpose.	A	x		
2.4.3. Focus Order				
The order of links in the navigation and in the content is logical.	A	x		
2.4.4. Link Purpose (In Context)				
Link texts can be understood either alone or based on the context.	A	x		
A change in format is indicated by the link text or the context.	A	x		
2.4.5. Multiple Ways				
In addition to navigation, the website presents at least one other method for accessing content:				
A search function or	AA		x	
A sitemap / table of contents or both	AA	x		
2.4.6. Headings and Labels				
Informative page headings and labels are used:				

Guideline	Level	Yes	No	n.a.
The website has headings that group the content.	AA	x		
The headings describe the subsequent section of content concisely and meaningfully.	AA	x		
Descriptive labels are present in forms.	AA			x
The functions or instructions are labeled, and active zones can be recognized on image maps and maps.	AA			x
2.4.7. Focus Visible				
Elements with focus are visibly emphasized when they are activated using the keyboard.	AA	x		
Skip links become visible when they receive keyboard focus.	AA			x
2.4.8. Location				
The current position of the user within the website or a process is indicated.	AAA	x		
2.4.9. Link Purpose (Link Only)				
The purpose of each link can be identified from the link text alone. Link texts describe the target or the purpose and change of format.	AAA	x		
2.4.10. Section Headings				
Headings introduce all sections of content.	AAA	x		
3. Principle: Understandable				
3.1. Readable: Make text content readable and understandable				
3.1.1. Language of Page				
Every web page has a correct language declaration.	A	x		
3.1.2. Language of Parts				
Sections of text in languages other than the default language are marked up using the lang attribute.	AA			x
Individual words in another language that could be understood incorrectly or not at all are marked up using the lang attribute.	AA			x
3.1.3. Unusual Words				
Unusual words, technical terms and foreign words are explained in a glossary or with some other mechanism.	AAA	x		
3.1.4. Abbreviations				
Abbreviations are explained in a glossary or	AAA	x		
Abbreviations are explained with some other mechanism.	AAA		x	
3.1.5. Reading Level				
Summaries or alternatives are made available for texts that are too complicated to be understood by people with a basic education (9 years of school).	AAA	x		

Guideline	Level	Yes	No	n.a.
3.1.6. Pronunciation				
A mechanism is provided to detect the pronunciation of content if this is necessary to distinguish between options.	AAA			x
3.2. Predictable: Make web pages appear and operate in predictable ways				
3.2.1. On Focus				
Context does not change when a section of the page receives focus.	A	x		
3.2.2. On Input				
Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised beforehand.	A			x
3.2.3. Consistent Navigation				
Navigation within a website is structured and arranged consistently.	AA	x		
3.2.4. Consistent Identification				
Elements with the same function are identified consistently within a website.	AA	x		
3.2.5. Change on Request				
The context is only changed upon receiving confirmation from the user.	AAA			x
3.3. Input Assistance: Help users avoid and correct mistakes				
3.3.1. Error Identification				
If input errors are automatically detected, the error is clearly described in text form in the error message.	A	x		
3.3.2. Labels or Instructions				
Labels or instructions are given when user inputs are required.	A	x		
3.3.3. Error Suggestion				
Suggested corrections are made in case of input errors.	AA			x
3.3.4. Error Prevention (Legal, Financial, Data)				
It must be possible to check, change, delete or confirm inputs that have legal or financial consequences before sending.	AA			x
3.3.5. Help				
Help is available when:				
Specific inputs must be entered in a form;	AAA			x
Inputs must be made in a particular format.	AAA			x
3.3.6. Error Prevention (All)				

Guideline	Level	Yes	No	n.a.
It must be possible to check, change, delete or confirm all inputs before sending.	AAA			x
4. Principle: Robust				
4.1. Compatible: Maximize compatibility with current and future user agents, including assistive technologies				
4.1.1. Parsing				
The markup language used, HTML or XHTML, conforms to standards and is free of errors.	A	x		
4.1.2. Name, Role, Value				
In case of generated and self-programmed content, markup is used in a way that supports accessibility.	A	x		

Acronyms

ÖH Österreichische Hochschülerschaft. 11

ÖVP Österreichische Volkspartei. 54, 55, 75

ARIA Accessible Rich Internet Applications. 53

BEM Block Element Modifier. 52

BZÖ Bündnis Zukunft Österreich. 40

CMS Content Management System. 30

CSS Cascading Style Sheets. 23, 24, 28, 29, 42, 44, 52, 53, 59, 67, 69

CW Cognitive Walkthrough. 32

FACE German Federal Agency for Civic Education. 13

FRANK Team Stronach. 40

GUT Group Usability Testing. 3, 33, 56

HTML Hypertext Markup Lanuage. 23, 24, 27, 28, 44, 51–53, 66, 67, 69

HTTP Hypertext Transfer Protocol. 29, 42, 52, 68, 69

IoT Internet of things. 23

KPÖ Communist Party of Austria. 40

MEAN MongoDB, Express.js, Angular and Node.js. 50

OS operating system. 22

SCSS Sassy CSS. 52, 55

SCW Streamlined Cognitive Walkthrough. 32

SEO Search Engine Optimization. 26, 67, 68

SPA Single Page Application. 51, 53, 68

SVG Scalable Vector Graphics. 52, 69

UCD User Centered Design. 3, 36

UI User Interface. 49, 50

URL Uniform Resource Locator. 29, 38

UX User Experience. 53

VAA Voting Advice Application. 1–3, 5–11, 14, 15, 17–20, 36–38, 40, 43, 51, 53, 60, 65, 73, 75

W3C World Wide Web Consortium. 24, 26, 27

WAI Web Accessibility Initiative. 24

WAVE Web Accessibility Evaluation. 65

WCAG Web Content Accessibility Guidelines. 24, 25, 40, 65, 67

WCED World Commission on Environment and Development. 30

WDR Westdeutscher Rundfunk. 10

WPO Web Performance Optimization. 31

Bibliography

[1] BEM — Block Element Modifier. http://getbem.com/. Accessed: 2017-09-24.

[2] Zuriani Hayati Abdullah, Jamaiah Yahaya, and Aziz Deraman. Towards anti-Ageing model for the evergreen software system. In *Proceedings - 5th International Conference on Electrical Engineering and Informatics: Bridging the Knowledge between Academic, Industry, and Community, ICEEI 2015*, pages 388–393, 2015.

[3] Shahanawaj Ahamad. Study of Software Aging Issues and Prevention Solutions. *International Journal of Computer Science and Information Security*, 14(08):307–313, 2016.

[4] Wouter A Alberts and Thea M Van Der Geest. Color Matters: Color as Trustworthiness Cue in Web sites. *Technical Communication*, 58(2):149–160, 2011.

[5] Seleste Albury and Steve Kim. Facebook Forensics: Preliminary Studies on Open Graph Protocol and Location Services. *Proceedings of Student-Faculty Research Day*, 2012.

[6] R Michael Alvarez, Ines Levin, Alexander H Trechsel, and Kristjan Vassil. Voting advice applications: How useful and for whom? *Journal of Information Technology & Politics*, 11(1):82–101, 2014.

[7] Ioannis Andreadis. Voting Advice Applications: A Successful Nexus Between Informatics and Political Science. In *Proceedings of the 6th Balkan Conference in Informatics*, BCI '13, pages 251–258, New York, NY, USA, 2013. ACM.

[8] Miguel Barbosa, Trevor Blades, Joseph Homsy, Marvin Sanchez, and Matheson Murray. Pollenize America 2016. http://pollenize.org/america/. Accessed: 2017-09-08.

[9] Tom Barker. *High Performance Responsive Design - Building Faster Sites Across Devices*. O'Reilly Media, Inc., 2015.

[10] Thomas Beale. Archetypes: Constraint-based Domain Models for Futureproof Information Systems. *In OOPSLA 2002, workshop on behavioural semantics.*, pages 1–18, 2002.

[11] Michael L. Bernard, Barbara S. Chaparro, Melissa M. Mills, and Charles G. Halcomb. Comparing the effects of text size and format on the readibility of computer-displayed Times New Roman and arial text. *International Journal of Human Computer Studies*, 59(6):823–835, 2003.

[12] Iris Bonavida and Karl Ettinger. Nachspielzeit für Koalition läuft. http://diepresse.com/home/innenpolitik/5219010/Nachspielzeit-fuer-Koalition-laeuft, 2017. Accessed: 2017-09-11.

[13] Bundeszentrale für politische Bildung. Wahl-O-Mat. https://www.wahl-o-mat.de/. Accessed: 2017-09-21.

[14] Bundeszentrale für politische Bildung. Bundestagswahl 2017 | bpb. http://www.bpb.de/politik/wahlen/wer-steht-zur-wahl/bundestagswahl-2017/, 2017. Accessed: 2017-09-21.

[15] Jerry Cao. Web design color theory: how to create the right emotions with color in web design, 2015.

[16] Jerry Cao, Kamil Zieba, Matt Ellis, and Krzysztof Stryjewski. *Web UI Design for the Human Eye - Colors, Space, Contrast*. UXpin, 2015.

[17] Jerry Cao, Kamil Zieba, Krzysztof Stryjewski, and Matt Ellis. *Color Theory in Web UI Design: A Pratical Approach to the Principles*. UXPin, 2015.

[18] Alex Cazanas and Esther Parra. Strategies for Mobile Web Design. *INCISCOS 2016*, 2016.

[19] Lorella Cedroni. Voting Advice Applications in Europe: A Comparison. *Voting Advice Applications in Europe: The State of Art*, pages 247–258, 2010.

[20] Lorella Cedroni and Diego Garzia. Voting advice applications in Europe. *The State of the Art. Napoli: ScriptaWeb*, 2010.

[21] Center for Persons with Disabilities USA. WebAIM: Color Contrast Checker. https://webaim.org/resources/contrastchecker/. Accessed: 2017-09-25.

[22] Jae Min Cha, Eunjung Choi, and Jihyoun Lim. Virtual Sliding QWERTY: A new text entry method for smartwatches using Tap-N-Drag. *Applied Ergonomics*, 51:263–272, 2015.

[23] John Connell and Linda Brice. Prolonging the life of software. In *Proceedings of the July 9-12, 1984, national computer conference and exposition*, pages 243–249. ACM, 1984.

[24] Gary Cook, Tom Dowdall, David Pomerantz, and Yifei Wang. Clicking Clean : How Companies are Creating the Green Internet. *Greenpeace International*, pages 1–84, 2014.

[25] María Laura T Cossio, Laura F Giesen, Gabriela Araya, María Luisa S Pérez-Cotapos, RICARDO LÓPEZ VERGARA, Maura Manca, R. A. Tohme, S. D. Holmberg, Tim Bressmann, Daniel Rodrigues Lirio, Jelitza Soto Román, Rodrigo Ganter Solís, Sanjay Thakur, SVD Nageswara Rao, E L Modelado, Artificial D E La, Cabeza Durante, U N A Tradición, Maya En, E L Espejo, D E L A S Fuentes, Universidad Autónoma De Yucatán, Cruz Moreno Lenin, Laura Franco Cian, M Joanne Douglas, La Plata, and Françoise Héritier. Web Design Book of Trends 2015-2016. *Free E-Book*, XXXIII(2):81–87, 2012.

[26] Domenico Cotroneo, Roberto Natella, Roberto Pietrantuono, and Stefano Russo. A survey of software aging and rejuvenation studies. *ACM Journal on Emerging Technologies in Computing Systems*, 10(1):1–34, 2014.

[27] Carrie Cousins and Jerry Cao. *Web Design Book of Trends 2017*. UXPin, 2017.

[28] Alexander Dawson. *Future-Proof Web Design*. Wiley Publishing, 1st edition, 2011.

[29] Jochum De Graaf. The irresistible rise of Stemwijzer. *Voting Advice Applications in Europe: The state of the art*, pages 35–46, 2010.

[30] Hyrum Denney. A Sticky Menu Is Quicker To Navigate, 2012. Accessed: 2017-09-18.

[31] DerStandard.at. Kurz: Habe klare Vorstellungen, Programm aber erst im September. *derstandard.at on May 28th*, 2017.

[32] DerStandard.at. Wahlkabine.at: Pilz für Überwachung, FPÖ für Seenotrettung, sep 2017.

[33] DiePresse.com. wahlkabine.at: Entscheidung im Internet, sep 2017.

[34] Quang Do, Ben Martini, and Kim Kwang Raymond Choo. Is the data on your wearable device secure? An Android Wear smartwatch case study. In *Software - Practice and Experience*, volume 47, pages 391–403, 2017.

[35] DoubleClick by Google. The need for mobile speed: Better user experiences, greater publisher revenue. Technical report, 2016.

[36] Laura L Downey. Group Usability Testing : Evolution in Usability Techniques. *Journal of Usability Studies*, 2(3):133–144, 2007.

[37] Zsolt Enyedi. The Influence of Voting Advice Applications on Preferences, Loyalties and Turnout: An Experimental Study. *Political Studies*, 64(4):1000–1015, 2016.

[38] Henry Farrell. The Consequences of the Internet for Politics. *Annual Review of Political Science*, 15(1):35–52, 2012.

[39] Stefan Fenz. *Online-Wahlhilfen. Ein Vergleich der Antwortprofile und der im Wahlkampf generell kommunizierten politischen Positionen österreichischer Parteien.* PhD thesis, University of Vienna, 2009.

[40] Jan Fivaz and Daniel Schwarz. Smart Democracy f{ü}r Smart Cities – Online-Wahlhilfen und ihr Beitrag zu einer modernen Demokratie. *HMD Praxis der Wirtschaftsinformatik*, 52(4):482–501, aug 2015.

[41] Jan Fivaz, Daniel Schwarz, and Albert Waajenberg. smartvote. https://smartvote.ch/. Accessed: 2017-09-21.

[42] Brian Fling. Mobile Design and Development. In *Mobile Design and Development : Practical concepts and techniques for creating mobile sites and web apps*, page 336. O'Reilly Media, Inc., 2009.

[43] Thomas Fossen and Joel Anderson. What's the point of voting advice applications? Competing perspectives on democracy and citizenship. *Electoral Studies*, 36:244–251, 2014.

[44] Tim Frick. *Designing for Sustainability: A Guide to Building Greener Digital Products and Services*. " O'Reilly Media, Inc.", 2016.

[45] Brett S Gardner. Responsive web design: Enriching the user experience. *Sigma Journal: Inside the Digital Ecosystem*, 11(1):13–19, 2011.

[46] Diego Garzia. The effects of VAAs on users voting behaviour: An overview. *Voting Advice Applications in Europe: The state of the art*, pages 13–33, 2010.

[47] Diego Garzia and Stefan Marschall. Voting Advice Applications under review: the state of research. *Int. J. Electronic Governance*, 5(3/4):203–222, 2012.

[48] Diego Garzia and Stefan Marschall. Research on Voting Advice Applications: State of the Art and Future Directions. *Policy and Internet*, 8(4):376–390, 2016.

[49] Diego Garzia, Alexander Trechsel, and Lorenzo De Sio. Party placement in supranational elections. *Party Politics*, 23(4):333–341, 2017.

[50] Diego Garzia, Alexander H Trechsel, and Andrea De Angelis. Voting Advice Applications and Electoral Participation: A Multi-Method Study. *Political Communication*, pages 1–20, 2017.

[51] Anwar Ghuloum, Terry Smith, Gansha Wu, Xin Zhou, Jesse Fang, Peng Guo, Byoungro So, Mohan Rajagopalan, Yongjian Chen, and Biao Chen. Future-Proof Data Parallel Algorithms and Software on Intel Multi-Core Architecture. *Intel Technology*, 11(4), 2007.

[52] Ramiro Gonçalves, José Martins, Jorge Pereira, Manuel Au Yong Oliveira, and João José P Ferreira. Enterprise Web Accessibility Levels Amongst the Forbes 250: Where Art Thou O Virtuous Leader? *Journal of Business Ethics*, 113(2):363–375, 2013.

[53] Lev Grossman. Invention of the Year: The iPhone. (Cover story), 2007.

[54] J P Guilford and P Smith. A system of color preferences. *American Journal of Psychology*, 72:487–502, 1959.

[55] Aaron Gustafson. *Adaptive web design: crafting rich experiences with progressive enhancement*. New Riders, 2015.

[56] Richard H Hall and Patrick Hanna. The impact of web page text-background colour combinations on readability, retention, aesthetics and behavioural intention. *Behaviour & Information Technology*, 23(3):183–195, 2004.

[57] Sandra Harris. Evasive action: How politicians respond to questions in political interviews. *Broadcast talk*, 7699, 1991.

[58] Rachel Harrison, Derek Flood, and David Duce. Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Science*, 1(1):1, 2013.

[59] Chris Haycock and Mark Stevens. Who Shall I Vote For? http://www.whoshallivotefor.com. Accessed: 2017-09-22.

[60] I Heitlager, T Kuipers, and J Visser. A Practical Model for Measuring Maintainability, 2007.

[61] Tasha Hollingsed and David G. Novick. Usability inspection methods after 15 years of research and practice. In *Proceedings of the 25th annual ACM international conference on Design of communication - SIGDOC '07*, pages 249–255, 2007.

[62] Simon Holmes. *Getting MEAN with Mongo, Express, Angular, and Node*. 2013.

[63] Kai Huotari and Juho Hamari. Defining Gamification - A Service Marketing Perspective. *Proceeding of the 16th International Academic MindTrek Conference on - MindTrek '12*, pages 17–22, 2012.

[64] Institut für neue Kulturtechnologien/t0. wahlkabine.at. https://wahlkabine.at/. Accessed: 2017-09-07.

[65] International Organization for Standardization. ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) - part 11: guidance on usability. *International Organization for Standardization*, 1998(2):28, 1998.

[66] International Organization For Standardization Iso. ISO/IEC 9126-1, 2001.

[67] Madhuri A Jadhav, Balkrishna R Sawant, and Anushree Deshmukh. Single Page Application using AngularJS. *International Journal of Computer Science and Information Technologies*, 6(3):2876–2879, 2015.

[68] Mikael Jergefelt. Abandoning native apps to become future friendly. *Journal of EAHIL*, 11(2):36–38, 2015.

[69] Tim Kadlec. *Implementing Responsive Design: Building sites for an anywhere, everywhere web.* New Riders, 2012.

[70] Noriaki Kano, Nobuhiku Seraku, Fumio Takahashi, and Shinichi Tsuji. Attractive quality and must-be quality. *Journal of the Japanese Society for Quality Control*, 14(2):39–48, 1984.

[71] Joseph Robert Kerr. *Future-Proof Design: Must all Good Things Come to an End?* PhD thesis, University of Calgary (Canada), 2011.

[72] Bohyun Kim. Responsive Web Design, Discoverability, and Mobile Challenge. *Library Technology Reports*, 49(6):29–30, 2013.

[73] Jinwoo Kim and Jae Yun Moon. Designing towards emotional usability in customer interfaces—trustworthiness of cyber-banking system interfaces. *Interacting with Computers*, 10(97):1–29, 1998.

[74] André Krouwel. Kieskompas.nl. https://kieskompas.nl/. Accessed: 2017-09-21.

[75] Andreas Ladner, Gabriela Felder, and Jan Fivaz. More than toys? A first assessment of voting advice applications in Switzerland. *Voting Advice Applications in Europe. The State of the Art*, pages 91–123, 2010.

[76] Benjamin LaGrone. *HTML5 and CSS3 Responsive Web Design Cookbook.* Packt Publishing Ltd, 2013.

[77] Jeonghyun Lee, Imsu Lee, Iseul Kwon, Hyejin Yun, Jongwon Lee, Mansung Jung, and Hyenki Kim. Responsive web design according to the resolution. In *Proceedings - 8th International Conference on u- and e-Service, Science and Technology, UNESST 2015*, pages 1–5, 2016.

[78] J. Lees-Marshment, Y. Dufresne, G. Eady, D. Osborne, C. van der Linden, and J. Vowles. *Vote Compass in the 2014 New Zealand election: Hearing the voice of New Zealand voters*, volume 67. 2015.

[79] Ethan Marcotte. *Responsive Web Design.* A Book Apart, 2011.

[80] Stefan Marschall. Idee und wirkung des wahl-o-mat. *Aus Politik und Zeitgeschichte*, 51(52):41–46, 2005.

[81] Stefan Marschall and Martin Schultze. Voting Advice Applications and their effect on voter turnout: the case of the German Wahl–O–Mat. *International Journal of Electronic Governance*, 5(3/4):349–366, 2012.

[82] Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship.* Pearson Education, Inc., 2009.

[83] Christine Mayer and Martin Wassermair. wahlkabine. at: Promoting an Enlightened Understanding of Politics. *Voting Advice Applications in Europe: The State of the Art. Scriptaweb*, 2010.

[84] Chrstine Mayer and Monika Mokre. Why Austria's most successful vote match tool is now more important than ever. *Elections in Europe Volume 5: Participation in Electoral Processes. ACEEEO (Association of European Election Officials)*, 5:18–20, 2009.

[85] Deborah J. Mayhew. The usability engineering lifecycle. *CHI'99 Extended Abstracts on Human Factors in Computer Systems*, pages 147–148, 1999.

[86] Ian McAllister. The personalization of politics. *The Oxford handbook of political*, 2007.

[87] Elise Marie Leonore Moerel. Big data protection: How to make the draft EU Regulation on Data Protection Future Proof. *Lokke Moerel, Tilburg University*, 2014.

[88] Ahmad Zamzuri Mohamad Ali, Rahani Wahid, Khairulanuar Samsudin, and Muhammad Zaffwan Idris. Reading on the computer screen: Does font type has effects on Web text readability? *International Education Studies*, 6(3):26–35, 2013.

[89] Peter Morville. *Ambient Findability: What We Find Changes Who We Become*, volume 25. O'Reilly, 2005.

[90] Jakob Nielsen. Usability inspection methods. In *Conference companion on Human factors in computing systems - CHI '94*, pages 413–414, 1994.

[91] Jakob Nielsen. Ten Usability Heuristics. *Communications of the ACM*, 3(1990):1–2, 2005.

[92] Jakob Nielsen and Rolf Molich. Heuristic Evaluation of user interfaces. *CHI '90 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (April):249–256, 1990.

[93] D A Norman and S W Draper. User centered system design. *... on Human-Computer Interaction*, 1986.

[94] Tomoyuki Ohkubo and Kazuyuki Kobayashi. A color compensation vision system for color-blind people. In *Proceedings of the SICE Annual Conference*, pages 1286–1289, 2008.

[95] Todd Parker, Patty Toland, Scott Jehl, and Maggie Costello Wachs. *Designing with progressive enhancement*. New Riders, 2010.

[96] David Lorge Parnas. Software aging. In *Proceedings of 16th International Conference on Software Engineering*, pages 279–287, 1994.

[97] Serena Pastore. Website development and web standards in the ubiquitous world: Where are we going? *WSEAS Transactions on Computers*, 11(9):309–318, 2012.

[98] Andrew Perks. Advanced vision guided robotics provide "future-proof" flexible automation. *Assembly Automation*, 26(3):216–220, 2006.

[99] Pingdom. Website Speed Test. https://tools.pingdom.com/. Accessed: 2017-09-30.

[100] Guy Podjarny. Performance Implications of Responsive Design. In Michael Nolan, editor, *Implementing Responsive Design*, pages 102–103. Pearson Education, Inc., 2012.

[101] Guy Podjarny. *Responsive & Fast: Implementing High-Performance Responsive Design.* O'Reilly Media, Inc., 2014.

[102] Jacob Poushter. Smartphone Ownership and Internet Usage Continues to Climb in Emerging Economies. Technical report, Pew Research Center, Washington DC, 2016.

[103] G. Bingham Powell. *Elections as Instruments of Democracy.* Yale University Press, 2000.

[104] Christopher Power, André Freire, Helen Petrie, and David Swallow. Guidelines are only half of the story: accessibility problems encountered by blind users on the web. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 433–442. ACM, 2012.

[105] ProDemos. StemWijzer. Accessed: 2017-09-21.

[106] José Rodrigues-Filho, Cynthia J. Alexander, and Luciano C. Batista. E-Voting in Brazil - The Risks to Democracy. *Electronic Voting 2006, GI Lecture Notes in Informatics*, pages 85–94, 2006.

[107] Martin Rosema, Joel Anderson, and Stefaan Walgrave. The design, purpose, and effects of voting advice applications. *Electoral studies*, 36:240–243, 2014.

[108] Doralyn Rossmann and Scott W H Young. Social Media Optimization: Principles for Building and Engaging Community. *Library Technology Reports*, 52(8):1, 2016.

[109] Jeffrey Rubin and Dana Chisnell. *Handbook Of Usability Testing.* 2nd edition, 2008.

[110] Sandijs Ruluks. 9 basic principles of responsive web design, 2014.

[111] Bernhard Rütsche. *Elektronische Wahlhilfen in der Demokratie: Beurteilung im Lichte der Wahl-und Abstimmungsfreiheit, Spannungsverhältnis zu den politischen Parteien, Fragen staatlicher Regulierung.* Helbing Lichtenhahn, 2008.

[112] Outi Ruusuvirta. Much ado about nothing? Online voting advice applications in Finland. *Voting advice applications in Europe. The state of the art. ScriptaWeb, Napoli*, pages 47–77, 2010.

[113] Bernd Schueler. Farben als Wegweiser in der Politik, 2006.

[114] Winfried Schulz. *Politische Kommunikation: Theoretische Ans{ä}tze und Ergebnisse empirischer Forschung*. Springer-Verlag, 2009.

[115] Dan Scott. White Hat Search Engine Optimization (SEO): Structured Web Data for Libraries. *Partnership: The Canadian Journal of Library and Information Practice and Research*, 10(1):1–21, 2015.

[116] Andrew Sears. Heuristic Walkthroughs: Finding the Problems Without the Noise. *International Journal of Human-Computer Interaction*, 9(3):213–234, 1997.

[117] SeoSiteCheckup.com. SEO Checkup. https://seositecheckup.com/. Accessed: 2017-09-25.

[118] Sujan Shrestha. Mobile web browsing: usability study. In *Mobility 07 Proceedings of the 4th international conference on mobile technology applications and systems and the 1st international symposium on Computer human interaction in mobile technology*, volume Singapore, pages 187–194, 2007.

[119] Donna Spencer. How to Conduct A Content Audit. https://uxmastery.com/how-to-conduct-a-content-audit/. Accessed: 2017-09-18.

[120] Rick Spencer. The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '00*, pages 353–359, 2000.

[121] Gerfried Sperl. Statt Schwarz nun Türkis: Von eindeutig zu schillernd. http://derstandard.at/2000058354239/Statt-Schwarz-nun-Tuerkis-Von-eindeutig-zu-schillernd, may 2017. Accessed: 2017-09-17.

[122] Andreas Taranetz. *Building a high-performance and long-term maintainable web application on the example of wahlkabine.at*. PhD thesis, Vienna University of Technology, 2017.

[123] Lisbeth Thorlacius. The Role of Aesthetics in Web Design. *Nordicom Review*, 28:63–76, 2007.

[124] Stanley Trollip and Gregory Sales. Readability of Computer-Generated Fill-Justified Text. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 28(3):159–163, 1986.

[125] Kristjan Vassil. *Voting Smarter? The impact of voting advice applications on political behavior*. PhD thesis, Citeseer, 2011.

[126] Kristjan Vassil, Mihkel Solvak, Priit Vinkel, Alexander H. Trechsel, and R. Michael Alvarez. The diffusion of internet voting. Usage patterns of internet voting in Estonia between 2005 and 2015. *Government Information Quarterly*, 33(3):453–459, 2016.

[127] Jeffrey Veen. Doing a Content Inventory (Or, A Mind-Numbingly Detailed Odyssey Through Your Web Site). http://adaptivepath.org/ideas/doing-content-inventory/, 2002. Accessed: 2017-09-18.

[128] Vox Pop Labs. Vote Compass. http://votecompass.com/. Accessed: 2017-09-28.

[129] W3C WAI. How to Meet WCAG 2.0. http://www.w3.org/WAI/WCAG20/quickref/, 2008. Accessed: 2017-09-11.

[130] Stefaan Walgrave, Peter van Aelst, and Michiel Nuytemans. 'Do the Vote Test': The Electoral Effects of a Popular Vote Advice Application at the 2004 Belgian Elections. *Acta Politica*, 43(1):50–70, 2008.

[131] M. Wall, A. Krouwel, and T. Vitiello. Do voters follow the recommendations of voter advice application websites? A study of the effects of kieskompas.nl on its users' vote choices in the 2010 Dutch legislative elections. *Party Politics*, 20(3):416–428, 2014.

[132] WebAIM. WAVE Web Accessibility Tool. http://wave.webaim.org/. Accessed: 2017-09-24.

[133] Westdeutscher Rundfunk Köln. Der Kandidatencheck | WDR.de. http://kandidatencheck.wdr.de/kandidatencheck/. Accessed: 2017-09-08.

[134] Cathleen Wharton, John Rieman, Clayton Lewis, and Peter Polson. The cognitive walkthrough method: A practitioner's guide, 1994.

[135] Marcin Wichary. Using UI System Fonts In Web Design: A Quick Practical Guide. https://www.smashingmagazine.com/2015/11/using-system-ui-fonts-practical-guide/, 2015. Accessed: 2017-09-20.

[136] Jens Wiese. Zur Bundestagswahl: Facebook stellt Politikern und Parteien neuen "Themen" Tab zur Verfügung - allfacebook.de. https://allfacebook.de/allgemeines/bundestagswahl-facebook-themen-tab-politiker. Accessed: 2017-09-12.

[137] World Commission on Environment and Development. Report of the World Commission on Environment and Development: Our Common Future (The Brundtland Report). *Medicine, Conflict and Survival*, 4(1):300, 1987.

[138] World Wide Web Consortium. The W3C CSS Validation Service. https://jigsaw.w3.org/css-validator/. Accessed: 2017-10-05.

[139] World Wide Web Consortium. The W3C Markup Validation Service. https://validator.w3.org/. Accessed: 2017-10-05.

[140] Luke Wroblewski, Brad Frost, Lyza Gardner, Stephanie Rieger, Bryan Rieger, Tim Kadlec, Andrea Trasatti, Scott Jenson, Jeremy Keith, Scott Jehl, Jason Grigsby, Josh Clark, and Brian Leroux. Future Friendly, 2011.

[141] Pallavi Yadav and Paras Nath Barwal. Designing Responsive Websites Using HTML And CSS. *International Journal Of Scientific & Technology Research*, 3(11):152–155, 2014.

[142] Jamaiah H Yahaya, Zaiha Nadiah Zainal Abidin, Noorazean Mohd Ali, and Aziz Deraman. Software ageing measurement and classification using Goal Question Metric (GQM) approach. In *Science and Information Conference (SAI), 2013*, pages 160–165. IEEE, 2013.

[143] Jamaiah H Yahaya and Aziz Deraman. Towards the anti-ageing Model for Application software. In *Proceedings of the World Congress on Engineering*, 2012.

[144] Lee Young. Great UX Design Involves Simplifying & Unifying User Experiences Across All Platforms & Devices. | LinkedIn. https://www.linkedin.com/pulse/great-ux-design-involves-simplifying-unifying-user-across-lee-young/, 2016. Accessed: 2017-09-25.