FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Efficient Integration of Usability and Accessibility into the Lifecycle of Web Development Projects

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Information & Knowledge Management

eingereicht von

Tanja Wenter
Matrikelnummer 0405221

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Associate Prof. Dipl.-Ing. Dr.techn. Hilda Tellioğlu
Mitwirkung: Univ.Ass. Florian Cech, MSc

Wien, 30. Oktober 2017

Tanja Wenter

Hilda Tellioğlu

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Efficient Integration of Usability and Accessibility into the Lifecycle of Web Development Projects

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Information & Knowledge Management

by

Tanja Wenter
Registration Number 0405221

to the Faculty of Informatics

at the TU Wien

Advisor: Associate Prof. Dipl.-Ing. Dr.techn. Hilda Tellioğlu
Assistance: Univ.Ass. Florian Cech, MSc

Vienna, 30th October, 2017

Tanja Wenter	Hilda Tellioğlu

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

Tanja Wenter
Druckergasse 13, 6850 Dornbirn

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, 30. Oktober 2017

Tanja Wenter

Acknowledgements

First, I owe thanks to my advisor, ao. Univ.-Prof. Dr. Hilda Tellioğlu, for her scientific advice, her interest in my topic as well as her patience with my progress. Furthermore, I would like to thank my assistant advisor Florian Cech, MSc for his valuable inputs, his open mind for new ideas, and his continuous encouragement.

To my former employer and friend, DI Michael Stenitzer, who not only inspired me, but also awoke my interest and joy for Accessibility, I owe thanks.

Also, I would like to thank my family, especially my grandmother, for the values they thought me, their continuous support, and endorsement to use my talents in every possible way.

Last but not least, I owe a debt of gratitude to my boyfriend, Ralph. Not only his exceptional inputs, but also inspiring discussions and impulses to change perspectives, helped me during my thesis, and my whole life with him. I thank him for his patience, support and his critical review of my thesis.

Kurzfassung

Eine immer dichter werdende Vernetzung erlaubt es mehr und mehr Personen miteinander zu interagieren. Dies geschieht sowohl untereinander als auch mit DienstleisterInnen. Aus diesem Grund drängen klassische DienstleisterInnen, wie etwa Banken, mit ihren angebotenen Services ins Internet. Ein Großteil von Dienstleistungen wird bereits online abgewickelt. Um mit dieser Entwicklung Schritt zu halten und keine Personengruppen auszuschließen, ist es unerlässlich diese Dienstleistungen so zu gestalten, dass ein Großteil der Bevölkerung diese ohne Probleme beziehungsweise Nachteile nutzen kann. Diese Usability und Accessibility (Nutzbarkeit und Barrierefreiheit) Anforderungen sollten daher oberste Priorität haben. Dabei geht es um Anforderungen, die beispielsweise motorisch dauerhaft bzw. kurzfristig eingeschränkte Menschen berücksichtigen, aber auch eine Berücksichtigung kognitiver Einschränkungen oder die Beachtung von altersbedingten Beschwerden.

Leider werden diese Anforderungen oft nur sehr rudimentär umgesetzt. Während der Entwicklungsphase von Web-Applikationen finden diese Anforderungen oft keine Beachtung und werden erst nachträglich berücksichtigt. Dies verursacht sowohl hohe Kosten und Mehraufwand für die AnbieterInnen und die EntwicklerInnen als auch Unzufriedenheit seitens der EndverbraucherInnen. Ein einheitlicher Entwicklungsprozess, bei dem Usability und Accessibility im Fokus steht, wäre wünschenswert.

Diesem Punkt widmet sich die vorliegende Arbeit. Ein auf agilen Entwicklungsmethoden basierender Entwicklungsprozess wird definiert, welcher für Webentwicklungen in jeder Phase des Prozesses den Fokus für Usability und Accessibility schärft. Mittels Experteninterviews wurde ein breites Spektrum an Usability- und Accessibility-Ansätzen untersucht. In dem vorliegenden Prozess werden gängige Standards, wie die *Web Content Accessibility Guidelines (WCAG)*, berücksichtigt und deren Anforderung für den Entwicklungsprozess aufbereitet. Dies erlaubt es EntwicklerInnen, eine vollständige Spezifikation zu erhalten und Usability und Accessibility direkt in der Implementierungsphase zu berücksichtigen.

Dazu wird für jeden Schritt des Entwicklungsprozesses, beginnend mit der Konzeptionsphase, ein klar festgelegter Regelsatz, basierend auf den WCAG Richtlinien, definiert. Dieser Regelsatz beschreibt, was in der jeweiligen Prozessphase berücksichtigt werden muss, um Webseiten barrierefrei beziehungsweise barrierearm umzusetzen. Weiters wird berücksichtigt, dass der darauffolgende Prozessschritt alle notwendigen Information erhält.

Dadurch werden Information während des Entwicklungsprozesses weitergereicht und so sukzessive vervollständigt.

Durch diesen Proof-of-Concept Prozess sollen Entwicklungskosten niedrig gehalten und die Wartungskosten in Bezug auf Barrierefreiheit minimiert werden. Zusätzlich soll die Akzeptanz unter dem Benutzerkreis erhöht und die Verwendung der Dienstleistung einer breiteren Benutzergruppe ermöglicht werden.

Abstract

Nowadays, people are using the Internet not only to interact and connect with each other, but also to handle routine tasks offered by businesses. Since the user base is rapidly growing, a wide variety of business service providers, like for example banks, are increasing their offered online services. A large part of their services is already processed using the Internet via Websites. To keep pace with ever-changing requirements by diverse user groups, it is indispensable to design these services in a way that includes everybody. It should be possible for all users to access and use the offered online services without any problems or discrimination. Thus, Usability and Accessibility should be key requirements when designing Web interfaces or Web applications. This not only includes requirements for temporary or permanent movement disorders as well as considerations for cognitive limitations, but also respecting restraints which older people face.

Unfortunately, these Usability and Accessibility requirements are often ignored or poorly implemented. During the development of a Website they are commonly disregarded and suspended to a later phase. Hence, the integration of Usability and Accessibility features is often only considered as an afterthought. This not only causes higher costs, but also results in dissatisfaction and frustration for users as well as the developers. The goal should be a homogeneous process that includes Usability and Accessibility requirements from the beginning.

Such a homogeneous process is the focal point of the hereby presented work. It describes the definition of a process for Web development projects based on agile development methods. Each phase in this process has its emphasis on Usability and Accessibility. Expert interviews have been conducted to get acquainted with various approaches to Usability and Accessibility. The proposed process is built upon Usability and Accessibility standards like the Web Content Accessibility Guidelines (WCAG) and the requirements are formalized accordingly. Therefore, the process provides a full specification for the developers on how to integrate Usability and Accessibility during the development.

To accomplish such a process, a clearly defined set of rules, based on the WCAG standard, is necessary. Each step throughout the process is addressed accordingly. The rule-set describes what kind of considerations are necessary in each process phase to create an accessible Website. In addition, each step of the process provides information for the following step. This guarantees a continuous information flow and a steady development of accessible Websites.

This proof-of-concept process also helps to minimize the costs during development and reduce maintenance costs related to Accessibility requirements. By accomplishing accessible Websites through the defined process the consumer acceptance is also raised and the use of online services is made available for a broader user group.

Contents

CHAPTER 1

Introduction

Nowadays software is indispensable in our daily lives. We use software that supports us in our everyday tasks, may it be at work, in our hobbies, or just to simplify daily routines. Many computer-based products in a variety of areas exist and are used everyday by a diverse group of people. Functionality is one of the key factors for a successful software project. This is especially true for Web applications and Websites, where potentially thousands of people with different background use it at the same time. Thus, many software companies place their emphasis on the functionality and it often seems like the actual Usability is considered only as an afterthought. Accessibility, as a special kind of requirement for Usability, is even less considered during development. One reason might be that it requires effort to develop applications, that are accessible in multiple environments such as smart-phones or full-fetched desktop Web applications. Another reason is that developers are typically not well-trained in these areas. This is unsatisfactory for both the users consuming software as well as the companies providing the software. The users are hindered using the software and the companies developing the software have higher costs maintaining the software.

The goal of this thesis is to provide a methodology on how Accessibility, and with that also a baseline of Usability standards, can be integrated into a software development process for Web projects. A framework supporting such a process is proposed and described in detail. The process provides a standardized way for all stakeholders to consider Usability and Accessibility in each phase of the project life-cycle from the get-go. The result is a framework for a development process where each phase is enriched by guidelines focusing on Usability and Accessibility. The handling of these guidelines is supported by providing a toolbox with methodologies.

1.1 Motivation

The main motivation for this thesis was to support the creation of Websites that are accessible by a large variety of users. In our daily lives everyone comes to a point, where she/he is in any way handicapped in using software of any kind. Be it because of the sun that is shining into the screen, broken arms or because information is presented inconsistently. The majority of people don't think about cases in which Usability or Accessibility could be revised to make software more useful for themselves. They are not aware of what could be improved in software to make everyone's life easier. Statements like "This is too complicated for me", "I'm to stupid to use that" or "I don't know how that works" should not be reality. People see the reason of their failure in themselves, but not in the software. It should be the other way around and software should support us in our daily lives rather than frustrate the user.

The necessity of Usability in Websites has been studied for quite some time, but the minority of developers have a good understanding or "feeling" for it. Accessibility, as part of Usability, is even less known by developers. This is worrisome for users who are hindered to use Websites. In addition, it is also worrisome for the companies developing these Websites, since there are upcoming rules and regulations on a legal basis. Hence, Accessibility is becoming more and more important in our society. Still, most of the developers are not yet schooled in Usability and Accessibility techniques. Typically, their main goal is functionality, which is often even in contrast to the Usability. Thus a well-defined software development process with a focal point on Usability and Accessibility is needed.

1.2 Problem Statement

The user base of online services is a very diverse one and is rapidly growing. To fulfill the needs of this diverse user base simple and for everyone accessible User Interfaces (UIs) are of high importance. All the more they should be considered in every software application from the get-go. Focusing on an easily usable application has to be one of the top priorities in software development. Furthermore, it should be considered in each step of the software development project life-cycle starting from the beginning. However, many software companies are not aware of the importance of Usability and Accessibility and sometimes consider it only as an afterthought. Even if the awareness exists, it seems that there is often no clearly defined process, which uses Usability and Accessibility methodologies. The most common way of integrating Usability and Accessibility is to queue the integration up until the end of the development process. This results in either complete ignorance of the evaluated problems or a lot of extra effort for the needed improvements. In addition, this often leads to a prolonged project development time and further increases the costs.

Due to the high importance of the Internet and its constantly growing usage, more and more companies offer their services as Web applications. These applications are used by a wide variety of people. Thus, usable and for everyone accessible Websites should be a natural consequence. The life-cycle of Web development projects is similar to the one of classic software projects and many software project management approaches can be adopted. In fact, different project management approaches are already available and used in practice. However, they do not account for the specific issues that arise in the Usability and Accessibility areas. Especially Accessibility as a special type of Usability is not commonly supported during the development processes. This is particularly an issue for developers as they are often not well-trained in this area. A well-defined process with guidelines, that supports project managers during the entire software development life-cycle, could lead to a more efficient handling of Usability and Accessibility problems. In consequence a software developed following this approach is more easily accepted by the users.

The main questions that arise, and this thesis is focused on, are

- How is Usability and Accessibility defined? What standards do exist?
- Why is there a lack of awareness for Usability and Accessibility in software projects?
- How can we integrate the methodologies of Usability, Accessibility and Software Engineering?
- Is it possible to measure the integrated approach and its advantages?
- What tools and methods, concerning Usability and Accessibility, can be used to support software developers and project managers in the entire project life-cycle?

Since Usability is an extensive research topic, only parts of it can be studied in detail and this thesis focuses on Accessibility in Web applications and their environment. The thesis uses agile project management approaches and integrates Accessibility methodologies within them. One of the main focal points of this thesis is to create awareness for Usability and particularly Accessibility problems early on in the project life-cycle. Also, it provides guidance and guidelines for project managers as well as software developers tackling these issues. These guidelines provide developers with development methods that allow them to easily adopt their programming style for guaranteeing basic Accessibility standards. These methods and guidelines are utilized during the entire project development life-cycle. This means, that in different phases during the project life-cycle the project manager as well as the software developers receive feedback on the current status concerning Usability and Accessibility.

However, this thesis does not focus on the implementation of applications that support the project managers and software developers during the project life-cycle. So, the result of this thesis is a detailed specification of how and with which tools and methods, concerning Usability and Accessibility, project managers and software developers can be supported during the entire development process of a software project. The result is a well-defined development process with a focal point on Usability and Accessibility.

1.3 Related Work

The necessity to create awareness for Accessibility in Web applications has been well documented. Hashemian [Has11] studied 20 Finnish Websites of higher education Institutes and found that although the majority of the sites took Accessibility into consideration, most of the Websites still had Accessibility issues. According to Hashemian the main reason is the lack of Accessibility awareness of the developers. Also, Nganji et al [NBT13]. claim that one of the problems is, that the minority of Web developers are lacking knowledge about Accessibility standards such as WCAG and only few Web designer follow these guidelines at all.

In the area of project development supporting Usability there have already been some interesting approaches introduced. They take Software Engineering and Usability Engineering methodologies into account and build upon them.

A good overview on literature about how Usability Engineering can be integrated into Agile Software Development is given by Sohaib and Khan [SK10].

Brajnik [Bra08] describes different Accessibility evaluation methods. He gives a discussion on the problems arising when dealing with Accessibility. He provides an overview about Accessibility and presents reviews of several Accessibility evaluation methods.

Theofanos et al. [TR03] give an overview on their research project, where they observed 16 blind people over four months to see how they use the Web. The authors studied how blind people interact with Websites and how assistive technologies, e.g. screenreaders, are used. They give an overview on the problems the users stumbled upon and formulated guidelines on how to solve the identified issues. Even though their study was conducted in 2003/04 a lot of the issues still remain. Generally speaking the Accessibility of Websites has not been increased by a whole lot, but the assistive technologies have improved a lot.

Fischer[Fis12] enriches the software development process with Usability Engineering Standards. In his work he combines and links the existing Usability Engineering Standards with Software Engineering Standards to guarantee Usability integration and thus ensure a good level of quality concerning Usability. However, his work is primarily focused on the available International Organization for Standardization (ISO) standards for Usability Engineering and Software Engineering and does not account for Accessibility issues.

The research of Abuaddous et al. [AB16] shows the importance of finding a way to properly integrate Accessibility in Websites. The authors describe the existing problems concerning Accessibility and present the difficulties disabled people face. In their work they explain the reasons for the missing Accessibility implementations and investigated the main challenges during design, development and evaluation. Finally, they give recommendations to manage these challenges.

Lárusdóttir et al. [LCG12] conducted interviews with User Experience Experts and discussed their experience with Scrum-based projects. They give details about emerging challenges within the User Experience domain and show how the big picture of User

Experience is often missing. As their work is focused on finding issues through interviews, it does not provide an approach for a solution.

The recent literature review of Brhel et al. [BMMW15] is an extensive research on publications on the state of integrating Agile Software Development and User Centered Design. Their findings show that there are still open issues. They propose to further investigate in identifying principles in the social as well as the technological dimension. This paper can be used as a foundation that the hereby presented approach builds upon.

Nebe et al. [NZP08] define an approach for integrating Software Engineering and Usability Engineering. They built upon the according ISO standards from Software and Usability Engineering. In addition, they expand and enrich their findings with the results from expert interviews, and are trying to find an approach to integrate both disciplines on a process level. In contrast to this thesis they focus on merging the process models of Software Engineering and Usability Engineering, rather than how they can be integrated on a conceptual level. That is, they do not provide the means for a fully integrated process supported by a well-defined framework. In addition Accessibility is not considered as a separate discipline.

1.4 Structure of the Thesis

Chapter 2 provides some basic principles and definitions of the covered topics, mainly Usability and Accessibility. An overview on the subjects is given and a basic understanding for the topics covered in this thesis is provided. The legal fundamentals and why Accessibility is important are discussed as well.

The expert interviews, which helped to identify methodologies to use during the process development, are described in Chapter 3.

The following Chapter 4 describes the central idea of the thesis - the integrated process. It shows how a process with Accessibility as a main focal point is specified and how a framework supporting it could be implemented. Limitations of this integrated process are discussed as well.

Chapter 5 provides a toolbox with methods and techniques to support development during the proposed process.

The thesis ends with Chapter 6, which sums up the presented work and points out possibilities for future research.

CHAPTER 2

Theoretical Basics & Fundamentals

This chapter gives an overview of the basics and fundamentals that are necessary to fully understand the remaining parts of the thesis. In the first part of this chapter a short overview on key concepts of Usability is given and the next part provides a detailed overview on Accessibility. Finally the Kanban method, which the proposed framework of this thesis is based on, is outlined.

2.1 Usability

When surfing the Web we often find ourselves angry or frustrated due to the inability to efficiently work with it. Be it because of convoluted widgets, complicated navigation or because we “feel” we don’t get how the Website is supposed to work. Even experienced and Web-affine users stumble upon these barriers on Websites and decide to look for alternatives. Knowing this, it shows, that the problem is not the user, but very often to Website itself. Often they are simply not quite as usable as they are supposed to be. With that in mind it is clear, how important a clean and usable Web design is.

Usability describes how easy and pleasant a Website, a program or any software is to use. According to Nielsen [Nie93] it is important to see Usability not as a one dimensional property of an UI, but rather recognize that it consists of multiple facets. In his Usability definition Nielsen identifies the following five Usability attributes as the key facets:

Learnability It should be straightforward for new users to work with the system and achieve simple tasks adhoc - learning the system should be easy and intuitive.

Efficiency How quickly can a user, who knows and learned the system, perform tasks? The users productivity with the system should be on a high level.

Memorability Users, who haven't used the system for a prolonged time, should easily remember how to handle tasks and how the system operates.

Errors The system should prevent the user from making errors. This is especially true for severe errors. In case of any appearing errors, the user should be able to recover from them.

Satisfaction The system should be pleasant to use. That is, the user should like to work with it.

The ISO 9241-11 standard [fSIa] provides an often cited definition of Usability as follows

> *Usability is the extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.*

In the context of this quote it is necessary to provide the definition of effectiveness, efficiency and satisfaction. They define effectiveness as *the accuracy and completeness with which users achieve specified goals.* With efficiency they mean the *resources (time, human effort, costs and materials) used in relation to the results achieved.* Satisfaction is defined as *person's perceptions and responses that result from the use of a system, product or service.*

Analyzing the definitions from Nielsen and the ISO 9241-11 standard it is clear that they overlap. In essence they both describe the same concepts. The keys for having a usable product, respectively a usable Website, is, that users quickly understand what they do and how to achieve it. Websites should enable users to achieve their goals with minimal time and effort, while being pleasant to use.

Dumas and Redish [DR99] came up with a very similar definition of Usability. Their definition simple states, that people using a product need to be able to accomplish their tasks as quickly and easily as possible. They identify four scoring elements to accomplish this:

- *Usability means focusing on users.*
  The goal is to not just realize or implement the own ideas and wishes, but know and understand the potential target users and groups.

- *People use products to be productive.*
  Understanding the goals, or even better the purpose, of the user using a particular product or Website. Knowing what the users want to accomplish, helps designing the Website in a specific way. The design should focus on allowing users to achieve their goals quickly and easily.

- *Users are busy people trying to accomplish tasks.*
  Connecting usability with productivity - people judge a product's usability based on how productive they are using it. How much time does it take the user to accomplish what she/he wants?

- *Users decide when a product is easy to use.*
  Generally speaking, users only use a limited amount of the provided functionality of a Website. Thus, it is important to highlight the main functions and hide the complexity for advanced users. If users feel that the product is consistent, predictable and easy to use, they will learn more quickly and further explore the product.

After all, Usability should not be reduced to how "easy to use" a product or a Website is. The term describes a much more complex topic. Knowing why and how people use a product, instead of deciding for the user whether a product is simple to use or not, is the main goal of Usability [Kru14]. The user is in the center of the attention and the designers and development team have to make sure that the product they develop works well.

2.2 Accessibility

Accessibility in the context of this thesis deals with the ability of a software being used by a wide and diverse group of people with various disabilities. Although there are tools, see Section 2.2.7, available that support people with disabilities in their everyday lives, the software itself has to be developed in a specific way to provide the best possible results.

2.2.1 What is Web Accessibility?

It is clear to most what a disability is and that removing barriers, e.g. by providing a ramp for wheelchairs, in real life is important. In contrast, it is still not clear to most people what Web Accessibility means. A lot of people don't think of people not being able to use the Web and, of course, they also do not know what prevents a Website from not being accessible for everyone. Since Web Accessibility is not yet a popular topic a clear definition, on what Web Accessibility is, is missing.

The World Wide Web Consortium (W3C) defines Web Accessibility as follows:

> *The Web is fundamentally designed to work for all people, whatever their hardware, software, language, culture, location, or physical or mental ability. When the Web meets this goal, it is accessible to people with a diverse range of hearing, movement, sight, and cognitive ability.*[1]

[1] `https://www.w3.org/standards/webdesign/accessibility`, accessed Jul 2017

Though this definition is abstract it makes clear, that the Web should work for everyone and should be usable by any audience - it pictures the universality of the Web.

Another definition from the W3C Web Accessibility Initiative (WAI) has a stronger focus on disabled people and states:

> *Web accessibility means that people with disabilities can use the Web. More specifically, Web accessibility means that people with disabilities can perceive, understand, navigate, and interact with the Web, and that they can contribute to the Web. Web accessibility also benefits others, including older people with changing abilities due to aging.*
>
> *Web accessibility encompasses all disabilities that affect access to the Web, including visual, auditory, physical, speech, cognitive, and neurological disabilities.*[2]

This definition targets people who face barriers when accessing the Web. Users with any kind of permanent or short-term disabilities should be able to work with the Web regardless of her/his environment or technical limitations. The content needs to be obtainable for everyone, no matter which tools or technologies are used. Furthermore, it should be independent from the surrounded setting.

Another definition is given by Wikipedia on their page on Web Accessibility. It is mentioned here because of its simplicity and clear definition of Web Accessibility.

> *Web accessibility refers to the inclusive practice of removing barriers that prevent interaction with or access to websites. When sites are correctly designed, developed and edited, all users have equal access to information and functionality.*[3]

This definition shows the implicitness of the topic and also gives an insight on how easy it would be to design barrier-free Websites.

2.2.2 The World Wide Web Consortium (W3C)

The international organization World Wide Web Consortium (W3C)[4] develops standards for the World Wide Web (WWW). The founder and current leader is Tim Berners-Lee - one of the key figures in developing the Web - together with CEO Jeffrey Jaffe. As of July 2017 the W3C counts 476 members worldwide working on developing Web Standards. The mission[5] of the organization is to lead the Web to its full potential. The W3C defines *Open Standards Principles* and *Design Principles* to fulfill the vision of a standardized Web.

[2] `https://www.w3.org/WAI/intro/accessibility.php`, accessed Jul 2017

[3] `https://en.wikipedia.org/wiki/Web_accessibility`, accessed Jul 2017

[4] `https://www.w3.org/`, accessed Jul 2017

[5] `https://www.w3.org/Consortium/mission`, accessed Jul 2017

Open Standards Principles

The *Open Standards Principles*[6] define an agreement on a set of principles of five global organizations to establish *The Modern Paradigm for Standard - OpenStand.* The base is an effective and efficient standardization of processes on which the modern Internet and Web is based on.

Design Principles

W3C's work is based on *Design Principles* which are the foundation for a standardized Web.

- **Web for All**

  The Web, including all its social and human values such as communication, commerce and sharing opportunities, should benefit all people. Thus, it should be usable and accessible by everyone. W3C's goals are to make the Web's benefits fully available independent of by whom, where or how it is accessed.

  - Web Accessibility Initiative (WAI)[7]
  - Internationalization[8]
  - Mobile Web for Social Development[9]

- **Web on Everything**

  Increasingly more different devices are able to access the Web and use its functionality. Hence, Websites need to support a wide variety of devices. These devices differ, for example, in screen sizes and browsers they use. Websites should be developed in a way that allows to use them without any limitations by any device.

  - Web of Devices[10]
  - Mobile Web Initiative[11]
  - Browsers and Other Agents[12]

When the Web was first introduced, the inventors aim was to use it as a communication and sharing tool for everyone regardless of the user, the device and the location. The W3C pursues this idea by an active participation on the further development of the Web and helps to create a standardized basis [Con16b].

[6] `https://open-stand.org/about-us/principles/`, accessed Jul 2017

[7] `https://www.w3.org/WAI/`, accessed Jul 2017

[8] `https://www.w3.org/International/`, accessed Jul 2017

[9] `https://www.w3.org/2008/MW4D/`, accessed Jul 2017

[10] `https://www.w3.org/standards/webofdevices/`, accessed Jul 2017

[11] `https://www.w3.org/Mobile/`, accessed Jul 2017

[12] `https://www.w3.org/standards/agents/Overview.html`, accessed Jul 2017

2.2.3 Web Accessibility Initiative (WAI)

The WAI is an initiative of the W3C, divided in various working groups, focusing on Web Accessibility. WAI is working on principles, guidelines and educational material to support developers and help getting a better understanding on how to standardize the Web and thus make it accessible for everyone [Con16a].

2.2.4 Web Content Accessibility Guidelines (WCAG)

For this thesis the most important of the WAI guidelines are the Web Content Accessibility Guidelines (WCAG). They were first introduced in May 1999. WCAG contains a set of rules for a more accessible Web for disabled people. The guidelines aim to improve the overall Usability, responsive behaviour, enhanced Search Engine Optimization (SEO) functionality and in general optimized machine readability. The WCAG recommendation provides reference for developers to create a barrier-free Web involving 14 principles and guidelines.

- Guideline 1: Provide equivalent alternatives to auditory and visual content
- Guideline 2: Don't rely on colour alone
- Guideline 3: Use markup and style sheets, and do so properly
- Guideline 4: Clarify natural language usage
- Guideline 5: Create tables that transform gracefully
- Guideline 6: Ensure that pages featuring new technologies transform gracefully
- Guideline 7: Ensure user control of time sensitive content changes
- Guideline 8: Ensure direct accessibility of embedded user interfaces
- Guideline 9: Design for device independence
- Guideline 10: User interim solutions
- Guideline 11: Use W3C technologies and guidelines
- Guideline 12: Provide context and orientation information
- Guideline 13: Provide clear navigation mechanisms
- Guideline 14: Ensure that documents are clear and simple

Each guideline consists of various checkpoints assisting developers in designing accessible solutions [Con99].

An optimization and extension replaced the WCAG 1.0 in December 2008 with the updated WCAG 2.0. This new version includes twelve guidelines organized within four principles. Since techniques to accomplish accessibility are secondary, this version of the guideline adds much more flexibility and more open space for the developer.

The POUR Principle

The *POUR Principle* allows to test and verify the level of Accessibility. It consists of four principles, *Perceivable, Operable, Understandable* and *Robust*, and each principle is divided into various success criteria [fPwD16]. These success criteria are used to define the level of Accessibility. Depending on how many of the criteria are fulfilled it is determined between level A - the lowest, AA and AAA - presenting the highest level of conformance [Con08].

Perceivable - Information and UI components must be presentable to users in ways they can perceive.

- Guideline 1.1 Text Alternatives
- Guideline 1.2 Time-based Media
- Guideline 1.3 Adaptable
- Guideline 1.4 Distinguishable

No matter who - different persons with or without disabilities and also agents (e.g. search engines) - is accessing a Website, the content, meaning the information on the site and also the UI, must address the special needs of the visitor. Appropriate alternatives have to be considered and provided accordingly.

Operable - User interface components and navigation must be operable.

- Guideline 2.1 Keyboard Accessible
- Guideline 2.2 Enough Time
- Guideline 2.3 Seizures
- Guideline 2.4 Navigable

The different ways or variety of devices that use a Website have to be considered. The navigation possibilities for a user are not limited to a mouse and thus operation alternatives have to be served as well.

Understandable - Information and the operation of UIs must be understandable.

- Guideline 3.1 Readable
- Guideline 3.2 Predictable
- Guideline 3.3 Input Assistance

The user needs to be able to easily recognize and grasp the intention or functions of a Website and its content. The Website needs to be designed in a way that any user is able to understand it, work with it and does not get lost on it.

Robust - Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

- Guideline 4.1 Compatible

This technology dependant principle states, that the code should be clean and respect standards. In essence, different browsers should be able to interpret and render the page without problems.

In 2012 the WCAG became an ISO standard [fSIb]. Currently the W3C is working on further updating the WCAG standard to version 2.1[13]. The final version of the update is supposed to be published in 2018 and focuses on low vision, cognitive and learning disabilities as well as mobile accessibility.

2.2.5 Why Web Accessibility?

> *"We need to make every single thing accessible to every single person with a disability."* (Stevie Wonder, Grammy Awards 2016)

The best way to explain why Web Accessibility is important is not arguing about laws, guidelines and rules, but because it has to be done to make life easier for people with disabilities. The Web should be a universal place, where everyone counts the same and everyone has identical rights. This is not because of some ideological spirit, but because this is how the Web was intended to be. In other words, to quote the inventor of the World Wide Web (WWW) and Director of the W3C Tim Berners-Lee:

> *"The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect."* (Tim Berners-Lee)

[13] `https://www.w3.org/WAI/WCAG20/wcag2faq#v21`, accessed Jun 2017

Nowadays information is one of the most important goods. A quick and constant availability as well as simple utilization of data, information and knowledge is indispensable. Not only is the number of people using and accessing the Web constantly growing, but also the amount of offered services is rapidly rising [LGC10]. Public authorities and also private companies are ever expanding their online offers and minimize their offline portfolio. The rise of the Internet and the provided online services is inevitable, which makes an ease of use and access for them essential.

Worldwide 49,7% of the population are using the Internet. This number increases for Europe to 77,4% and in North America 88,1% use the Internet regulary[14]. For Austria the statistic of online usage is similar to North America, but the growth within the past ten years is much bigger. In 2016 already 84,3% of the people are using the Internet, whereas in 2002 only 36,6% of the Austrian population were online [AUS].

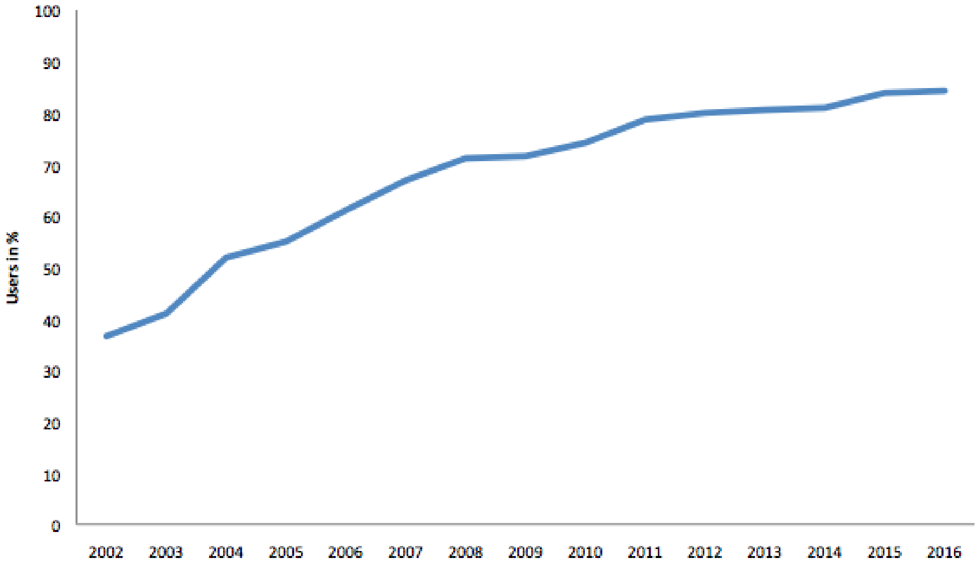


Figure 2.1: Internet Usage Evolution in Austria between the years 2002 and 2016. Data source Statistics Austria [AUS], European Surveys on ICT usage in households and by individuals 2002 to 2016[15].

As a result of the increasing number of users and provided online services, the targeted audiences get broader and more diverse. The audience varies by age, education levels, distinct cultures, as well as various physical or mental disabilities. In addition, changing environments, e.g. smartphones and tablets, have to be considered as well. All these factors have to be considered designing the Web so that no one is excluded. For some users new technologies and the Internet are the only medium to communicate with others, which makes Accessibility of the provided services especially important to them.

[14] http://www.internetworldstats.com/stats.htm

[15] https://www.statistik.at/web_en/statistics/EnergyEnvironmentInnovationMobility/information_society/ict_usage_in_households/041019.html, accessed Jun 2017

The W3C addresses the need for the Web in general and Web Accessibility in particular as part of its definition:

> *Thus the impact of disability is radically changed on the Web because the Web removes barriers to communication and interaction that many people face in the physical world. However, when websites, web technologies, or web tools are badly designed, they can create barriers that exclude people from using the Web.*[16]

Moreover, Web Accessibility not only benefits people with disabilities, but also meets various user needs. Due to the tight relationship with Usability an improved Accessibility also helps with user friendliness. For example, on one of the fundamentals of Web Accessibility - flexibility, which describes a suitable solution for diverse user needs, preferences and situations - targets everybody. With this flexible design in mind, it supports, among others, people having slow Internet connections, people with disabilities as well as elderly people who have age-related limitations [Con16a].

People with disabilities in Austria

According to the *Disability Report 2008* from the Austrian Ministry of Labour, Social Affairs and Consumer Protection there are 630.000 people living in Austria with a strong disability and about 1 Million weaker disabled people [oLSAP08]. The total number of disabled people between 16 and 64 years of age in Austria was about 1,7 Million in 2008, which is about 20% of the Austrian population [oLSAP13]. Since these numbers do not include people over 64 years of age, the target group for Web Accessibility is even larger. The life expectancy of the population is constantly getting higher and so is the number of people using the Internet. Also, this number does not take temporary disabled people, for example people with a broken arm, into account. The numbers in Austria are representative and comparable to those of other countries.

Not considering these people means that more than 20% of potential Web users are left out. Thus, the goal has to be to offer them the best possible Web usage potential. It is not just a moral, ethic and legal argument to include everyone, but also a business argument.

2.2.6 Target Group

Accessible and simple usable Websites are not only beneficial for people with physical disabilities or sensory impairment, but are advantageous for all people. Especially nowadays where people are having a higher life expectancy, digitalization is on the rise and new technologies are introduced constantly.

[16] `https://www.w3.org/standards/webdesign/accessibility`, accessed Jun 2017

It is very important to identify and know the target group before drafting and implementing software. For Accessibility it is especially helpful to be aware of the different barriers that people could face. A good understanding on what such barriers are is essential to identify the problems people with disabilities run into.

People with disabilities

There exists a vast variety of disabilities. Not all of them can be discussed here in detail. In this section an overview of some of the more common ones in relation to Web Accessibility is given.

Visual Impairment

Vision problems are the most common barriers when using computers. There are various eye diseases, that cause different problems and have to be handled differently in the implementation of software. According to the Blinden- und Sehbehindertenverband (BSVÖ)[17] there are, as of 2007, about 318.000, which is almost four percent of the overall population in Austria, visually impaired people.

Debility of Sight

There are different reasons for low vision and they have different affects on the vision of a person. A good overview and insight into the various eye conditions are given by the following simulators:

- `http://www.pro-retina.de/simulation` (accessed Jun 2016)
- `http://www.absv.de/sehbehinderungs-simulator` (accessed Jun 2016)

The most common tools used by low visioned people are inbuilt enlargement technologies.

Blindness

Blind people have a complete or nearly complete vision loss. They are fully dependent on assistive technologies like screen readers, which reads them the content displayed on the screen, or braille displays, which allow them to read with their fingers.

Programmers have to make sure, in particular for Websites, that their code is readable by assistive technologies. A good code and content structure is of high importance.

Color Blindness

Color blindness means the inability or decreased ability to see color or differences in colors. It is a complex disease and can have various effects on how color is perceived by the user. The key factor, when it comes to taking color blind people in consideration for Web design, is that color is not the only method to display information, e.g. colored

[17] `http://www.blindenverband.at/home/wissen/sehen/977`, accessed Jun 2017

links. Moreover, it is important to bear in mind that certain color combinations are known to be difficult for color blind users. Males are more effected than female user, which makes it especially important to be careful with color combinations if the target group of a Website are predominantly male users [fPwD17b]. Overall, according to *Color Blind Awareness*[18], 4,5% of the world population are color blind.

Figure 2.2 shows the different types of color blindness and how affected people perceive the colors. The most common type of color blindness is the *red-green* color blindness (2), where affected people cannot differentiate between red or green. People not seeing any color at all (4) is very rare. Figure 2.2 illustrates the missing color spectrum accordingly and how each single color is perceived.

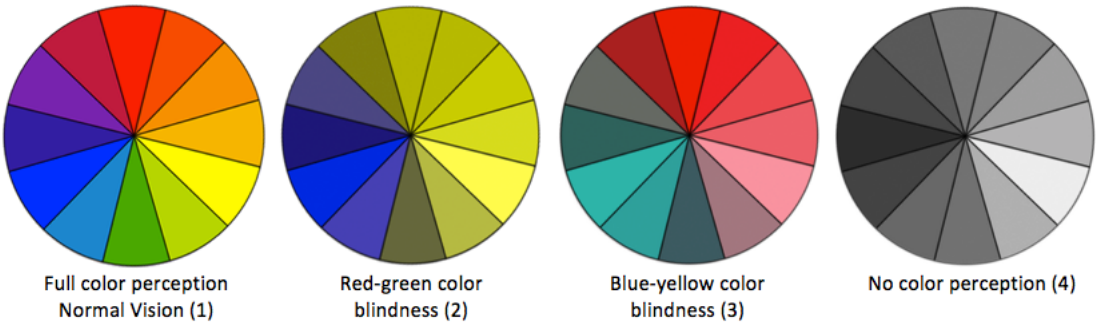


Figure 2.2: Colors as perceived by people with different vision problems[19].

Knowing how color could be experienced by users can guide designers when creating Websites. They should especially take care of this fact when it comes to readability of text. Combinations of front and background color as well as overlays on images have to be thoughtfully chosen. For example, a good compromise for using text on images is an overlay with reduced background opacity. Furthermore, the awareness of color blind users makes it clear that relying on color as the sole source of information is simply not enough. Some widgets, such as color pickers, might need additional labels and links within a text need further features, beside a color highlighting, to be recognizable. These are just a small example set of considerations a design needs to respect. Some tools, e.g. Adobe Photoshop and Adobe Illustrator, provide the means to simulate color blindness and thus allow to check for problems.

Light Sensitivity

Some people are extremely sensitive to light or bright light makes it difficult for them to see. Especially users with low vision are affected by that and often they can get headaches from it. Changing the standard contrast settings is commonly used and the use of a custom color scheme, e.g. invert the colors, is not unusual.

[18] `http://www.colourblindawareness.org/colour-blindness/`, accessed Dec 2016

[19] `http://w3c.github.io/low-vision-a11y-tf/requirements.html`, accessed Dec 2016

Hearing Impairment

Deaf people are often overlooked when it comes to Accessibility. There may be various reasons for that, one in particular is, that people just don't think of deaf people's way to communicate - the sign language[HP11]. Sign language is, especially in Austria, recognized as an own language. Furthermore, it is often the only language that some of the hearing impaired people are talking. Thus, it has to be considered that deaf people might not only be able to not hear video or audio content on a Website, but also that they are not able to read and understand the Website properly. Operators with deaf people in their target group might consider having special sign language videos on their Websites. Generally speaking, it is often sufficient to provide videos and audio with captions.

Limited Mobility

When thinking of people who are limited in their mobility we generally think of wheelchair users. But speaking of Web or software Accessibility and limited mobility, it concerns people who are either completely or partly immobile in their arms and hands. This also affects people who are temporarily affected, be it because they have of a broken arm, are immobile or handicapped in any way. Use-cases for this target group are often forgotten when it comes to software- or Web-design. Especially on mobile devices it is important to think of use-cases involving only one arm or hand. It is important to take mobility into consideration when designing a Website. A Website should not rely on a specific input device, typically a mouse, but should also work with a keyboard only and a variety of other input devices.

Epilepsy

Blinking effects or quickly animated graphics could cause an epileptic seizure for people with epilepsy. Considering this, automated moving effects on Websites should always have a manual pause and play function available at any time.

Cognitive disabilities - Learning Disabilities

When looking into cognitive disabilities there are various known deficiencies. Disabilities such as mental retardation, autism, the effects of traumatic brain injuries, learning disabilities, Attention Deficit Disorder (ADD) and Attention Deficit Hyperactivity Disorder (ADHD) are all considered as cognitive disorders [FB07]. These defects result in limited reading abilities, lower comprehension in general or slower learning capabilities [fPwD17a]. All of them can cause obstacles when using the Web.

Illiteracy

According to the definition of The United Nations Educational, Scientific and Cultural Organization (UNESCO)[20] illiteracy is:

[20] `http://en.unesco.org/`, accessed Dec 2016

> *The ability to identify, understand, interpret, create, communicate, compute and use printed and written materials associated with varying contexts.*

According to this definition there are nearly 17 percent of the world's adult population still not literate.[21]

Even if the situation in Austria occurs to be better, there are, according to *PIAAC-Studie 2013*, still also about 17 percent functional analphabets in the population between 16 and 64 [AUS13]. Functional illiteracy means the ability to read single simple sentences, but the inability to understand a full text or unable to communicate in written form. Assistive technologies, e.g. screen reader software, are often used, but also editorial optimization for the content can be of help, e.g. providing alternatives for "easy to read"-content and also the use of icons or pictograms as supporting measurement for easier understanding [TN05]. Websites should be designed accordingly.

Impaired Ability to Read - Dyslexia

Dyslexia is a reading disorder affecting people with normal intelligence. People diagnosed with dyslexia are having trouble reading and also understanding a text. The number of affected people is varying, but estimations are at about 10 percent of the worldwide population [RBY13]. The main problems dyslexic users face are also affecting other non-disabled people, e.g. confusing page layout, unclear navigation, poor color selections, too small graphics and text or complicated language [MS10].

A simulation that illustrates how a person with dyslexia experiences a written text is given at `http://geon.github.io/programming/2016/03/03/dsxyliea` (accessed Jun 2016)

Rello et al. [RBY15] and the WAI [RBY12] give an overview on how to deal with issues that dyslexic users face, how to avoid them and how software design can optimize the overall user experience.

Old and Inexperienced People

Web Accessibility not only helps people with disabilities, but can also be of advantage for older people [SB09]. An accessible Website also supports inexperienced users since a big part of Accessibility is clear and well-arranged structure of information [PN12].

With increasing life expectancy more and more people are using modern technologies. In addition, modern technology constantly introduces new devices and applications. This is especially difficult for older and inexperienced people to handle. Figure 2.3 shows the increasing percentage of users using the Internet. Table 2.1 gives a more detailed overview on the Internet usage statistic by age in Austria.

[21] `http://www.unesco.org/new/en/education/themes/education-building-blocks/literacy/resources/statistics`, accessed Dec 2016

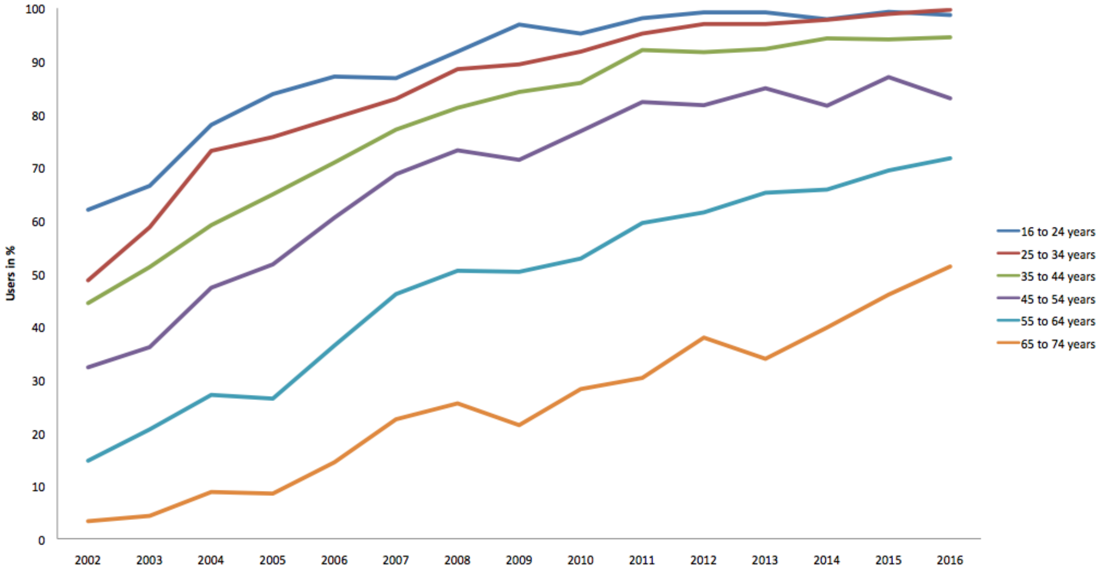


Figure 2.3: Internet Usage Statistics by Age in Austria between the years 2002 and 2016. Data source Statistics Austria [AUS], European Surveys on ICT usage in households and by individuals 2002 to 2016.

2.2.7 Assistive Technologies

For a better understanding on how blind people use a computer a short description on screen reader software is necessary. A screen reader is a software application supporting people with visual impairments during their work on a computer. Simply put, the software tells the user what is visible on a screen. This information may be provided via speech or braille display. The screen reader software does not only support browsing the Web, but provides help using computers in general. Using a mouse is not very effective for blind people and so the screen reader software listens to keyboard commands for interaction. To accomplish this every screen reader software has different keyboard commands predefined. The software supports all common tasks done on a computer. One example is reading, which is in this case done by the software and provided to the user in an alternative form, e.g. speech or braille display. In addition, the screen reader helps with navigating, opening, editing and closing documents among many more tasks.

Year / Age	**2002**	**2004**	**2006**	**2008**	**2010**	**2012**	**2014**	**2015**	**2016**
16 to 24	62,0	78,0	87,1	91,8	95,2	99,2	97,9	99,3	98,7
25 to 34	48,7	73,1	79,3	88,5	91,8	97,0	97,8	98,9	99,7
35 to 44	44,4	59,1	70,9	81,2	85,9	91,7	94,3	94,1	94,5
45 to 54	32,3	47,3	60,5	73,2	76,8	81,7	81,6	87,0	83,0
55 to 64	14,7	27,1	36,4	50,5	52,8	61,5	65,8	69,4	71,7
65 to 74	3,3	8,8	14,4	25,5	28,2	37,9	39,8	46,0	51,3

Table 2.1: Internet Usage Statistics over Time by Age in Austria. Data source Statistics Austria [AUS], European Surveys on ICT usage in households and by individuals 2002 to 2016.

Different screen reader software is available for a large variety of operating systems. A compendious list of screen reader software is available at Wikipedia.[22] The WebAim screen reader survey shows the distribution of the different software among users.[23]

Some of the most common ones are listed here:

JAWS *Job Access With Speech*

Jaws[24] is a widely used proprietary screen reader software for Windows. Keyboard commands are available online.[25]

NVDA *NonVisual Desktop Access*

Is a free screen reader software for the Windows operating system. A full list of commands can be found on their Website.[26]

VoiceOver *VoiceOver*

MacOS has an integrated screen reader software called *VoiceOver*. It can simply be activated on any Apple Computer using the *CMD + F5* keys. Detailed instructions can be found in the full *VoiceOver* commands list.[27]

[22] https://en.wikipedia.org/wiki/List_of_screen_readers, Sep 2017

[23] https://webaim.org/projects/screenreadersurvey6/, Sep 2017

[24] http://www.freedomscientific.com/JAWSHQ/JAWSHeadquarters01, Sep 2017

[25] http://doccenter.freedomscientific.com/doccenter/archives/training/jawskeystrokes.htm, Sep 2017

[26] https://www.nvaccess.org/files/nvda/documentation/userGuide.html#toc27, Sep 2017

[27] https://www.apple.com/voiceover/info/guide/_1131.html, Sep 2017

2.2.8 Legal Fundamentals - Legal Basis in Austria and the European Union

Based on the European Union (EU) and national regulations the Accessibility of Websites has a national legal foundation. National law and EU guidelines refer to the Accessibility guidelines of the WAI/W3C (WAI-Guidelines) [Con16a, Wik15c]. They primarily focus on Web Content Accessibility Guidelines (WCAG) 2.0 (Guidelines for Accessible Webcontent) as technical reference [Con08]. The *OENORM EN ISO 9241-171:2008 11 01* offers ergonomic guidelines and requirements for designing accessible software also takes the WCAG 2.0 as a reference [Wik15b].

OENORM EN ISO 9241-171:2008 11 01

The relevant part of the norm offers ergonomic guidelines and requirements for designing accessible software with interference to the WCAG 2.0 standard [Wik15b, Con08].

Austrian Federal Constitution (B-VG, Bundesverfassung)

Article 7 of the *B-VG* formulates the principle of equality and states a nondiscrimination rule for disabled people [Öst13]. The equality of disabled and not disabled people has to be guaranteed in all areas of everyone's daily life.

> Artikel 7. (1) Alle Staatsbürger sind vor dem Gesetz gleich. Vorrechte der Geburt, des Geschlechtes, des Standes, der Klasse und des Bekenntnisses sind ausgeschlossen. Niemand darf wegen seiner Behinderung benachteiligt werden. Die Republik (Bund, Länder und Gemeinden) bekennt sich dazu, die Gleichbehandlung von behinderten und nichtbehinderten Menschen in allen Bereichen des täglichen Lebens zu gewährleisten.

E-Government Law (E-GovG)

The *E-Government Law* took effect in 2004 and builds a save and legal regulation for electronic communication with public institutions [Öst04]. *§ 1 Abs. 3* states that governmental Websites, which provide information or electronic support for procedures, have to be fully accessible. Furthermore, they have to comply with international standards for disabled people accessing the Web.

> § 1. (3) Bei der Umsetzung der Ziele dieses Bundesgesetzes ist Vorsorge dafür zu treffen, dass behördliche Internetauftritte, die Informationen anbieten oder Verfahren elektronisch unterstützen, so gestaltet sind, dass internationale Standards über die Web-Zugänglichkeit auch hinsichtlich des barrierefreien Zugangs für behinderte Menschen eingehalten werden.

Bundes-Behindertengleichstellungsgesetz (BGStG)

The *Austrian Federal Act on the Equalization of Persons with Disabilities* stipulates the prohibition of discrimination of disabled people and therefore grants people with disabilities equal social living conditions and a self-determined living. The law states the legal consequences of discriminating disabled people.

§ 6 Abs. 5 BGStG defines that not only barriers in the daily life are of importance, but also technical devices and information processing systems have to be designed to be accessible for users with disabilities [Öst05]. That is, disabled people have to be able to use systems in a common way without any special barriers and basically without any help. For the Internet the WAI-Guidelines are the basis for judgment.

> § 6. (5) Barrierefrei sind bauliche und sonstige Anlagen, Verkehrsmittel, technische Gebrauchsgegenstände, Systeme der Informationsverarbeitung sowie andere gestaltete Lebensbereiche, wenn sie für Menschen mit Behinderungen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe zugänglich und nutzbar sind.

European Standard EN 301 549

The *EU Norm EN 301 549 - "Accessibility requirements suitable for public procurement of ICT products and services in Europe"* [EN 14] regulates the requirements for online and offline applications within governmental services on a European level [Wik15a].

European Accessibility Act

As of October 2016 the European Parliament published new guidelines - *The European Accessibility Act* - to regulate Accessibility standards for the Web on a European level [Par16b, Par16a, HM16]. The guideline states, that public services have to guarantee easy access to their Websites and apps for disabled people. The goal is to provide E-Government Services that are usable by everyone including disabled and elderly people.

2.3 Kanban - Software Development Methodology

Developed by Taiichi Ohno for the Toyota production system in the 1950s the *Kanban* concept was originally intended to improve the efficiency in manufacturing facilities. In the early 2000s the *Kanban* method gained popularity in Software Development and was increasingly used in software projects.

The concept represents a way for *Lean Thinking* where new ways of thinking about topics are studied [FMS14]. In general, the methodology of *Lean Thinking* provides a way of thinking that benefits the customer and one of the key principles is to eliminate waste. Translated to the Software Engineering domain these lean principles are defined by Ikonen et al. as *amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity in, and see the whole* [IPF$^+$11]. In the *Kanban* methodology this line of thinking is implemented as a flow of tasks or items.

According to Ikonen et al.[IPF$^+$11] the basic key activities in the Kanban method are:

Vizualize the Workflow The overall work is split into single pieces. Each of these pieces, the work items, is written on a card and visualized on a wall (see Section 2.3 Kanban Board). The different states of the workflow, and thus the states an item can appear in, are represented by named columns.

Limit Work In Progress (WIP) The number of items in progress at each workflow state are explicitly limited.

Measure the lead time The lead time refers to the average time to complete an item. The goal is to optimize the workflow to decrease the lead time to a minimum. This enables the team to make predictions about items and time accordingly.

Kanban Board

Typically the *Kanban* method uses a card wall - the *Kanban Board* - to visualize the work items [FMS14]. This board acts not only as a visual control element, but also supports the team in keeping an overview on the current workload and the status of the process.

To provide such an overview the board is organized into three viewpoints *Time, Task, and Team* [Hir07]. This is accomplished by structuring the board vertically, hence as columns, in different working states [Hir08]. These working states represent a specific state an item is in. The simplest way of organizing these states are *Todo, Doing, and Done*. However, the number of working states can be adjusted and expanded according to the needs of a project. This means, that *Kanban Boards* differ based on a project or the team that is working with it. The work items, i.e., the tasks a team has to work on, are represented by cards. Overall, this visualizes the Work In Progress (WIP) of a project.

An example for a simple *Kanban Board* (source Wikipedia[28]) is shown in Figure 2.4. The illustrated board only uses three vertical separations for tasks.

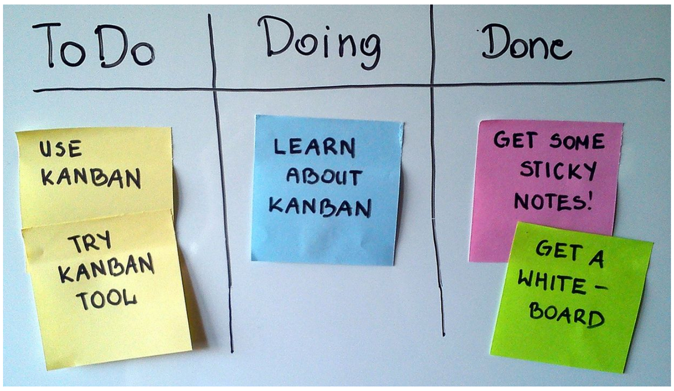


Figure 2.4: Simple Kanban Board

In the *Kanban* principle it is essential that a specific state of the workflow does not contain an unlimited number of work items (tasks). That is, each column of a *Kanban Board* should only contain a limited number of cards at any point in time. Thus, each state only has a limited number of work items assigned to them. This allows an easy overview of the WIP. Each card, and thus each task, has a member of the team attached so that it can be identified who is working on the task. By moving cards along the columns, hence changing the state that they are in, the (work)flow of the project is illustrated.

[28] Image from Wikipedia `https://en.wikipedia.org/wiki/Kanban_board`, accessed Oct 2017

CHAPTER 3

Empirical Research: Interviews

For this thesis interviews with experts in the Accessibility field have been conducted. The idea was to get a better understanding and insight about accessible Websites, the work practise and on how practitioners approach the concept of barrier-free Websites.

On the one hand the interviews should help to find out whether the experts face the same issues and problems in their experience as described in Section 1.2 of this thesis, and on the other hand to identify how practitioners handle the challenges in their daily work. The interviews aimed at providing insights on the practitioners approach on how to develop accessible Websites. The interviews revealed challenges that practitioners face and also provided a chance to present the idea, an integrated process with an emphasis on Usability and Accessibility, of the thesis. The feedback and valuable input from experts working in the field helped to refine the approach of this thesis.

3.1 Basics & Methodology

This section provides an overview on the interview method and how the interviews have been conducted. A description of the interview partners is given and the interview guideline is described in detail.

3.1.1 Method

An open interview guideline was used to lead through a semi-structured interview. Questions were prepared in a guideline, which outlined the rough structure while performing the interview. The prepared questions are of an open type and no possible answers are provided. The question guideline acts as a leitmotif for the interviewer and structures the interview. It helps to guide the interviewer and to keep focus on the questions.

Interviews are a common method used in small-scale research and in this thesis the focus is on the semi-structured interview technique. Although this technique is not suitable for a large number of participants, it is perfect for the purpose of this thesis where only a small study was performed. The flexibility of this interview technique allows to explore the views and opinions of the interview partners [Dre95]. The interview itself does not follow a strict sequence of questions, but rather is open and facilitates the freedom to direct interviews in slightly different directions each time. This enables the interview partners to be more flexible and express themselves more freely. In addition, it helped them to propose new ideas and provide valuable input for this thesis.

As mentioned above the interviews should help to get a better understanding of the current situation and get insights on the opinions of practitioners working in the business. The questions targeted the current status in the Accessibility community and how Accessibility is perceived. Another important aspect of the interviews was to identify what the interview partners notice in their daily business and what problems they face. Further questions focused on how they structure their work and how Usability and Accessibility is handled in their various projects. These questions targeted specific parts of the proposed process of this thesis. Although the approach of this thesis was not the sole focal point of the interviews, some of the questions aimed to get input and thoughts on the proposed process.

The main areas of interest for this thesis were defined as:

1. *Profile* - to get general information about the interview partner and set the atmosphere for the interview
2. *General Perspective* - to gain information about their perception of the topic
3. *Projects and Workflow* - project specific and workflow questions
4. *Expense factor* - to compare the cost-advantages a structured process would yield
5. *Rules and Guidelines* - to find out how the interview partners work, what guidelines and tools they use

3.1.2 Interview Partner

For the semi-structured expert interviews four people from the Viennese Accessibility industry, who are well-known in the sector, were chosen. Most of them are part of the Accessible Media[1] association and all of them are still working, mostly as project managers and/or consultants, in the sector.

Additionally two interviews with software developers were conducted. Here the focus was on getting a better insight in their perspective on the topic, what kind of tools they use and what guidelines they follow. The developers are working for a Web-agency with strong focus on Accessibility.

[1] http://www.accessible-media.at/, accessed Mar 2017

Generally speaking, the goal was to identify process models and workflows that are already in use. In addition the interviews should help to see how Accessibility can be integrated into a development process. Since developers are one of the main groups responsible for integrating and implementing Accessibility it was interesting to see their perspectives, how they feel about Accessibility and what their insights are.

Experts

Interviewee A Founder and Managing Director

The interviewee's company is a Web-agency specialized on accessible Web development. Interviewee A, as one of the founders of the company, is still handling project management for development projects as well as doing consulting in the Usability and Accessibility field. Interviewee A is working in the Web industry since 2005 and has had a strong focus on Accessibility ever since. The interview took part at the interviewee's office in Vienna on February 13, 2017.

Interviewee B Founder and Managing Director

Interviewee B's company is specialized on Web development with a strong focus on Accessibility. The Interviewee is also working as an Accessibility and Usability Consultant since 2003. The interview took place at the interviewee's office in Vienna on February 23, 2017.

Interviewee C Project Manager

Interviewee C is working for an online agency based in Germany and Vienna. The agency operates in various fields within the Web industry. Since they have clients in the public sector they are also specialized, respectively guarantee a special level of, Accessibility. Interviewee C works in the e-business since 2011. The interviewee's focus is not strictly on Accessibility, but in his role as project manager he is involved in every project step and oversees the Accessibility responsibilities as well. The interview took place in Vienna on February 2, 2017.

Interviewee D Lead Portalmanagement

Interviewee D is the lead of a regional Web portal maintained by the City of Vienna. The portal offers various information and services for the Viennese population. The interviewee is responsible for the sites to get involved with Accessibility and fulfill certain levels. Interviewee D is part of the company and the Vienna Accessibility scene since 2001. Personal communication took part via Email on February 20, 2017.

Developers

Interviewee E Web Developer

Interviewee E is a Web developer since nine years and specialized in Accessibility since 2016. He works as a frontend developer at a Web agency that is specialized in Web Accessibility. The interview took part in Vienna on February 17, 2017.

Interviewee F Web Developer

Interviewee F works as frontend developer in a Viennese Web agency with strong Accessibility focus. The interviewee is working in the Web development field since six years and is specialized in Accessibility since 2015. The interview took part in Vienna on February 20, 2017.

3.1.3 Interview Guideline

The *Interview Guideline* outlines the questions for the interviews.

Profile

1. What is your role in the company?

2. When did you start focusing on Accessibility? For how long are you in the industry?

3. (Developer) Did you have any special Accessibility training?

4. (Developer) How much time did it take you to develop yourself from a general software developer to an Accessibility developer?

General Perspective

1. (Consulting) What do you think is the main motivation for your customers to ask for your guidance?

2. In your opinion, what are the main reasons why Accessibility is often left out? Why do you think is it implemented so rarely?

3. Is there a way to counteract the existing problems? Do you think there could be a standardized way to develop (almost) fully accessible and usable software without a lot of additional effort?

4. Does Accessibility (and Usability) stand in a designer's way? That is, can good usable Websites be beautiful?

Project Specific & Workflow Questions

1. Are the developers you are working with doing any special training for Accessibility?
    a) How long does it take to train someone accordingly?
    b) How do you ensure that your stakeholders (developers) take Accessibility into account?
2. In your opinion, how can Accessibility be positively influenced within a project?
3. When / In which project phase should Accessibility or Usability measures be applied?
4. Who, or which role within a project(team) should have the responsibility for Accessibility, or also Usability? (Developer, Project Manager, Experts,...)
5. How can the Accessibility specifications be ensured during project management?
6. How should Accessibility Guidelines be specified to be systematical processed by each stakeholder (Designer, Developer, Editor)?
7. How could a standardized process or specification with an emphasis on Accessibility look like? How is your internal procedure? Do you think Accessibility first, meaning, do you consider it right from the beginning?
8. How do you guarantee Accessibility? Are you making any final testing? Is continuous testing used?

Expense Factor

1. Does Accessibility / Usability need a lot of extra effort and thus costs?
2. How high would you say is the extra effort for a supplementary integration of Accessibility into an existing project? How high are the costs compared to the effort it takes to consider the integration of Accessibility right from the beginning?

Rules & Guidelines

1. What Accessibility methods are available? How do you work? What tools are you using?
2. How understandable are the WCAG guidelines in your opinion?
3. According to literature the WCAG guidelines are partly out-dated and should be revised. They also contain criteria, which are not relevant any more. How do you see the WCAG? What is your opinion, from your practice?

3.2 Analysis

The theoretical basis of the empirical research is discussed in this section. After a description of the used methodology for the analysis, the implementation of the approach is demonstrated.

General Approach

The interpretation of the data gathered in the interviews is based on Philip Mayring's approach for *Qualitative Content Analysis* [May14]. He distinguishes the following three analysis techniques:

Summary The goal is to reduce the complexity of the data through abstraction and extracting the essential content.

Explication Additional material on unclear text parts has to be defined. This helps to increase understanding and interpretation of the text.

Structuring Pre-defined criteria are used to filter out particular aspects of the data.

For this thesis the *Summary* approach seemed to be a sufficient way of analyzing the data since the topic is well differentiated already. Making annotations appeared unnecessary and finding pre-defined categories seemed not appropriate for the amount and character of the data.

The *Summarizing* technique reduces the given content to core aspects within the following seven process steps:

Step 1 Define units of analysis

Step 2 Paraphrasing

Step 3 Generalization

Step 4 Reduction - through erasing semantically identical paraphrases

Step 5 Second Reduction - through binding and integration of paraphrases

Step 6 Collection of the new statements in form of categories

Step 7 Re-testing the new category system

Note that steps 2 to 5 can be combined for large quantities of data.

Practical Implementation

The actual proceeding of the interviews in this thesis was conducted in the following manner. The interviewees where all contacted via Email. A short introduction on the aim of the contact was included in this Email. All interviews, except for one, took place in Vienna in the participants offices for convenience reasons. Due to time limitations one of the interviews was conducted via Email and on the phone. The interviews where recorded via digital dicta-phone and lasted between 30 to 40 minutes. In general the interview guideline was followed, but in individual cases the order of the questions and their discussions differed and led to various side notes.

Upon transcribing the recordings the content was clustered accordingly in Microsoft Excel to grant a structured review and thus be able to apply the analysis approach. Applying the first step *Paraphrasing* results in a new column, where all text components, which do not add to the content are removed. The resulting content was transposed and transformed into an abbreviated form. Subsequently the *Generalization* took part where paraphrases are rephrased and abstracted to formulate new ones meaning the same. Semantically identical paraphrases are reduced by combining similar ones and removing not substantial parts within the *First and Second Reduction.* In a further run of the steps, the result was further reduced and limited. The finally resulting set of statements - categories - was compared with the original set of data.

Taking the goal of the interviews - acquiring insight and input into the Accessibility field - into consideration, a deeper analysis of the content was omitted. The conclusion of the analysis was used to refine the proposed process and framework.

All expert interview quotations throughout this thesis have been translated from German by the author. The full transcripts of the omitted interviews can be found in the Appendix 6.

3.3 Evaluation & Findings

The interviews showed that there is no strict way or process that the experts follow to guarantee a proper integration of Accessibility. They all have their own methods, mainly based on experience, but none of them have a stringent workflow they follow.

All of the interviewees agreed that there is a lack of Accessibility implementations and awareness in the Web. This also corresponds with recent research [AB16]. A lot of Websites do not consider Accessibility at all and, from the experience of the interviewees, if Accessibility is taken into consideration then it is primarily because of legal requirements. Even, if in theory legal demands affect everyone, there are still many companies ignoring it. Apart from the public sector only a minority of companies see the need to consider taking any actions and even less are in fact making any arrangements. The following quotes provide some insight on the lack of Accessibility awareness:

> *The majority of my clients would rather invest in visually pretty features than in Accessibility.* (Interviewee C, p. 142)

Interviewee D answered the question on the general Accessibility implementations with the following statement:

> *It is only considered by the ones who are forced to take Accessibility into consideration, otherwise it is barely considered. Legal demands are helping though [...]* (Interviewee D, p. 165)

Interviewee A noticed strong swayings in the past twelve years and he experienced better and worse Accessibility awareness trends. Though, recently it appears to be a negative peak again:

> *[...] there were cycles, where the awareness was growing, then it got setback [...]*
>
> *Now there appears to be another setback. One reason might be big Web front-end frameworks, like for example AngularJS, that do not focus on Accessibility. These frameworks provide the means to easily create Websites and thus good development skills or specific front-end developers are not essential [...] which results in terrible and not accessible Websites.*
>
> (Interviewee A, p. 160, 161)

Answering the question on the general implementation of Accessibility in the Web, Interviewee F clearly states:

> *In general it could be much better, not even Websites of big companies are accessible.* (Interviewee F, p. 158)

These quotes show, that the interviewees experience a lack of Accessibility awareness, not only by talking to their customers, but also in their perception on how they observe the Web themselves.

But this is not only a problem of the companies. The interviews also point out, that generally there is a lack of knowledge when it comes to awareness. Not only awareness by companies or people who should feel responsible for taking Accessibility into account, but also of the developers themselves. In addition, many developers are not well trained in the Accessibility field. Developers who know and understand the problems and who are able to properly implement the additional Accessibility requirements are still a minority.

This lack of knowledge exists, although, big influencers like Google or Microsoft are investing more and more into good Usability and also taking Accessibility initiatives. Specifically, common browsers nowadays have already a lot of features integrated (see

Section 5.5.3) and also a lot of best practices on how to write proper markup for Accessibility exist. In addition, many large companies have campaigns to increase the awareness of developers and users. However, often they do not implement all their Websites in an accessible manner. The perception of pushing the importance of Accessibility was mentioned more than once during the interviews. Interviewee F explains this shift of the big companies as result for their own benefit, which is also a shared opinion between the interview partners.

> *The big players, like Google and Microsoft, are currently extremely forcing the topic. [...] Machine Learning is getting more attention and gains importance, and clear structure and a correct mark-up, advances the Accessibility, but also helps machines to understand the code and indexing the sites gets easier.* (Interviewee F, p. 157)

According to Interviewee C, one of the main reasons for the lack of knowledge seems to be the ignorance of the educational institutions. The developers skilled in the field did not learn it at universities or in courses, but rather did the research themselves online. In school or university courses targeting Web-design the topic of Accessibility is usually not covered. In addition, the courses that give attention to Accessibility unfortunately do not attract a lot of people. So, it really is a matter of pro-activeness and the developers own initiative. And precisely that often seems to be the problem as Interviewee B explains. Missing motivation for the topic by developers is causes concerns. In Interviewee B's opinion there is already a lot of knowledge developers have to gain themselves - developers already have their own emphasis on topics and they mostly learned them via self-studying. Hence, Interviewee B sees developers already as experts in their areas and mostly these areas do not include Accessibility. Interviewee B added that in his view developers simply do not see the necessity for Accessibility.

> *[...] developers all have their particular focuses, e.g. JavaScript with its various Frameworks [...] they don't see the need or better, they don't demand to become experts in Accessibility.*
>
> *[...], they know what it is about, but they don't claim to become experts in the area. Actually, they do not see the necessity for learning Accessibility in their daily work.* (Interviewee B, p. 152)

Interviewee A's view is even more radical. It takes a lot of effort to gain the knowledge about Accessibility and knowing just the basics is often not enough as Interviewee A's statement shows.

> *Smattering is not enough when it comes to Accessibility.*
>
> (Interviewee A, p. 161)

Anyway, Interviewee A does not seem to have strict requirement specification for developers, but rather is trying to spark the developer's motivation through including them in the entire development process. The developers in Interviewee A's company regularly attend user tests, which helps in their understanding and awareness for Accessibility.

Interviewee A forces early test cycles to guarantee a proper Accessibility implementation. Also, instead of having detailed specifications Interviewee A's strategy is to have a so-called *Semantic Map*[2]. This *Semantic Map* utilizes wireframes to visualize the heading structure and the basic elements of the software. Common known Accessibility issues, which Interviewee A experiences regularly, are especially mentioned in an additional document, or simply as concrete tasks for the developer.

Having proper specifications, though, seems to help developers as stated by Interviewee B:

> *Accessibility demands need to be specified in the functional requirement specification and that also helps the developers.* (Interviewee B, p. 153)

Moreover, the interviews clearly showed that the WCAG standard is a good basis, but need a lot of knowledge in advance to understand them. The interviewees shared their critical opinion on the guidelines. For all interviewees it is necessary to have a standard, but it cannot directly be used as guidelines or best practices for developers. For the experts the WCAG defines the basis for their work, but talking to the developers also showed that this form of guidelines is not practicable for them. It takes effort combined with extensive background knowledge to understand the guidelines. Or, as Interviewee E states, specifications are meant to be for experts and that is how they are written.

> *Specs are usually very technical and written for experts - it is definitely necessary to really engage with the topic.* (Interviewee E, p. 151)

However, writing proper markup and semantically correct code has a lot of advantages and slowly gets higher attention - not only by developers, but also by customers. Even if the main reason for this might not be Accessibility per se, but rather technological advances such as Search Engine Optimization (SEO). Interviewee B explains it as follows:

> *Accessibility has a strong link to search engine optimization and to Usability. That means an accessible site is automatically better usable and is also automatically search engine friendly.* (Interviewee B, p. 154)

Interestingly Interviewee C had a similar view. Interviewee C's perception is alike and he also mentioned the SEO advantages. Also he feels, that the recent trend to search engine optimization affects the Accessibility in a positive way. The proper markup, which is an

[2] Translated from German by the author

important condition for SEO, advances the Accessibility. Even if that is just an indirect achievement, it still helps creating better and more accessible Websites.

When talking about the process and a possible integration of Accessibility into the whole life-cycle of the project, the interviewees agreed on the necessity of it. For projects with high Accessibility relevance some of the experts are already trying to realize a similar process, but they are still lacking any clear definitions or specifications. The strictness on how the implementation and the pre- and post-phases are executed differs between the interviewees. Interviewee D for example already utilizes testing iterations:

> *For large projects we apply several Accessibility test iterations, starting with the concept, then the design and later the implementation.*
>
> (Interviewee D, p. 165)

Some of the interviewees are already trying to specify clear Accessibility requirements. Others who don't have clear Accessibility requirements yet like the idea of doing so. Interviewee C states:

> *[...] These non-functional requirements (relating to Accessibility requirements) for each implementation module already defined in the requirement specification - that would be really cool. We don't do that, but it would be helpful.* (Interviewee C, p. 146)

Also the developers favor the idea of having clear specifications. It seems like they usually don't get the requirements accordingly presented and doing so appears to be advantageous for them:

> *It would be good to have best practices describing what should be taken care of concerning Accessibility.*
>
> *[...]*
>
> *Concrete requirement specifications would be nice, or at least kind of a reminder - like a checklist.* (Interviewee E, p. 150)

Altogether, the interviews clearly showed that a structured approach would be an advantage. There is no defined process on how they proceed and also not really any tool support available. Accessibility relevant requirements, if specified at all, are defined project-based. Having a framework providing pre-defined reusable general rules for different projects would support all involved parties. Developers would particularly welcome a standardized, yet still understandable way of having the demands and requirements of a project presented to them.

Another aspect taken into account for the interviews was the measurement of the costs for Accessibility and the effort it takes to implement proper Accessibility requirements. The focus here was in the distinction between an accessible implementation upfront compared to a later integration of Accessibility. The answers from the interviews where all going in a similar direction and stated that it always depends on the complexity of the project. In general, though, the interviewees all agreed that a proper integration of Accessibility from the beginning is definitely an advantage. Or, as Interviewee E states more concretely, considering Accessibility from the beginning and having clear specifications helps a lot and minimizes extra effort.

> *[...] in the best case the Accessibility requirements are given within the specifications, which guarantees the implementation of them, and thus results in none or only minimal additional effort.* (Interviewee E, p. 150)

Concluding, the interviews state, that a defined process which considers Accessibility demands right from the beginning seems to be a valid approach. In addition, having a process integrating an Accessibility requirement specification is desirable. Providing a framework, which on the one hand helps the project manager or the person defining the requirements, and on the other hand also acts as a checklist for the developer, would assist in many ways. Combining the framework with the WCAG is an advantage for the project managers as they already use the guideline as a basis. Providing further implementation tips and input supports the developers, who often lack the specific Accessibility knowledge, helps them to deal with the complexity of the WCAG. Overall, such a process would help in creating better and more accessible Websites.

CHAPTER 4

An Integrated Process and a Supporting Framework

One of the essential parts of this thesis is the definition of a software development process with a focal point on Usability and Accessibility. This process should support the development team during the entire life-cycle of the project and bring Usability and especially Accessibility into focus in each phase. Software Engineering and project management methods are used to define the overall process. An special emphasis is given to the needs of the target group and thus Accessibility. These requirements are formally specified and considered in each phase and iteration during the project life-cycle. In addition a framework as a possible implementation of the process is proposed. The proposed process was refined based on the results of the expert interviews (see Chapter 3) and research of related work.

4.1 Process Description - Overview

The process is defined in dependence of general agile Software Engineering process approaches [Hou14]. A mixture of various proposed approaches have been studied and chosen so that a sufficient and satisfying result can be achieved.

Its phases are based on standard agile Software Engineering techniques, but differ from them as each phase has an distinct emphasis on Usability and Accessibility. At first this seems to introduce a lot of overhead and extra effort, but in reality it helps the development team as they get more accustomed with it. The additionally available *Accessibility Toolbox* for developers described in Chapter 5 provides further assistance. Thus, the proposed process should have a high impact on the users with minimal additional effort for the development team.

The proposed process is structured into the following phases and a detailed description is provided in the following sections:

- Vision, Research & Plan
- Concept, Specification & Requirements
- Design
- Implementation
- Testing (manual and automated)
- Deploy & Release
- Continuous Quality Assurance & Monitoring
- Launch (final)
- Further development and enhancement

Even though this enumeration suggests a linear approach, an iterative approach is eligible. In an iterative approach, phases, which require more work and time, should be repeated as needed. The proposed process is visualized in Figure 4.1 and a detailed description of each phase and the proposed workflow is given in Section 4.3.

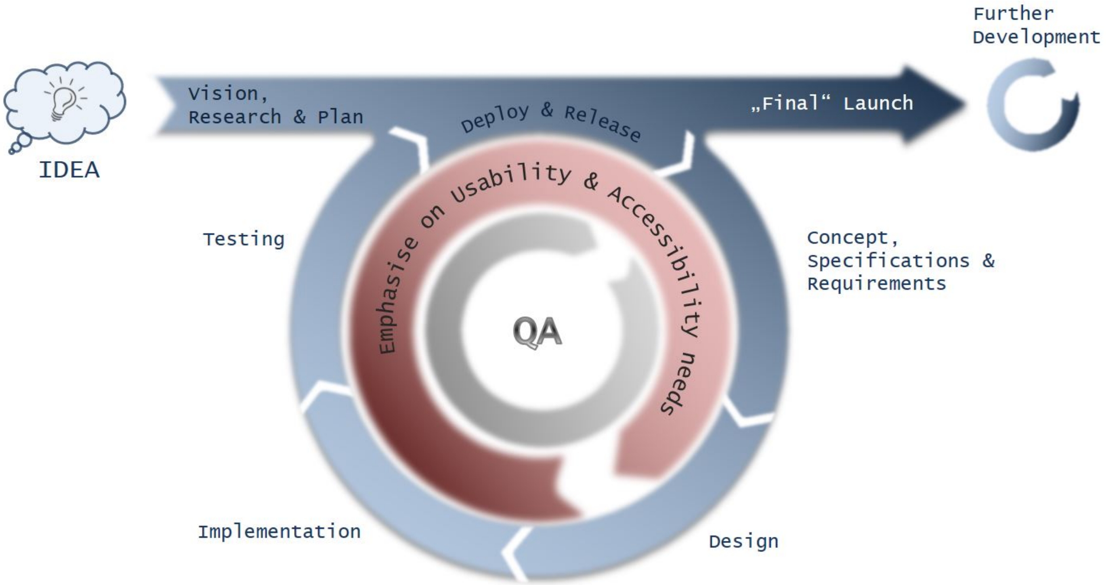


Figure 4.1: Proposed process life-cycle with an emphasis on Usability and Accessibility, based on general agile Software development [Hou14]

At the beginning of each Web project is an *Idea* and this initiates the process. The idea, be it a new Website entirely or additional features for existing Websites, leads to a vision on how it should and could be integrated. A clear understanding of this idea and how the vision can be accomplished, is performed in the *Vision, Research and Planning* phase. The result of this phase is a specification of the goals. As this phase is not within the overall process loop it is only executed once at the beginning of a project. That is, the vision and main goal of the project do not change throughout the process, but the requirements may be adapted and refined in following phases and iterations. An important part of this phase is the commitment to keep the users of the target group in focus. To ensure this, all involved stakeholders should be informed accordingly and have an understanding of the Accessibility background.

With the goal of the project defined, the specification of a rough first concept and requirements can start. The concept that is specified in this *Concept, Specification & Requirements* phase can and will be refined with each subsequent phase and iteration. This also applies to the requirements, which are subject to change, and they may be adapted in consequence to previous iterations. An important part of this phase is the definition of the basic Usability and Accessibility requirements. The planning of specifications and requirement has to include the actions that are needed to achieve the targeted level, e.g. AA-level, of Accessibility. The result is a clear and comprehensive specification addressing the user needs of the target group.

Based on the result of the *Concept, Specification & Requirement* phase, a first design draft can be evaluated in the *Design & Prototype* phase. In this phase a design manual providing the basis for the look and feel of the product is created. The design manual has to consider and address all the Usability and Accessibility requirements. The design related Accessibility requirements should be set as early as possible, this means in one of the earliest iterations of the process cycle. The necessity for this is that the Accessibility requirements are preconditions for many design decisions. Developing wireframes and a prototype allows to conduct early user tests and thus to receive feedback, but also serves as a basis that the developers built upon. Visualizing the requirements and the workflows not only helps the developers to understand the overall concept, but also allows to create a prototype for first user tests. Identifying early on, via user tests, if a concept holds or fails is crucial.

In succession to the requirement specifications, design guidelines and the prototype the *Implementation* phase starts. During this phase the developers start with implementing the project. It is mandatory that the developers have a basic understanding of Accessibility. However, they do not necessarily need to be experts as the specifications should already contain all relevant information and requirements.

After the defined components and requirements have been implemented, they have to be tested in the *Monitoring & Test* phase. A, during the *Concept, Specification & Requirements* phase, predefined test-plan is used to check whether the functionality and the Accessibility requirements are fulfilled. The test-plan should define function tests as well as Accessibility measures that can and have to be checked within a particular component. Some of these tests are implemented as automated unit tests while others require manual testing. Manual tests are especially needed and helpful if the situation makes it necessary and the test cases can not be automated. This is already defined within the test plan. In case of failing test cases, the software parts can not be released. If so, an additional development cycle is necessary. Assuming all tests are positive and none of the test cases are failing, the software component can be released and deployed. Smaller releases will be deployed until a feature or the whole Website is ready for *Launch*.

As mentioned before, each release deployment allows the specification and the requirements to be updated and refined. This is necessary since requirements often change over time as the project team gets a better understanding of the topic at hand. In this case the specifications are adapted and a new iteration is initiated.

During each phase of the process *Quality Assurance* is of high importance. Therefore continuous testing as well as constant monitoring are indispensable during the project life-cycle.

After the final *Launch* further development and enhancements can be started. These additions to the software are handled via *feature requests* or a new enhancement process. To ensure the same quality as with the original project, each of these requests should follow the same process and at least iterate through it once.

4.2 Stakeholder

The key to an efficient process is to integrate all stakeholders, have discussions and to find a viable solution together. Not every project will involve all stakeholders, e.g. small feature requests might only need a small subset of stakeholders, and for big websites or sites with very high standards there could be multiple people occupying single roles. Even if not all roles are used, it is still important that someone is in charge of the tasks and responsibilities of each stakeholder.

Product Owner Customer or Product Owner

The *Product Owner* is commonly the initiator of a project. Thus she/he is one of the most important stakeholders overall, and also when it comes to Accessibility. The *Product Owner's* commitment to Accessibility is the key element and without it Accessibility won't be a focal point of the project. Also if the *Customer* does not see Accessibility as a major part in the project, she/he won't be willing to accept the extra effort and restrictions Accessibility could demand. As the *Product Owner* initiates the project she/he has to decide right from the beginning whether the product should be usable and accessible for everyone.

Project Manager Project Manager or Project Lead

The *Project Manager* coordinates the project and the team. Essentially, she/he is responsible for the outcome of the project and that quality standards for the product are adhered. This does not necessarily mean that the project manager is in charge of testing and/or that she/he is an expert in Accessibility, but rather that she/he is responsible to coordinate all necessary tasks. Though, for small projects or small teams, she/he could also take the role of an *Accessibility Expert.* However, basic knowledge about Usability and Accessibility should be present. Important is the commitment to and interest in Accessibility, otherwise the integration of Accessibility will fail. If the project lead does not stand behind the ideology and takes care of it, then it will be lost during the software development process. If problems occur the *Project Lead* is responsible to reiterate and advocate the advantages a good Usability and Accessibility yield.

Requ. Engineer Requirements Engineer and/or Business Analyst

The *Requirements Engineer* supports the *Project Manager* during the *Vision, Research & Planning* phase. If a project is large enough to demand a dedicated *Requirements Engineer*, then this stakeholder is responsible that the result of the *Concept, Specifications & Requirements* phase is created accordingly. For many projects this role is optional and often this stakeholder position is performed by the *Project Manager.* It is of high importance that the results of the requirement specifications already contain all necessary Accessibility demands.

Designer Working with a dedicated *Designer* is necessary for almost all Web projects. Sometimes Corporate Design (CD) or Corporate Identity (CI) is already available and in this case this stakeholder is devalued. But for most projects a *Designer* is a must. In terms of Usability and Accessibility it is of high importance that the *Project Manager* and the *Designer* work closely together. Since Accessibility demands concerning the design have to be targeted in the *Design* phase, it is necessary that strict specifications are already in place from the *Concept, Specifications & Requirements* phase. It is, for example, necessary to have color definitions with specific contrast values, because otherwise an accessible solution will not be achievable. The main goal for this stakeholder is to find a design where all other stakeholders are happy with. Most importantly the *Product Owner* has to agree with the design and the design has to be functional and usable.

Programmer Frontend and Backend Developer

This stakeholder is responsible for implementing the requirements and functions accordingly. Since the requirements already target Accessibility demands, this stakeholder should have a basic understanding of the area, because otherwise the implementation will fail. However, it is not mandatory for them to be especially well trained beforehand. Of higher importance is, that the specifications are clear and detailed enough so that the *Programmers* know exactly what is expected of them. It has to be mentioned that this does not mean, that the requirements need to show all cases and how they have to be implemented, but rather that all requirements are specified clearly and all necessary information is available.

Test Engineer The *Test Engineer* is in charge of testing the implementation. By doing so she/he guarantees a certain quality standard. Based on the requirements and the specified test cases the *Test Engineer* tests the product. Automatic testing can be used as well as manual testing. Some Accessibility and Usability requirements cannot be tested automatically and thus have to be tested manually. The *Test Engineer* should already be included in early stages, during the *Concept, Specifications & Requirements* phase, of a project as her/his task is to specify corresponding test cases for the requirements. She/he also works closely together with the *Expert* stakeholder. It is important to note that the *Test Engineer* is not the only one testing the application and assuring quality. During the entire process life-cycle automated testing should be carried out. Thus, also the *Programmer* stakeholder is a big part of testing and continuous integration. Only together a good level of quality can be guaranteed.

Expert Usability and/or Accessibility Expert or Consultant

For large projects with high Accessibility demands it is necessary to include an *Expert* or *Consultant.* This stakeholder should provide insight into Usability and Accessibility and also help with advice or answer occurring questions. Generally, this stakeholder works closely with all other stakeholders beginning in the *Vision, Research & Planning* phase. In addition she/he provides support for the *Programmers.* For Accessibility projects this stakeholder may be also responsible for Accessibility testing.

Content Manager Web Editor or Content Manager

One stakeholder who is excluded quite often is the one responsible for the actual content. A perfect system can still be used in a wrong way and ignore all the Accessibility features that are implemented. Using a Website and providing content properly is also an essential part of the process life-cycle. If the content is not edited and added properly the construct of structured designing and programming will fail. Thus, it is crucial to include the *Content Manager* and teach them accordingly.

4.3 Project Phases in Detail

In this section the phases of the proposed process are discussed in detail. For each phase a description of what is done, what methods are used and who is involved is given. In addition, the results of each phase are specified in detail and also a checklist for tasks to be completed is given. These checklists can be interpreted as a guideline for new projects. Although the process is well-defined it also allows some flexibility in handling tasks.

4.3.1 Vision, Research & Planning

Each project starts with an initiative because someone has the need for something new - in this case we focus on building a new Website. Therefore the process starts with an idea where a *Product Owner* takes the initiative and presents her/his vision of a new Website with focus on Accessibility to a *Project Manager*. The distinction between the *Product Owner* and the *Project Manager* is that the *Product Owner* initiates the project, whereas the *Project Manager* coordinates the development of the project. One of the key factors is, that the *Product Owner* values Accessibility highly. If she/he does not show commitment to Accessibility, it will be hard to implement and argue for it. After the *Product Owner* presents her/his idea, the actual project can start with a kick-off. For the sake of this thesis the bureaucracy and business requirements for, e.g. contracts are not taken into account.

After the kick-off the *Project Manager* and the *Product Owner* start to plan the upcoming project. This *Vision, Research & Planning* phase is the basis for all other subsequent phases during the development cycle. Since there is a high dependence on the needs of the *Product Owner*, she/he is strongly incorporated. This phase is actually the one with the most input from the *Product Owner* as it is essential to clearly define the scope of the project.

There are various methods available that can be utilized in the *Vision, Research & Planning* phase. First of all it is important to clearly understand the *Product Owner* and her/his vision. Interviews are a good way to identify what the *Product Owner* has in mind and help to get a clear picture. Focus groups, Brainstorming or the Design Thinking method [UBP^{+}15] can also be used to analyze the problem, get new inspiration and implementation ideas. All this methods aim at getting a clear picture of what should be developed.

In some cases an existing Website should be redesigned. This scenario differs, although not drastically, from starting from scratch. In this case the existing Website can be used for evaluations and help the *Project Manager* to get a clearer idea about the content and the target users.

The *Vision, Research & Planning* phase does not necessarily involve a lot of detailed Accessibility semantics. However, it is mandatory to define Accessibility as a goal. The steps during this phase are usually fixed and not dramatically changed in process iterations. That is, the idea and the vision persist throughout the process and are only refined.

4.3.2 Phase I: Concept, Specifications & Requirements

Based on the results of the *Vision, Research & Planning* phase the *Concept, Specifications & Requirements* creates a clear specification with requirements of the software to develop. This phase is essential for achieving the Accessibility goals that are set, since it provides the specifications for all subsequent phases and iterations. Various needs have to be considered, which makes this a rather critical and important phase.

Evaluating the current site or other similar Websites, results in a list of rough requirements. These requirements can be differentiated into *must-haves* and *nice-to-haves*. Another aspect are the main focal points or hard requirements set by the *Product Owner*. In addition, there might be problems or issues of any aspect, which should be resolved with a relaunch of the Website. Considering all these aspects leads to a first concept draft with the main rough requirements not yet specified in detail.

Get to know your users

Before going into more detail concerning the requirements of the Website and how it should look like, the main target group has to be identified. It has to be clear which target group should be addressed and how they will use the Website. In the majority of cases the *Product Owner* knows the target group best and defines it. It also helps to define personas, one or if necessary more, who are the potential users of the product. A persona defines one particular potential user and describes this user as detailed as possible [CRC07]. This is done so that a clear picture of this fictive user can be created and, in the best case, identify oneself with this user and thus understand her/his needs.

Typical characteristics of a persona are:

- A name or an alias: This allows to talk about the user as a real person.
- An image: This can also be a drawing and is something to help with imagination.
- Demographic information: Where does the person life? How old is this person? What is the person's educational background? Family information about the user.
- Job title and functions: What are the daily tasks?

- Goals & Motivation: What could be this user's goals when she/he uses the Website? What is the user's motivation to use the Website? Define the expectations of the person as she/he uses the Website.
- Behaviour: What is the users behaviour on the Website? How does she/he interact with particular features?

Based on a good persona description typical scenarios or use cases can be derived. Having a detailed description of the user and her/his needs allows to describe typical use cases. In this case it is part of the persona description, since the goals, motivation as well as the user's behaviour is specified. These descriptions can be used to identify scenarios of use or typical use cases. It is a conventional Usability methodology, but can easily be adapted to include Accessibility. Classifying a persona with a disability could help adding a better understanding of disabled users in this early step.

In this early stage it also helps to consider some key questions targeting Accessibility. Having questions like that in mind, shows that options such as color contrast, accessible videos, keyboard usage, etc. are already considered in the use cases. Thus, they will not be missed while defining the requirements. Some examples for questions targeting Accessibility are provided here:

- How would a blind user access this information?
- Is there an option for transcriptions or captions of the video so deaf people can use it?
- What if the user does not have a mouse? Can she/he access all categories?
- Is the information architecture clear for everyone? Is the content easy to understand for everyone?

In case of uncertainty of the defined scenarios or use cases there is also the possibility for focus groups. These focus groups are used to get opinions and perceptions on the concept as early as possible. They are a group of people who are introduced to a concept, an idea, a product, etc., and during an open and interactive discussion on the product information is gathered. The groups share their thoughts, doubts, critics or endorsement towards the product. This form of qualitative research helps to get a more precise picture of the product. In addition, the focus group gives consent or disagreement towards the product depending on how they judge the product.

Utilizing these possible scenarios and use cases enables the development team to have the first requirements ready. They are still vague and not described in detail, but typical use cases are identified and requirements can be derived accordingly.

Requirement Specifications

With the help of the results from the user analysis the requirements can be specified. The requirement documentation is the basis for successive phases and thus also for the developers. Therefore it is of high importance to specify them clearly and comprehensively.

In this step Accessibility plays an important role, because the specifications for Accessibility have to be considered in this phase and are defined for the rest of the process. This is an essential part of the proposed process, to not only consider Accessibility from beginning, but also include them in the requirement specifications. If the requirements consider Accessibility from the start it should not be possible to have an implementation missing Accessibility features. Of course, this presumes that the developers implement all requirements accordingly. Having a clear and comprehensive specification enables the developers to know exactly what they are supposed to implement.

Requirement engineering itself includes, of course, more than just the user requirements. Usually the system architecture, system requirements, database models and much more are part of the requirement engineering process. Since they are not influential on Accessibility they are not described here in detail, but only mentioned for completeness. Since the mainly structural technical part of the requirement process is not relevant for the purpose of this thesis, it is excluded and left out.

As one can imagine, not all requirements will be specified right away in agile projects. There are vague requirements, which are already in the backlog, and there will be much requirements to add during the project. In a first cycle it is important to prioritize the existing requirements and describe them in more detail. This is where the Accessibility requirements have to be included. Important to note is, that the requirements do not show how to implement something, but rather specify what should be taken care of and how to handle issues. The same applies for Accessibility requirements where it is not described how something is made accessible, but rather what should be accessible in which way.

In essence, requirements indicate who is supposed to be doing what, where, why and in which way. The requirements are supported by the use case descriptions and user scenarios.

Wireframing

Wireframes or mock-ups are a great way to present a proof-of-concept and provide the *Product Manager* with an option to show the *Product Owner* how her/his ideas are visualized. Nowadays there are various tools available that allow the creation of wireframes or click-dummies, but also paper based mock-ups are a way of presenting a concept. Wireframes are essential for projects with a strong focus on Usability, but they are helpful in any case. With increasing complexity often HTML prototypes are used as a way to present a concept. Utilizing wireframes help to get a better understanding for the look and fell as well as the process flow within the software. Many tools also support the possibility for a simulation of the software.

The focus with such prototypes is on the layout and navigation. It is important to keep in mind, that mock-ups are not a one-to-one illustration of the product or how it will look like. Rather, they are only sketches to visualize the product. Mock-ups show how the layout is structured, how the navigation could look like and how the user interaction and the workflow could be realized.

Prototypes, no matter in which way they are realized, together with the requirement specifications are the basis for *Programmers*. The prototypes not only are a proof-of-concept, but also help the developer to visualize the requirements and provide them with a detailed specification of what should be implemented.

In addition to visualizing the product, prototypes can also be used for first tests. To perform such a test the prototypes, either as click dummy or papers which show the process flow, are typically presented to test-users. Ideally the test-users are “real users”, meaning they are either people who are going to use the application or they are part of the target user group. The test-users receive an introduction on the product and the purpose of it and are then testing the prototype. A common proposed test method is “Thinking Aloud” [Nie93]. Here the test-users explore the product. In case of click-dummies they can actually click and use the software, whereas with paper prototypes more interaction with the project team is necessary. During the user’s exploration it is very important that she/he articulates what she/he is thinking. The test-user should give an explanation for each step that she/he takes and verbalize her/his thoughts. Some common questions that should be answered are:

- Why is she/he doing it?
- What is she/he expecting after each step?
- In case of confusions, why is she/he confused?

These tests always include two project members and the test-user. One of the project members takes the interviewer part and the other transcribes all actions and thoughts of the user. Important is, that the interviewer is not asking the questions, but rather guides and helps the user if necessary. However, the interviewer is not answering open questions from the user. This is because the interviewer is using the questions to find out what the user would expect. So, each question that the test-user asks, provides information about the prototype. It is important that the interviewer leads the user and points her/him towards potential problems. Furthermore, the interviewer should always try to engage the test-user to articulate what she/he is doing. If the user seems unsure, it is the interviewer’s task to ask the user what she/he is thinking and what is troubling her/him.

This test method does not involve high costs and shows whether a concept is working in practice or not. However, it is also an unusual situation for the test-person, which should be kept in mind when analyzing the results.

The result of this step is a visualization of the concept including user-tests and their results. Adding these mock-ups to the requirement specifications provides *Programmers* with a solid foundation for the implementation.

4.3.3 Phase II: Design

Given the requirements and specifications of the previous phase a clear definition of what functionality the Website should offer is available. This phase builds upon that and provides a design and the look and feel for the Website. The clear focus in this phase is on how the Website should look like. The guidelines, e.g. Corporate Identity (CI) specifications or limitations considering Accessibility, are provided to the *Designer* as input and she/he has to take them into consideration. These considerations take especially color into account. For example, the ratio of the color contrast from foreground to background has to be at least 4,5:1. The Usability standards are already pre-defined via the mock-ups.

To get realistic screen designs, the *Designer* uses content which is close to the real content. This content has to be provided for the *Designer*. It is particularly important for navigation and the main sections of the Website. With this in mind, it is significant to think about the Information Architecture and how the Website is organized. *The Information Architecture Institute*[1] defines *Information Architecture* as follows:

1. The structural design of shared information environments.
2. The art and science of organizing and labeling Websites, intranets, online communities and software to support usability and findability.
3. An emerging community of practice focused on bringing principles of design and architecture to the digital landscape.

The structural design of a Website is, as shown in the definition, a very important part of the Usability considerations. Making a Website usable means, that the user finds her/his way around and clearly understands what she/he is doing. This is highly influenced by the way the Information Architecture looks like.

One useful method to get early user feedback on the Information Architecture is, for example, card-sorting. Potential users are provided with cards representing all menu items. The challenge for the test-users is to arrange the cards in a meaningful way. The results may differ from test-user to test-user since meaningful is subjective. A possible variation is to pre-define the main navigation and the cards have to be added to the

[1] `www.iainstitute.org`, accessed May 2017

navigation items accordingly. Another variation is to find adequate main topics to group the items into.

The result of this phase is a design prototype. This design prototype includes design guidelines on how the Website should look like, which colors to use and typographic specifications among others. Beside that, an additional outcome is the Information Architecture. The Information Architecture is a description of the first and second level navigation. The most important part, however, is not the description, but the naming of the single terms that are used on the Website. The navigation should look and behave the same for the entire site and features should have unvarying names.

4.3.4 Phase III: Implementation

The *Implementation* phase is a very important one as the Software is “written” here. However, in terms of Usability and Accessibility it is more a task of implementing given requirements. The specification of the requirements should clearly define what the *Programmer* needs to take into account and focus on. The result of this phase is a working implementation of all requirements ready for testing.

4.3.5 Phase IV: Testing

Testing is a very extensive and important phase within the process. Although continuous testing in each phase is employed throughout the process, it is important to have a clear test-plan before going into production. Thus, to guarantee a good level of quality, especially in terms of Accessibility, it is necessary to have a well structured testing environment and a comprehensive test-plan.

It is mandatory to test according to the pre-defined test-plan before releasing the software. In each phase functional tests, for example the developers checking the functionality of the part or module they implemented, are conducted. Unit-tests are a good way to cover the main functionality of the program. The clear focus on this thesis is on non-functional tests, respectively on Usability and Accessibility testing.

An Accessibility check should not just be performed at the end of the project. According to the expert interviews, it is common practice to do a final Accessibility test and afterwards try to integrate some of the proposed optimization to follow the law and at least have a minimum of the guidelines implemented.

Testing, implementing and correcting at the end of the project leads to frustration and to a lot of effort and extra costs. Frustration is a key phrase here because everybody is affected. The *Accessibility Tester* is frustrated because often a lot of the basic implementation is missing and the same problems have to be tested again and again. The *Programmers* are frustrated because they constantly have to restructure a lot of the code and re-implement large parts of the software. Work that seemed to be finished, suddenly returns and has to be reinvented. And of course the *Product Owner* is also frustrated if the project exceeds the time and cost estimate. This is especially true for functions that she/he cannot see or maybe does not understand. For people who have no experience with Accessibility the features that help to accomplish accessible Websites are invisible. Finally, the users who are supposed to work with the Website are frustrated since they don't have a finished product. That is what makes it so important to focus on a well-defined and structured process and keep Accessibility in focus. This phase verifies that Accessibility really was addressed comprehensively.

At the current technological state it is not possible to automate Accessibility tests completely. There are a lot of useful tools that support Accessibility testing, but there is always a manual intervention necessary. A detailed description on tools and testing is given in Section 5.

4.3.6 Phase V: Deploy & Release

To continuously test and further develop the product it is necessary to deploy the development state on a regular basis. The regularity is of high importance so that surprises after a large updated are avoided. This is known as continuous deployment/integration. Generally speaking, it is less effort to regularly check the implementation and report mistakes or even misconceptions as early as possible. Regular deployment cycles allow that requirements can be rearranged and optimized if necessary. In addition conceptual failures can be avoided and fixed at an early stage.

4.3.7 Content Guidelines

Although the *Content Guidelines* are not a particular phase in the process, they should be mentioned at this point. Before a Website can go live content has to be added. As mentioned in Section 4.2 a separate stakeholder, the *Content Manager*, is in charge of adding content. Thus, it is important to train this stakeholder accordingly. In the *Design* phase (see Section 4.3.3) the basic structure of the Website has been defined as well as the Information Architecture, but only at this phase of the project the actual content of the site is added. For static Websites the content is mostly part of the implementation itself. But if a Content Management System (CMS) is used, which is a common case, it is of high importance, that the *Content Manager* or *Content Editor* is properly instructed. The implementation of Accessibility does not end with a finished software, but also has to be targeted when entering the content. Even a perfect implementation will fail an Accessibility test because of false usage. For example, it is the *Programmer*'s or the CMS' duty to offer the possibility to add alternative text for images. However, if the *Content Editor* does not provide a good alternative text or does not enter one at all, one of the most basic Accessibility rules will fail. In this case it is not the *Programmers* fault, because the option to provide alternative text is available, but rather the fault of the *Content Editor*. Concrete examples and techniques for the *Content Manager* are shown in Section 5.

4.3.8 Launch

The *Launch* itself is actually just a milestone and not a phase. Even though the Website will be updated various times, there is only one launch date. The date is specified in the beginning and defines the deadline where the features have to be finished. As this thesis does not focus on roll-outs and time management the *Launch* is not described in detail and only mentioned here for completeness.

4.3.9 Continuous Quality Assurance & Monitoring

Throughout the process and especially in each cycle/iteration a particular level of quality has to be guaranteed. *Quality Assurance* is, of course, an essential part of the *Testing* phase, but it needs to be mentioned, that it applies to the other phases as well. Each phase should guarantee a certain level of quality, e.g. the *Programmers* using unit-tests to check the functionality of their implementation. Therefore each project member should know about the product goal and the quality level that should be achieved. This way basic testing can be executed by the responsible stakeholders in each phase.

Once the product is launched *Continuous Monitoring* is deployed as well. *Continuous Monitoring* is used to check if a Website adheres to the same Accessibility and quality standards even after, for example, bug fixes. In addition *Continuous Monitoring* helps to keep bug fixes small and that they are implemented on demand. However, even bug fixes should follow the defined process to ensure a constant level of quality. This is because even small changes to the implementation can cause negative influence on the Accessibility. For larger feature requests and follow up projects, a new process should be initiated. In this case some of the parameters and requirements of the previous projects can be reused.

In essence, *Continuous Monitoring* is employed to ensure a constant quality level even after minor changes to the implementation. In addition the Website should be checked on a regular basis for *Quality Assurance* after major changes to the Website. If updates, feature requests and their implementations are planned accordingly, the time and cost can be kept minimal.

4.3.10 Further development and enhancement

As mentioned in the previous section, further development, enhancement and new features have to be planned accordingly. To keep the user friendly quality standard of a Website, it is mandatory to plan even small enhancements. The requirements should be defined in detail and again the Accessibility should be part of the requirement description. New features should be developed accessible and the integration of the new features should not conflict with existing structure and rules. Since the same process and quality standards are used, e.g. the same kind of tests are performed, the result should be of a comparable standard. Taking this into account, the Accessibility standard could be kept at a constant high level and the rework after a feature integration is minimal.

4.3.11 Input and Output Table

Each phase has a specific result and each subsequent phase might use the results of previous phases as an input. The phases with their results and inputs are shown in Table 4.1.

Input	Phase	Output
Demand for a new Website Commitment to Accessibility	Idea, Vision, Plan	Clear picture on what is needed Basic Plan Definition of Accessibility Level
Basic Plan	Concept, Specifications & Requirements	Requirement Specifications Definition of Functionality Wireframes
Requirement Specifications Definition of Functionality Wireframes	Design	Design Prototype Design Guide Information Architecture
Requirement Specifications Design Prototype, Design Guide Wireframes Functionality Demands	Implementation	Implemented Functionalities
Implemented Functions Product Status	Deploy, Release	Testing Environment
Testing Environment Design Prototype, Design Guide Requirement Specifications Wireframes	Testing	New Tasks, Bugs Closed Tasks

Table 4.1: Input and Output of the Process Phases

4.4 Framework for Integrating Accessibility (FIA)

A reference to best practices or some kind of checklist was one of the concrete wishes the interview partners articulated. To meet this demand this thesis proposes a supporting framework for Accessibility tasks. This proposed framework targets common problems that occur in Web development projects that focus on Accessibility. It is closely related to the proposed process of Section 4.1 and is in fact a supplementary implementation of the process.

The framework is realized via a specialized, extended *Kanban Board* (see Section 2.3) and utilizes its advantages, e.g. its flexibility. *Kanban boards* can be used with various Project Management models and are typically used in agile methodologies such as Kanban or Scrum. However, they can also be applied to traditional Project Management methodologies, where the stories containing tasks are translated to different work packages. The proposed framework provides a template structure for Accessibility tasks, which can be automatically created and inserted onto a *Kanban Board*. This approach can be applied in agile methodologies as well as conventional software development methods. Since specifications are developed upfront the templates can also be generated upfront. For additional requirements that are added in later iterations of the process new tasks can be generated on-the-spot. The board is especially designed to support the proposed process.

Figure 4.2 shows how the process and the board elements are tied together.

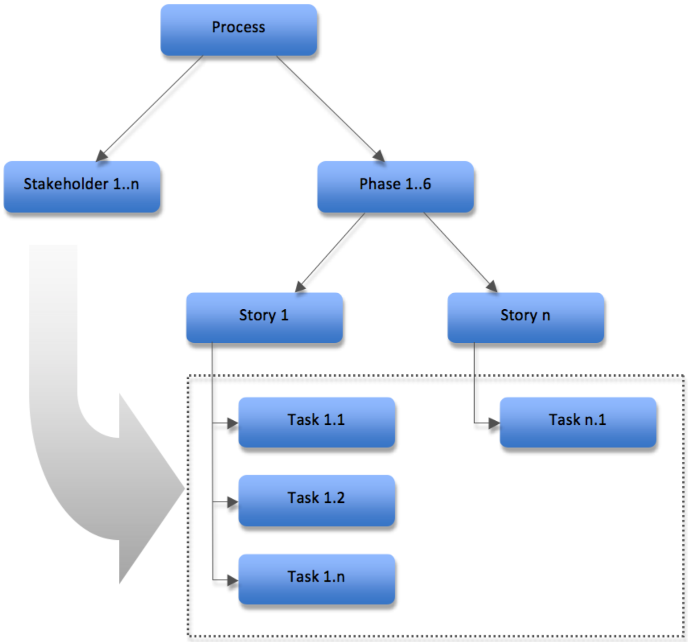


Figure 4.2: Connection of the proposed elements

The process has defined stakeholders, which comprise the project team, and different phases (see Section 4.3). The different phases have stories (or working packages) assigned to them and each story is divided into single tasks. Each of these tasks has a stakeholder attribute and is thus connected to one or more stakeholders.

Liskin et al. [LSFM14] defines user stories as an artifact illustrating a goal to be reached. The definition of Scrum by Gloger [Glo13] is in unison. These user stories are then further divided into smaller pieces - the tasks. Hiranable [Hir07] describes a similar granularity for user stories and tasks. He defines a story as a testable fragment of a feature and a task as its work unit. An example for a story is *As a blind user I want to quickly navigate through a Website.* This story leads to several tasks such as *Implement Skiplinks* or *Provide Breadcrumb Navigation.* In the proposed framework a story contains an entire feature and they are divided into tasks which are spread through various phases.

In essence, the proposed framework allows to define for a process which stakeholder is responsible for which tasks in which phase.

4.4.1 FIA Board

Based on Kanban Boards, as described in Section 2.3, a key component of the framework is the *FIA Board.* The *FIA Board* utilizes pre-defined templates for tasks related to Accessibility requirements. In addition, it introduces horizontal swim-lanes to differentiate the phases of the proposed process. A schematic illustration of the *FIA Board* is shown in Figure 4.3.

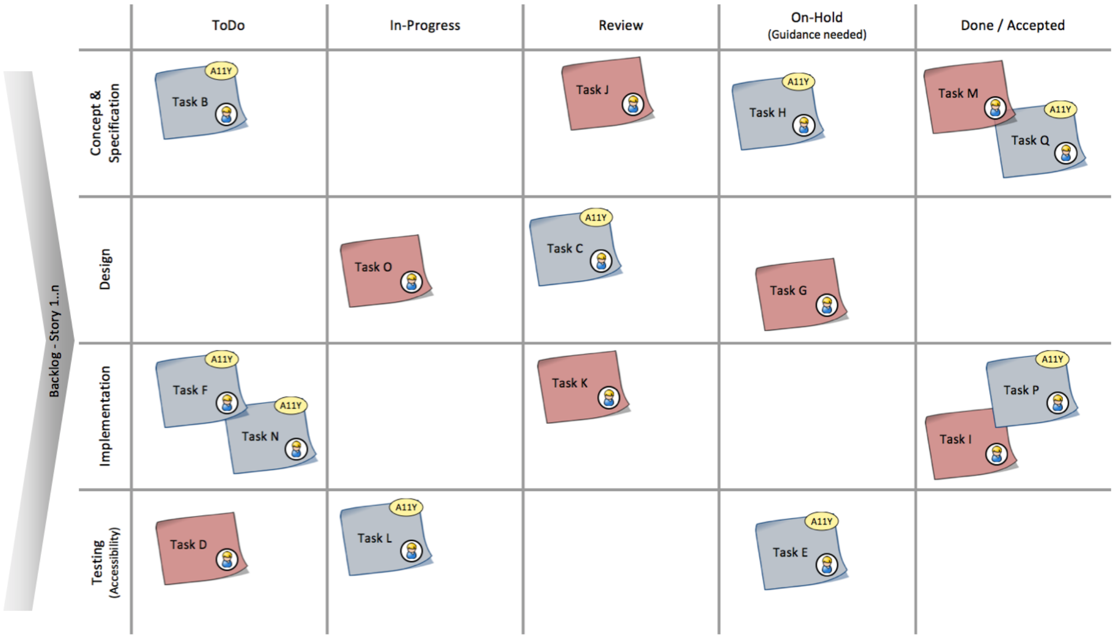


Figure 4.3: FIA Board

The *FIA Board* extends common Kanban boards by an additional defined horizontal separation. The horizontal separation is directly related to the phases of the proposed process. That is, each phase of the process is listed as a horizontal row. The vertical separation, e.g. *ToDo*, *In-Progress*. . . , is in regard of the state of a task. Hence, the board is a grid where tasks can be placed. Each area of the grid is correlated to a phase in the project as well as a status.

All stories regarding the project are kept in a backlog. These stories are not active yet and are not being worked on by any team member. If a story gets pulled, because it is time to work on the related requirements, it becomes active. From this point on the team is working on it and its contained tasks are visible on the board. A collection of all available but not yet active stories, including their assigned tasks, is listed in the backlog. However, since the framework focuses on the active tasks the backlog is not discussed here in detail.

As Figure 4.3 illustrates, the board does not only cover Accessibility tasks, but contains all defined requirements, in terms of stories and tasks, of the project. However, the framework provides pre-defined templates for Accessibility stories and tasks, and generates them automatically. Additional tasks and stories can be prepared and added as needed. To differentiate Accessibility from “normal” tasks, the cards representing these tasks are colored and tagged accordingly. The Accessibility tasks contain the tag *A11Y*. This approach helps to provide a quick overview on the current WIP of all tasks and also especially of Accessibility tasks.

In essence, the *FIA Board* relates each tagged task on the board to a phase in the project and a state that it is in.

Task Templates

A Kanban board is not complete without cards and their associated tasks. Since the proposed approach relies on pre-defined templates for tasks, it is essential to provide a detailed description of them. A schematic illustration of such a template is shown in Figure 4.4.

As Figure 4.4 illustrates, not all attributes are visible in each view. There are basically two views: the board view and the task detail view. The board view is a view of the overall board with all tasks. In this case only limited information about the tasks are displayed. The detail view provides all the details of the task in a single view.

The template of a task consists of various information such as *Title, Description, Type, Member, Comment, Attachment.* In addition, there are designated attributes relating the task to the *WCAG* standard and the *Accessibility Toolbox* of Chapter 5. Not all of these attributes are necessarily visible for all project members. It is conceivable that based on permissions some of them are hidden and only the relevant ones are displayed. Some of the values of the attributes are pre-defined in the template, while others have to be filled out manually. An example for a manual fields is *Comment* as the values here will differ from project to project, or even more precisely, from task to task.

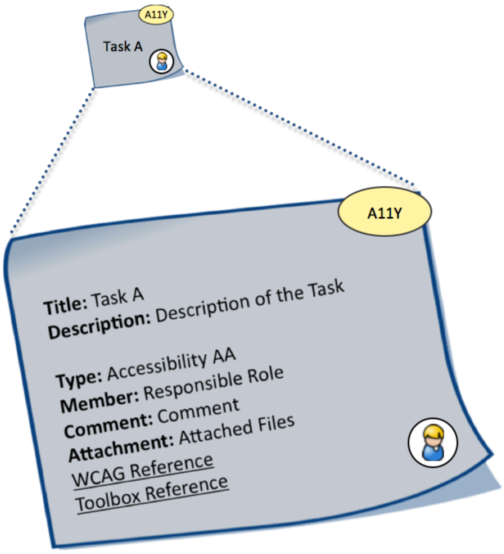


Figure 4.4: Schematic illustration of a task template definition

Some attributes of the templates are reserved to connect it to the *Task Generation Workflow*. Since tasks are automatically generated the templates need to contain information about predecessor and successor tasks. More information about the generation approach is given in Section 4.4.1.

A detailed description of the attributes and meta-data defined in the templates is provides here:

Internal ID Internal (hidden) field for task identification.

Accessibility Level Based on the WCAG levels A, AA, AAA, this attribute is visible in the details of a task.

Title Short, but descriptive, title of the task. This attribute is visible in the task detail view as well as the board view.

Description Clear description of the task. Only visible in the task detail view.

Stakeholder If applicable to a task, a stakeholder can be pre-defined. Otherwise the stakeholder has to be added manually. This attribute is visible in the task detail view and the board view.

Attachment Optional files can be attached to the tasks - within the process. In addition, attachments can also be pre-defined in the template. This attribute is visible in the task detail view.

Phase The phase of the process that the task should be automatically assigned to. This field is not visible since it is only used to classify the task.

Associated Story Superior story provided as a link. Visible in the task detail view.

Successor Task List of task IDs that follow afterwards. This is used for automated generation of follow-up tasks. Internal field that is not visible.

Dependencies Tasks can be dependent on other tasks. In this case they are not generated right away but rather when the dependent tasks are finished. Independent tasks can be generated right away. Internal field that is not visible.

WCAG Reference Link to the corresponding WCAG guideline. This attribute is visible in the task detail view.

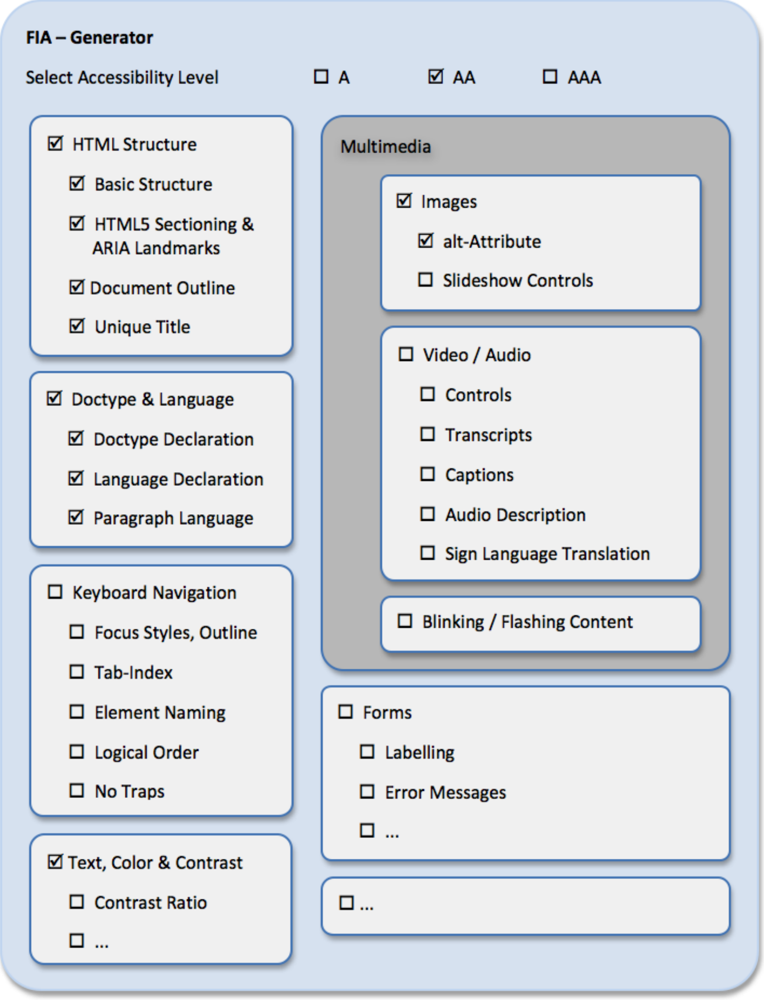
Toolbox Reference Detailed description and tips to implement the task. May provide a reference to an external tool or resource. This attribute is visible in the task detail view.

Defining templates for user stories works similar but uses less attributes. A user story only needs data like *ID, Title, Tasks* and a flag *Active*. The *Tasks* attribute is used to identify the associated tasks. This is necessary so that all relevant tasks can automatically be generated once a user story becomes active. The *Active* attribute is used to distinguish between active, stories that are already in development, and inactive stories. The activation of a story is triggered by the team by moving the story-card from the backlog.

FIA Board Management

Generating the Accessibility tasks on the board can be managed via the *FIA Board - Generator*. The generator can be configured as shown in Figure 4.5.

The *FIA Board - Generator* enables the project team, among other options, to chose the Accessibility level. The Accessibility level is essential so that the basic tasks for Accessibility requirements, which are generally needed on every Website, can be generated automatically. Individual options, which are known as being not necessary in the project, can be deselected manually. Beside the basic requirements also special requirements are displayed. These requirements represent commonly used elements on a Website like, for example, *Multimedia*.



FIA – Generator

Select Accessibility Level □ A ☑ AA □ AAA

☑ HTML Structure
- ☑ Basic Structure
- ☑ HTML5 Sectioning & ARIA Landmarks
- ☑ Document Outline
- ☑ Unique Title

☑ Doctype & Language
- ☑ Doctype Declaration
- ☑ Language Declaration
- ☑ Paragraph Language

□ Keyboard Navigation
- □ Focus Styles, Outline
- □ Tab-Index
- □ Element Naming
- □ Logical Order
- □ No Traps

☑ Text, Color & Contrast
- □ Contrast Ratio
- □ ...

Multimedia

☑ Images
- ☑ alt-Attribute
- □ Slideshow Controls

□ Video / Audio
- □ Controls
- □ Transcripts
- □ Captions
- □ Audio Description
- □ Sign Language Translation

□ Blinking / Flashing Content

□ Forms
- □ Labelling
- □ Error Messages
- □ ...

□ ...

Figure 4.5: *FIA Board - Generator* for template-tasks

The relevant stories and tasks are automatically generated based on this configuration. Figure 4.6 shows the requirements tree containing proposed requirements for the predefined stories and tasks. It has to be mentioned that the main focal point of this thesis is not to provide a complete list of requirements, but rather provide a concept on how these requirements can be handled. Generating a complete list of requirements is beyond the scope of this thesis. An expansion of the template definitions will be dependent on the project type and might warrant further exploration by the project team. The tree visualization as well as the template definition can be expanded to fulfill further requirements.

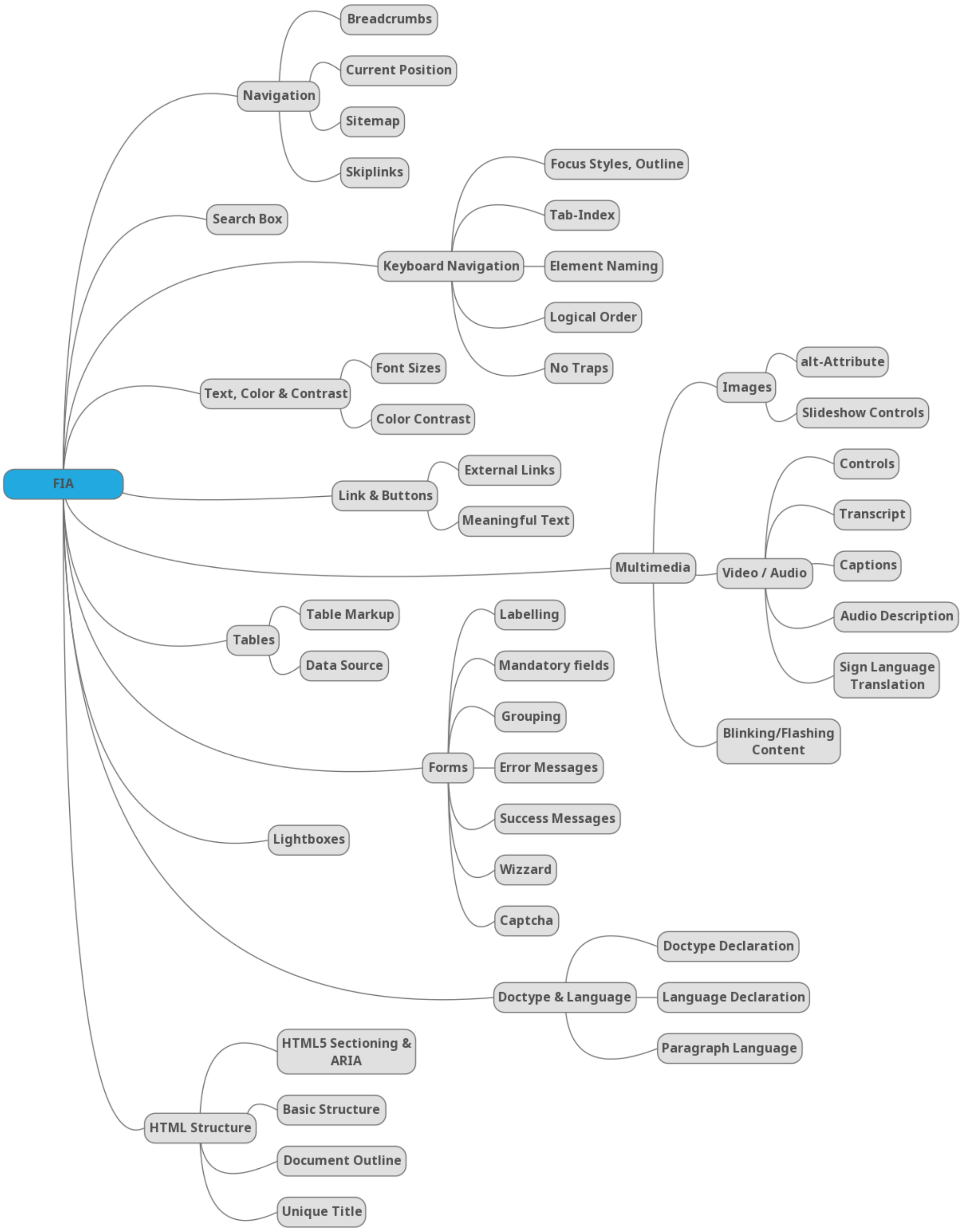


Figure 4.6: Standard Accessibility requirements of FIA Templates

Task Generation Workflow

One of the advantages that the *FIA Board* features, is automatically generating Accessibility tasks via pre-defined templates. As mentioned in Section 4.4.1, task-templates provide attributes for *Dependencies* and *Successors.* Tasks are created automatically based on these attributes.

As soon as a story gets activated all related tasks are analyzed. The ones without a predecessor attribute or the ones that are independent, e.g. empty *Dependencies* attribute, are automatically created and assigned to the *ToDo* column of the corresponding phase-lane. The task-card will be moved according its state.

If a card attains the *Done* status on the *FIA Board* all tasks defined in the *Successors* attribute are created automatically. Hence, moving a card to the *Done* state triggers the generation of the defined successor-tasks. To accomplish this the system checks whether the task has defined follow-up tasks and creates them accordingly. The creation is based on the *Phase* and *Stakeholder* attribute. All newly created tasks are automatically put into the *ToDo* state.

4.4.2 Showcase

A showcase of FIA is given here to illustrate how the proposed framework behaves during development. The story chosen for this showcase is *As a customer I would like to play videos on my Website.* The required Accessibility level is set to AA.

User Story

ID	S001
Title	As customer I would like to play videos on my Website.
Description	Videos need to be embedded in the content. An Accessibility Level of AA should be granted. The videos need to have a heading and an introduction text...

Automatically Created Tasks by FIA

Table 4.2 shows how the first automatically created tasks for the user story *As a customer I would like to play videos on my Website* look like. It starts with the specification of the video player and what elements are needed. This task is generated for the phase *Concept, Specifications & Requirements* and assigned to the *Requirement Engineer*. Result of the task should be a requirement description including optional wireframes to visualize the output. Once this task is finished, i.e., moved to the state *Done / Accepted*, three new follow-up tasks are generated *T002, T003, T004*.

Task 1	**Video Player Concept - Wireframe**
ID	T001
Accessibility Level	A
Title	Video Player Concept
Description	Provide specifications and wireframes as basis for the video player (incl. controls, heading, pre-video-text, captions)
Stakeholder	Requirement Engineer
Attachment	–
Phase	Concept, Specifications & Requirements
Associated Story	S001
Successor	T002, T003, T004
Dependencies	no
WCAG Reference	Links to 1.1, 1.2, 1.3, 2.2
Toolbox Reference	5.1.1

Table 4.2: Template Task1: Video Wireframing

Table 4.3 shows how the task for the *Design* is evolved from the previous task. Based on the results from the previous task *T001*, this task's goal is to create a design for the video player, which includes the necessary control buttons. This task's goal is to make sure that the design of the video player includes the buttons in an accessible way in terms of the design, including according contrasts and focus styles, for example.

Task 2	**Design Video Player**
ID	T002
Accessibility Level	AA
Title	Design Video Player
Description	Design the video player element including the necessary control buttons (stop, play/pause, mute/unmute, volume control, hide/show CC)
Stakeholder	Designer
Attachment	T001 Specification and/or Wireframe
Phase	Design
Associated Story	S001
Successor	T005
Dependencies	yes
WCAG Reference	Links to 1.1, 1.2, 1.3, 2.2
Toolbox Reference	4.3.3

Table 4.3: Template Task2: Design Video Player

Table 4.4 shows the additional task that is automatically generated during phase *Design*. The specified heading and summary from task *T001* get designed within this task.

Task 3	**Heading and Summary**
ID	T003
Accessibility Level	AA
Title	Design Consideration Heading and Summary
Description	The design needs to take providing a heading and a summary to the video into consideration. Each video needs a related heading and summary in context of the video player
Stakeholder	Designer
Attachment	–
Phase	Design
Associated Story	S001
Successor	T006
Dependencies	yes
WCAG Reference	Links to 1.1, 1.2, 1.3
Toolbox Reference	4.3.3

Table 4.4: Template Task3: Video Heading and Summary

Table 4.5 shows another additional task that is automatically generated during phase *Design*. Another task generated from task *T001*, which takes care of a proper design for closed captions, to guarantee their readability.

Task 4	**Design Closed Caption**
ID	T004
Accessibility Level	AA
Title	Design Specifications for Closed Captions
Description	Provide specifications on how the closed captions look like.
Stakeholder	Designer
Attachment	–
Phase	Design
Associated Story	S001
Successor	T005
Dependencies	yes
WCAG Reference	Links to 1.1, 1.2, 1.3
Toolbox Reference	4.3.3

Table 4.5: Template Task4: Design Specifications for Closed Captions

Table 4.6 shows how the information for the *Implementation* phase task template is created for the user story *As a customer I would like to play videos on my Website.* This task is generated after the design is set, meaning the cards for tasks *T002, T004* are moved to state *Done / Accepted.* Output of the task is the implemented video player.

Task 5	**Implement Video Player**
ID	T005
Accessibility Level	AA
Title	Implement Video Player
Description	Implement the video player element including the necessary control buttons (stop, play/pause, mute/unmute, volume control, hide/show CC). Make sure the controls are keyboard accessible and have appropriate accessible naming.
Stakeholder	Developer
Attachment	T001 Wireframe, T002 Design Guide Video Player
Phase	Implementation
Associated Story	S001
Successor	T006, T007
Dependencies	yes
WCAG Reference	Links to 1.1, 1.2, 1.3, 2.1, 2.2
Toolbox Reference	5.1.3

Table 4.6: Template Task5: Implementation Video Player

Table 4.7 shows how the information for the *Implementation* phase task template is accumulated. This task takes the implemented video player from task *T005*, as well as the design specifications from task *T003* to further develop the video player and provide the surroundings of the player, like heading and summary.

Task 6	**Implement Heading, Summary and Transcript**
ID	T006
Accessibility Level	AA
Title	Implement Heading, Summary and Transcript
Description	Provide markup for heading and summary for the video. Furthermore add a possibility to access the video transcript.
Stakeholder	Developer
Attachment	T001 Wireframe, T002 Design Guide Video Player
Phase	Implementation
Associated Story	S001
Successor	T007
Dependencies	yes
WCAG Reference	Links to 1.1, 1.2, 1.3, 2.1
Toolbox Reference	5.1.3

Table 4.7: Template Task6: Implementation Video Player

Table 4.8 shows how the information for the *Testing* phase task template is accumulated. After finishing the implementation tasks *T005, T006* the testing task *T007* is generated. The output of the precious tasks builds the input for this task, including all necessary information, like design specifications alongside the implemented player itself. With the completion of this task the story is finished, and the video player is implemented including all Accessibility relevant aspects.

Task 7	**Video Player Testing**
ID	T007
Accessibility Level	AA
Title	Testing the Video Player
Description	Test the implemented video player according to the accessibility demands (keyboard usage, screen reader software)
Stakeholder	Test Engineer, Accessibility Expert
Attachment	T001 Wireframe, T002 Design Guide Video Player, T005/T006 Implementation
Phase	Testing
Associated Story	S001
Successor	–
Dependencies	yes
WCAG Reference	Links to 1.1, 1.2, 1.3, 2.1
Toolbox Reference	5.1.4

Table 4.8: Template Task7: Testing the Video Player

4.5 Limitations

The process alongside the supporting framework is a blueprint for developing accessible Websites. However, there are some limitations that apply and have to be considered.

First, as mentioned in previous sections, all involved parties, meaning the whole project team, should be incorporated and integrated in the project life-cycle. Furthermore, each member should have a fundamental knowledge about Accessibility basics. This is especially true for the stakeholder specifying the requirements (*Project Manager* and *Requirements Engineer*) and the stakeholder in charge with evaluating and testing the Website (the *Test Engineer*). These stakeholders should have a profound knowledge about Accessibility. Especially because one of the key problems of missing Accessibility is the lack of knowledge [AB16] in project teams, however this is not targeted in the thesis as the proposed process depends on knowledgeable members, or rather supports the team members on Accessibility matters.

Besides the fundamental knowledge about Accessibility, following the process also needs a commitment in real life's work. All stakeholders, including the *Programmers*, need to consider Accessibility as a matter of high importance. But most importantly the *Product Owner* must be willing to commit to the process and its structure. This means, that she/he needs to be aware of issues related to Accessibility and disabled users. Otherwise achieving accessible Websites is problematic. Abuaddous et al. showed in their work, that one of the main issues with Accessibility is the missing awareness throughout the stakeholders [AB16]. Without the *Product Owner's* compliance, or even better her/his demand for an accessible Website, it is not easy to persuade her/him to follow the process. This is, because the proposed process might seem to restrictive and with a lot of overhead. A *Product Owner* might not want to invest extra time and money into a feature she/he deems unnecessary. However, as described in previous sections, if a team is trained and familiar with the process, then the process will not require a lot of overhead and extra effort. The interviews made clear, that the commitment of the *Product Owner* is a key requirement for a successful project. The following quote from Interviewee A highlights this issue.

> *[...] An explicit commitment from the customer is needed from the beginning and should not be questioned. Because, if the customer questions Accessibility or continuously postpones the demands, it gets unequally more expensive.*[2] (Interviewee A, p. 161)

A further evident outcome of the interviews was that, in general, developing with a Usability and Accessibility focus in mind is more work. It also demands more restrictions and grants less flexibility in reference to various parts of software, e.g. design. By having a commitment and a trained team the downsides can be limited to a minimum.

[2] Translated from German by the author

Especially as the team gets more and more acquainted with the process and the framework, these limitations can be overlooked. The interviewees, especially the developers, claim specifications upfront would reduce their effort by a large factor.

A limitation of the thesis itself is, that it only considered the Austrian market. Though the Internet easily connects the world, the regulations and examples of this thesis are focused on Austrian law and domestic Websites. However, the literature research does not focus solely on Austria.

The basis for the thesis is the WCAG standard and the tasks proposed in the *FIA Template* (see Figure 4.6) as well as the implementation recommendations from Section 5 are based on this standard. This means that known limitations of the WCAG also apply to this thesis [PFPS12]. Still, the WCAG is an officially acknowledged standard, which is why the focus of the thesis was based on it nevertheless. This being said, it is clear that the thesis cannot guarantee a fully accessible Website. The process proposes a workflow, but granting full compliance without any barriers is beyond the focus of thesis.

Not only does the thesis focus on the Austrian market, but also the selected interview partners is one of the restraints of the thesis. The Accessibility scene in Vienna is rather connected, which could lead to an unidirectional view. This is not only because of the local closeness of the interview partners, but also because they have similar clients. The interviewee's business market is based in Austria, which again leads to similarities. A further consideration of experts and the situation in other countries would be of great interest. An especially intriguing investigation would be to compare experts from different countries.

The thesis also does not provide an implementation of the framework. However, the aim was to only propose and not implement a framework supporting the project manager as well as the team. This proposed framework is introduced in a theoretical manner. Providing an implementation along with an evaluation of the process as well as the framework in real life projects is desirable and a task for future work.

Another limitation of the proposed approach in this thesis is missing considerations for mobile apps. The thesis focuses on Web development and here specifically on desktop computing. Of course, with responsive and adaptive Websites, mobile devices are included somehow, but the specialties for mobile devices and testing have not been observed in the thesis. Anyhow, detailed research on mobile Accessibility behaviour and evaluations are left open for future work.

CHAPTER 5

The Accessibility Toolbox - Methods to Support Developers

This section describes methodologies and tools that are commonly used in the Accessibility field. They are also intended to support a project team when working with the proposed process of this thesis. In essence, this section identifies the main stumbling blocks when working on Accessibility related projects and how these stumbling blocks can be prevented.

The expert interviews showed that the WCAG descriptions are not easy to understand. This is especially true for someone who is not yet well-educated in the Accessibility field. It takes a lot of effort and time to familiarize oneself with these guidelines and to understand them in detail. Being able to utilize them in practice takes even more effort. Sadly, for a lot of the developers spending so much extra time is not viable. The proposed process together with this *Accessibility Toolbox* should make it easier for developers to apply the WCAG rules and guidelines in practice. Furthermore, the *Accessibility Toolbox* should support practitioners during the development process. The idea of this *Accessibility Toolbox* is, to provide an outline on how to integrate Accessibility, what kind of tools are available and how WCAG can be addressed the best.

The main focal points include:

- Understanding how to implement the proposed process.
- What is necessary for a proper Accessibility integration?
- How can Accessibility be tested? How can a particular level of Accessibility be guaranteed?

5.1 Accessibility Methods for each Process Phase

The *Accessibility Toolbox* shows which rules or guidelines can be applied in which phase of the proposed process cycle. Each phase of the process is addressed separately and methods supporting the stakeholders are introduced. However, the goal of the *Accessibility Toolbox* is not to fully comply with the WCAG guidelines, but to help to maximize the Accessibility of a Website. Developing a perfectly compliant Website could still lead to Usability and Accessibility problems. None of the standards cover every issue that a disabled person might face. Furthermore, not all needs of disabled persons can be addressed systematically at the same time. Thus, a standard and this *Accessibility Toolbox* can only assist to balance the various needs that different disabilities demand[PFPS12]. Moreover, with the rapid evolution of technology the requirements change and the need for updates of outdated standards and guidelines will arise.

5.1.1 Concept, Specifications & Requirements

The *Concept, Specifications & Requirements* where a requirements analysis is performed, is one of the key phases of a project. A proper definition on what should be achieved, how the user stories look like and how everything should be integrated, is the foundation of every project. Missing requirements and specifications lead to misunderstandings and may result in chaos during development and frustration in the project team. This will most likely cause time and budget issues. Thus, it is especially important to define the Accessibility demands already upfront. The basic requirements have prepared template tasks defined in the FIA (see Section 4.4) and thus can be chosen accordingly. Identifying the requirements is a manual task and FIA can only provide suggestions. If templates are missing they can be added or existing templates can be expanded.

5.1.2 Design

Designing and prototyping UIs based on requirements takes various options into consideration. A clean and comprehensive prototype without ambiguities upfront minimizes the effort and confusion in subsequent phases. One of the most important decisions is the general structure of the Website.

The Accessibility requirements are grouped into different subjects that the *Design* phase affects and recommendations are given in the following sections.

Navigation Design

The main structural element of a Website is the navigation. In the *Design* phase the look and feel of the navigation is set. Thus, some considerations are already necessary in this early phase. A maybe obvious, but still very important, point is, that the navigation has to be consistent. The look, placement, content and the functionality of the main navigation should be consistent within the entire Website.

Ideally the design also considers *breadcrumbs*, since this is one of the check-marks in WCAG 2.0. *Breadcrumbs* help the user to better understand the structure of the Website. They support the orientation on a Website and are also a simple way to navigate back and forth.

Another design consideration is, that the current position on the Website has to be visible. That is, the selected menu entry should be visually different from the rest and visualize on what page a user is.

An important widget, that increases the Usability of a Website, is a search box. Each large and content-rich Websites should have one integrated. Some "search-dominant" users will always look for a search box and most likely use the search to quickly navigate through a Website [PWNG97, Kru14]. Also, for blind users, the search is a good way to navigate. If the user has a purpose why she/he is browsing the Website, most of the times users are looking for something specific, then a search box is essential. For Websites featuring news or other chronological ordered content, search is indispensable anyway. The design should consider a search box and, due to Usability purposes, it should be placed at least above the fold or rather in context of the navigation since this is where users most likely will look for it.

Text, Colors & Contrast

Good readability is essential on Websites and thus the typography has to be chosen wisely. There is a rather large difference in reading on screen versus reading on paper and the designer should distinguish both. A general rule is to not skim on white space. This not only helps with orientation on a site, but also makes scanning pages easier and supports better readability. In addition the font should be chosen accordingly. *Sans Serif* fonts should be picked carefully because of the easy confusion between letters, for example "l" and "I". A study of fonts for dyslexic people showed, that fonts like *Helvetica*, *Courier*, *Arial* and *Verdana* are producing good results. In contrast, italic fonts lead to decreased reading performance as Rello and Baeza-Yates showed in their work. In future work they want to conduct the same study on font effects for people without dyslexia [RBY13]. It often helps to think of extremes first and by targeting them the overall result is increased as well.

During the *Design* phase not only fonts are important, but also their sizes. A standard and commonly used size is 16 px for content and respectively 36 px for main headings, which translates to 1 em and 5 em in common browsers. Paragraphs should be easily distinguishable by adding extra space in between them and the line height between the content text should be set to about 1.5 times the font size. The line length has to be be taken into consideration as well and should be about 60 to 80 characters long on desktop sites and for mobile devices a common rule is 35 to 40 characters length [Pam14]. Although these are general Web design rules they are necessary to increase the Usability and Accessibility.

Since the common reading style is from left to right, it is an advantage to align text from left to right. Text in justification works well for newspapers, but is not a recommended alignment for Web content since it is much harder to read. Big gaps between words do interfere with the reading flow as illustrated in Section 5.2.2.

Real typefaces are always to be preferred instead of text-images. This is, because assistive technologies do not correctly interpret text in images, and thus this text is hidden from users using screen reader software.

In recent times a common design choice is images with text on them. This is always a difficult design decision, because it is hard to make it readable for everyone. The image has to be picked carefully to make it readable at all, even for normal sighted people. If a Website is dynamic and editorial, with a lot of changes, the image choice takes extra effort. A compromise by adding semi-transparent or solid boxes underneath the text to increase the contrast ratio guarantees proper readability. Figure 5.1 shows that, with little impact on the image itself, the difference in readability is severe. Still, this added text has to be made visible for screen reader technology, either by providing text in the `alt` - attribute of the image or as a link target if the image is linked.

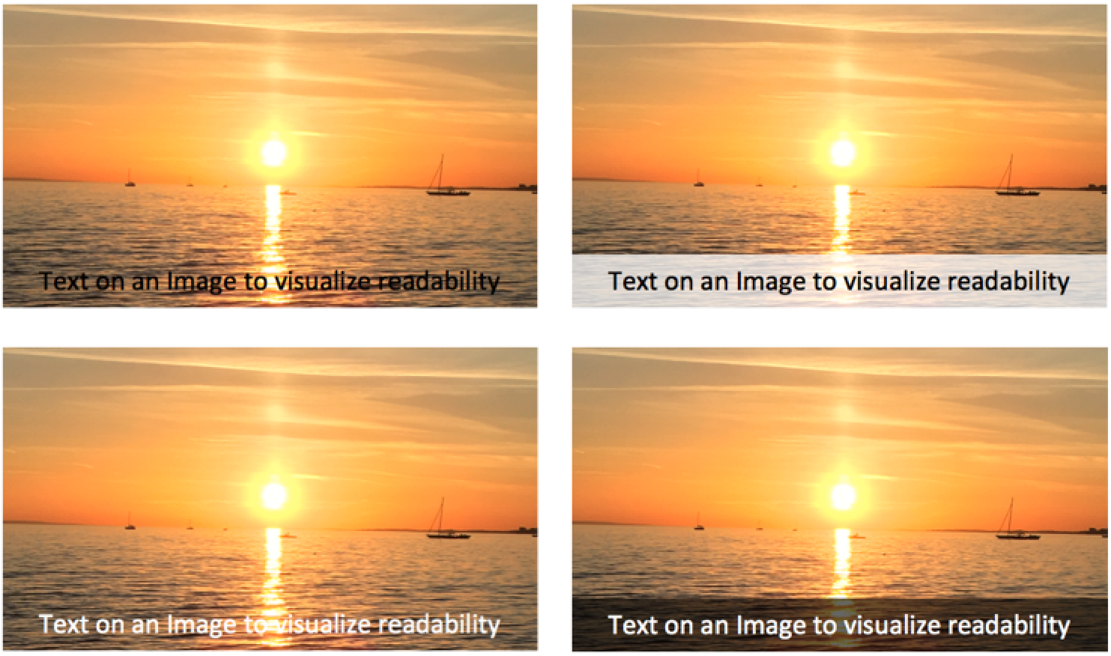
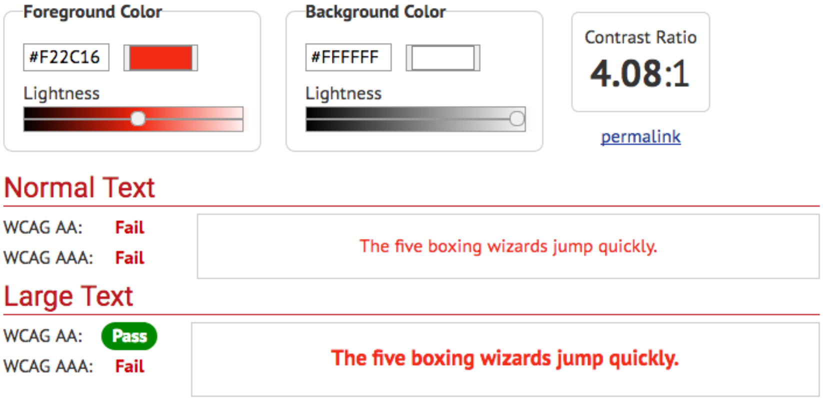

Text on an Image to visualize readability
Text on an Image to visualize readability
Text on an Image to visualize readability
Text on an Image to visualize readability

Figure 5.1: Effect on the readability on images with text

A design consideration with high Accessibility relevance is the contrast between background and foreground. For normal sized text the WCAG level AA requires a contrast ratio of 4.5:1 and 3:1 for large text. For level AAA a ratio of 7:1 for normal text is necessary. The definition, according to WCAG, for large text is from 14 point (about 18 px) upwards for bold text and 18 point (24 px) and larger for none bold text. There are various tools available that allow checking the contrast and guaranteeing a valid ratio. The online contrast checker[1] of the *WebAIM* can be used to analyze the contrast of colors [fPwD17c]. The online tool presents the results after entering the hexadecimal codes of the foreground and background colors. Figure 5.2 shows the *WebAIM* color contrast checker.



Foreground Color: #F22C16

Background Color: #FFFFFF

Contrast Ratio: 4.08:1

Normal Text

WCAG AA: Fail

WCAG AAA: Fail

The five boxing wizards jump quickly.

Large Text

WCAG AA: Pass

WCAG AAA: Fail

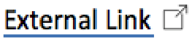
The five boxing wizards jump quickly.

Figure 5.2: WebAIM Color Contrast Checker

Designing Links & Multimedia

Although links are common on Websites their design still needs to be considered, because color should not be the only indication for a link. This is especially true for links in the content, which are hard to distinguish from the normal text. A best practise approach is to additionally underline the links contained in content.

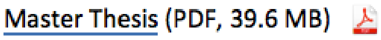
A differentiation between external and internal links should be clearly visible. External links or links opening a new window should be identified as such by, for example, including an icon indicating a different (external) link-type. Figure 5.3 illustrates an external link with an icon as an indicator.



External Link

Figure 5.3: Indicator for external links

If documents or other content is made available for download, the link text should describe the download properly. Not only should the link contain the name of the download, but the size and format should be stated as well. In case the content of the download is in a different language it should also be indicated. Figure 5.4 shows a download link of a file with the proper visual and text indication.



Master Thesis (PDF, 39.6 MB)

Figure 5.4: Download link with proper text indication

[1] http://webaim.org/resources/contrastchecker/

Icons can be an eye-catcher and their use definitely has some UX advantages. However, they can also be misunderstood quite easily. Explanations or navigation items only presented by icons are sometimes hard to interpret and introduce confusion. Only common universal icons, like for example the hamburger-icon (presented by three horizontal lines) used for mobile menus, can be used without a text alternative. The majority of icons need an additional text label though. A good example is the Apple Iphone, which still uses labels for, in the meantime already well-known, icons in the *phone*-App. Examples of these icons are shown in Figure 5.5. This is more of a Usability issue, than it concerns Accessibility, because in terms of Accessibility a text alternative in form of the `alt` element is needed anyway.

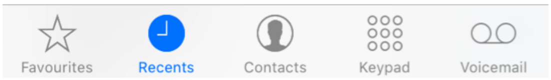


Figure 5.5: Icons with labels on an Apple Iphone

For videos it is good practice to provide a heading and/or a summary. There are several options to consider depending on the level of Accessibility. A video player needs to provide the means to control options of the video. This means, that the video player should grant a way to play, pause and stop the video and further offer options to control the volume, including mute and unmute functionality. It is important that these control elements are not only provided by design, but they should also be accessible by other means. Fully accessible videos also need Closed Captions (CC) and thus a fully accessible video player needs the functionality to display these as well. For the CC the design needs to ensure, that they are readable in terms of contrast between the text and the video. A format where the text is provided on an underlying box, similar to the previously proposed approach for text on images in Figure 5.1, is recommended. In addition, if a textual transcript is provided, than there needs to be an easy way of accessing it as well.

An integrated audio player has similar design needs. Heading and summery for the content would be preferable and, much more important, audio controls, which are the same as for video players, should be available.

Recently animations and auto played sounds are commonly used on Websites and thus it has to be considered in the *Design* phase. Big sliding banners, where the header image switches automatically, are very popular at present, but often lack support for Accessibility. Thus, the designer needs to include, for each slider, automatic scrolling or moving content, a way to stop the animation. A simple example would be the inclusion of a pause-button. Animated content which moves quickly and is not stoppable, is, not only for people with disabilities, a barrier as many users are annoyed by them. Another reason is that if the moving content provides information of interest, the user has to wait for the content to reappear. Hence, the implementation of moving content should either be stoppable or not automatic at all. Using arrows or other button-styles to slide manually through the content is a much better choice . Furthermore, any quick blinking or flashing content should be avoided, as people could react badly on it. Epileptic people for example are highly affected by such content.

Design of Forms

Forms have always been a noteworthy part in Accessibility as they are very easy in theory, but still have a lot of pitfalls to respect when taking screen reader software and keyboard users into consideration. Though, enhancements related to Accessibility also help all users. For forms the major attention takes the implementation, but the design can also grasp some of the common problems at an early stage.

First of all, consistency is a key part in making forms accessible. All form fields within a form and ideally all the forms on the entire Website should look alike. Consistency is a common design consideration and often the key to an easy usable and accessible Website. Making things with similar functionality look the same facilitates all users. Therefore, it is important that all input fields look the same, all labels are similar as well as consistent to use. This should be realized among different types of form fields such as date pickers or text fields.

To help users understand forms, each form field should have an explicit label and fields with the same purpose can have identical descriptive labels. The label should be placed in context of the corresponding field. Long forms should be grouped by similarity and including a heading for each group.

At the end of each form a submit button is necessary - it should be at the end (bottom) of the form and not somewhere in between. If the submit button is in the middle of the form, then most users will probably skip the remaining fields. So, either the user misses something due to the fact that she/he overlooked the fields, or the owner of the Website misses out on useful data. In some cases a validation step is performed and the user might receive an error message because some of the fields have not been filled out. The submit button should be clearly recognizable and its purpose should be identifiable. Hence, the text of the button should be chosen carefully. A generic button text like *ok, confirm, validate, ...* should be avoided and the purpose of the button, for example *confirm payment, register, add to list, ...*, should be highlighted.

When designing a form in multiple steps - a so called wizard - then the number of total sequences as well as the current step a user is on should be indicated. A possibility to switch between steps and returning to the previous one should be provided. Before submitting a form it is good practice to include a summary page presenting the entered data. An order process in Web-shops is an example for an implementation according to this common practise. It provides the user with an overview on what she/he just entered and enables the user to correct or change her/his data.

If the user is expected to enter data in a form then the design should also take validity and errors in consideration. Mandatory fields have to be marked accordingly and, for complex forms or form fields, a way of guidance for the user should be included. For example an indication for the measurement of a field (per piece, per kg, specific format for uploads,...) or small info-icons should be provided. If errors occur they have to be caught and explained to the user by depicting explicit error messages. The error message needs to be in context of the form field, where the error occurred, or at least has to be connected to it. It is not necessary to highlight the faulty input boxes, but a clear error message has to be displayed. It is especially important for visually impaired users that the context of a form field including a descriptive text is provided.

Last but not least, forms are a weak spot on a Website opening doors for hackers and bots misusing it. A low level security and anti-spam system preventing such attacks should be put in place. One option for such a system is CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart). The goal of CAPTCHAs is to differentiate between a human user and a computer. Usually CAPTCHAs implement a dialog with the user, in which the user is tasked with typing in a text displayed in an obscured image. These images can typically not be read by a computer, but are also in most cases not accessible at all.

Not so long ago, it was sufficient to implement a so called honeypot, which was a via CSS hidden input field. A real user could not see the field and thus would not enter data in it. A machine on the other hand, ignores the CSS and always fills in all the form fields found in the source code. Hence, adding this hidden field with a custom label, for example *Spamprotection! Do not enter anything*, which tells the screen reader user what to do, would do the trick protecting against bots. A bot would be the only one entering something in such a field since it could not understand the semantic of the associated label. This field was checked to be empty and thus provided protection against bots. In the meantime bots are trained to recognize this method and thus there is no real accessible alternative to CAPTCHAs available. Although the technology and the CAPTCHA implementation are getting better the perfect solution is yet to be found.[2] Again, this is not only a design issue, but has to be addressed in the *Implementation* phase as well. The design phase only makes the decision which CAPTCHA can or should be used.

Table - Design

Tables are taking a minor role in the *Design* phase. The most important consideration when it comes to tables is, that they should not be used for layouting purposes. This used to be common, because of the ease of usage and the fixed layout, but nowadays, with different output screen sizes and responsiveness, it is very bad practise to misuse tables for layouts. Although a table can be implemented correctly accessibility-wise, it is still not recommended to use tables for layouts. Tables should be reserved for data only.

Additional considerations concerning tables are more implementation issues rather than they are a design choices.

[2] `https://www.w3.org/WAI/GL/wiki/Captcha_Alternatives_and_thoughts`, accessed Feb 2017

5.1.3 Implementation

The *Implementation* phase uses the inputs of the *Concept, Specifications & Requirements* and *Design* phase. The following sections show common pitfalls and discuss possible solutions.

Doctype, Markup & Metadata

Conformance standards allow different browsers to render a HTML document correctly and consistently. The header of a document, as shown in Listening 5.1, encodes the information for the browsers. Each HTML document has to start with the `<!doctype>` declaration, to tell the browser the version of the HTML of the page.

```
<!doctype html>
<html lang="en">
  <!-- Only head and body elements as direct descendants of the html
      element. The hmtl element contains the language declaration for
      the page. -->
  <head>
    <!-- The head element must be a direct descendant of the html
       element -->
    <!-- character encoding -->
    <meta charset="utf-8" />
    <!-- additional meta information -->
    <title>Pagetitle</title>
    <!-- Every page within a Website must have a distinct page title.
       -->
  </head>

  <body>
    <!-- main content -->
  </body>
</html>
```

Listing 5.1: Doctype and Metadata of a HTML document

The markup of the HTML source code can be checked with the *W3C Validator*[3] and validated against common HTML standards. The problems or errors that are found, could lead to the identification of lacking Accessibility implementations. Thus, taking care of the identified problems guarantees a better quality of the code.

Language

The main language of the Website is specified in the `lang` attribute of the `<html>` tag. The language definition of a Website instructs screen reader software to use a specific language. Setting the screen reader to a wrong language makes it not only obnoxious to listen to, but also hard, if not even impossible, to understand.

[3] `http://validator.w3.org/`, accessed May 2017

```
<html lang="en">
```

The same applies if the language is changed within the content. If a paragraph is only available in a specific language, this has to be indicated. That is, the according paragraph has to include a `<div lang="en">` language attribute.

A regularly maintained up-to-date list of language codes is available at `http://www.loc.gov/standards/iso639-2/php/English_list.php`.

General Structure, Regions

A concise HTML structure is important in any case, but a well-defined structure is even more important accessibility-wise, for example if people navigate on the Website with assistive technology. Sectioning a page in different landmark roles is a powerful way to support screen reader users to orient themselves and assist keyboard users with quick navigation on the page. Figure 5.6 schematically shows how a Website could be structured.

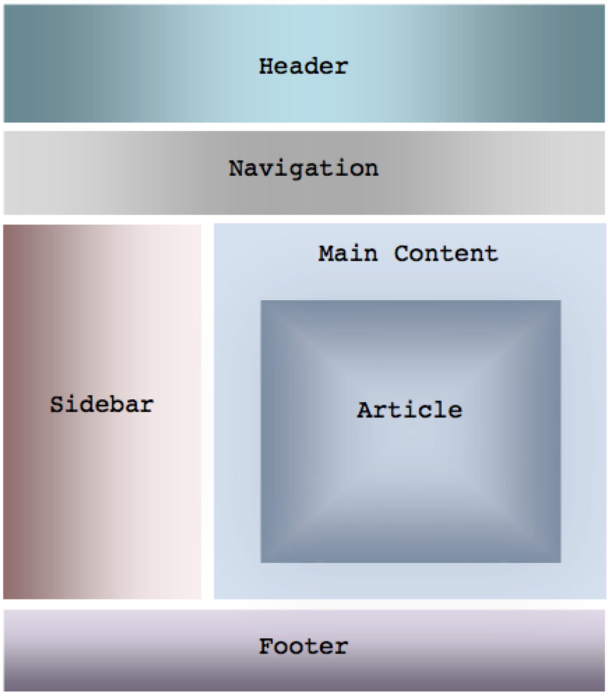


Figure 5.6: HTML Structure

The different sections of Figure 5.6 are described in Table 5.1 and partly could be switched and nested. Additionally, the table shows the default landmark roles of HTML5.

In the past HTML did not offer semantic markers to distinguish between page sections and to get this identification Accessible Rich Internet Applications (ARIA) attributes were used [Con17]. ARIA extends the HTML capabilities with additional attributes, which

HTML5	ARIA Landmark Role	
<header>	role="banner"	defines the header of any part of the page (entire page, <article>-element or <section>-element)
<nav>	role="navigation"	serves as a container for the navigation links
<main>	role="main"	defines the main content block of a page
<footer>	role="contentinfo"	represents the footer area of a page (a <section>, or the <article>-element)
<article>	–	an independent item on a Website (use aria-labelledby or aria-label)
<section>	–	serves as section of a document, to logically group the structure (use aria-labelledby or aria-label)
<aside>	role="complementary"	used to define an independent item on the page seperate to the content (e.g., ads)
–	role="search"	identifies the search section of a Website

Table 5.1: Default landmark roles for HTML5 sectioning and ARIA landmark roles, adapted from [Con17]

increase the Accessibility of Websites through better communication of dynamic content to assistive technologies. The available ARIA landmark roles `role="banner"`, `role="navigation"` and `role="main"` provide additional information for assistive technologies and assist users in identifying the structure of the Website, aid with navigation and improve their orientation. With the release of HTML5 distinguishing different sections in the HTML syntax is possible by using the elements `<header>`, `<nav>`, `<main>` and `<footer>`.

Most of the HTML5 elements automatically create ARIA landmark regions, but still neither all browsers nor all screen readers support HTML5. This is why it is still useful to add the ARIA attributes to the `<div>` elements as a fallback for a browser. HTML5 will be adopted as a standard in the near future, but until full support by browsers and also by screen reader software is available, exceptions for older browsers have to be defined. In general though, if the HTML element already has a defined role, it is not useful to duplicate it with the respective ARIA compliment.

As mentioned above, the schematic presentation in Figure 5.6 is variable, can be nested or rearranged and some of the elements are optional. This means for instance, the

`<nav>` tag could be in more than just one place, for example if sub-navigation in the side-bar is implemented. The *Main Content* can have as many nested `<article>` and `<section>` elements as necessary and an `<article>` element could again include a `<header>` element. But ultimately, too much of everything just adds more complexity and the structure should be kept clear and concise.

Listening 5.2 shows a simplified setup of a Web page structure in HTML.

```
<!doctype html>
<html lang="en">

  <head>
    <meta charset="utf-8">
    <title>The title of the Website</title>
  </head>

  <body>
    <!-- The body element contains the visible content of the Website
        -->
    <header id="page_header">
      <!-- The header usually contains a header image, logo, navigation
            -->
      <nav>
        <!-- main navigation, styled as unordered list -->
        <ul>
          <li><a href="/">Navigation List Items</a></li>
          <li><a href="/">Navigation List Items</a></li>
        </ul>
      </nav>
    </header>
    <main id="main">
      <!-- Main Content - The main element cannot be used anywhere else
            other than inside the body
      element and may be used only once per page. -->
      <nav>
        <!-- The nav element can be used on various places -->
      </nav>
      <article>
        <!-- Blog post or news articles could be wrapped in article
             tags -->
        <aside>
          <!-- The aside tag can be used within or outside an article,
               used to identify content which is independent of the main
               content -->
        </aside>
        <section>
          <!-- Separate major parts of an element -->
        </section>
      </article>
      <aside>
        <!-- The aside element can also be used to mark a sidebar, if
            used outside of the main element, e.g. for ads -->
        <section>
          <!-- Use section within a sidebar to separate the different
              parts of the sidebar. -->
        </section>
      </aside>
    </main>
    <footer id="footer">
      <!-- Footer content - typically something like: About Us, Privacy
          Policy, Contact Us,... -->
    </footer>
  </body>
</html>
```

Listing 5.2: Example for a HTML structure

Implementing the Navigation

Another orientation setting, for seeing users as well as for non sighted users, on a Website is the main navigation. There can be of course more than one navigation element on a Website and most rules apply to all of them. However, the difference from the main navigation is, that it should not change throughout the Website.

The navigation should be marked accordingly with the `<nav>` element or the `role="navigation"` landmark to make it recognizable for assistive technology. Common and best practice for the navigation markup is to implement the navigation itself as link elements via an unordered list `<ul>`. This way the screen reader software reads it accordingly and a user recognizes the elements as the navigation. In this case keyboard tabbing through the elements is possible and the semantics are clear even if the CSS loading fails.

Placing the navigation at the top of the page or below a header image, enables the UX getting quick orientation and recognition for sighted users and also advances interactions with the site. Since keyboard users usually navigate the HTML structure elements from top to bottom, they don't have to peruse the whole page for just navigating to a sub-site. Thus, it is common practise to organize the source code according to the reading order.

For easy orientation the active navigation item should be highlighted. Though, not only color, but at least with an additional indication should be used, to make sure visually impaired users can recognize it. Using an underline, increasing the size or highlighting the element to raise attention are all valid options. Figure 5.7 shows a neutral possible option. The figure just acts as visualization and does not depict one particular recommended style.

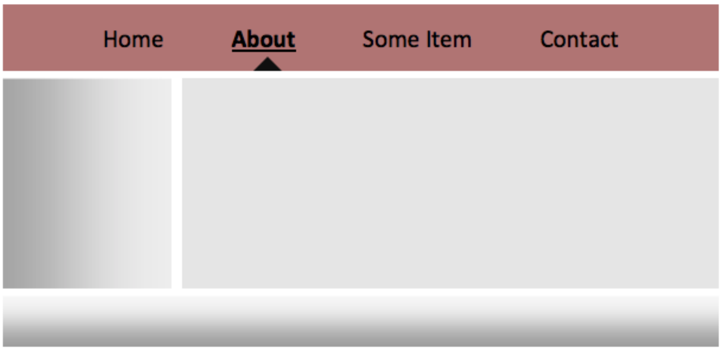


Figure 5.7: Main navigation with an active item

To indicate the active item for a screen reader a pure visual implementation is not sufficient. Thus, an indication for the software, which can be read, needs to be applied to the active element. There are different options available, for example via the `aria-current="page"`, `aria-describedby="current"` attribute or by adding a hidden element within the link `<span class="visually-hidden">Current page</span>`, to realize the correct behaviour. Listing 5.3 shows an example HTML navigation markup.

```
<nav>
  <ul>
    <li><a href="/">Home</a></li>
    <li><a href="/about" aria-describedby="current">About</a></li>
    <li><a href="/something">Some Item</a></li>
    <li><a href="/contact">Contact</a></li>
  </ul>
</nav>
```

Listing 5.3: Basic Navigation Structure with active element in HTML

When choosing the option with the additional `<span>` element, it is important to note, that `display: none;` and `visibility: hidden;` is also hidden from screen reader software. Since a visually hidden label is sometimes needed an example CSS code is given here in Listing 5.4 [Pic16]. For focusable content like links or buttons, it is important to highlight them, for keyboard users, on focus. In this case the second rule from the listing would apply.

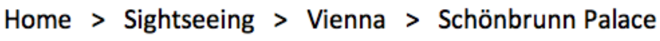
```
.visually-hidden {
 position: absolute;
 width: 1px; height: 1px;
 overflow: hidden;
 clip: rect(1px, 1px, 1px, 1px);
}

.visually-hidden a:focus,
.visually-hidden input:focus,
.visually-hidden button:focus {
 position: static;
 width: auto; height: auto;
}
```

Listing 5.4: CSS for visually hidden content

To comply with the WCAG guidelines more than just one way of navigating the site is necessary. This can be achieved by providing a search and/or a site-map. The site-map provides an overview of the entire Website containing links to various sections.

Implementing breadcrumbs also supports the user when orienting herself/himself on a Website and facilitates a quick navigation within sections. This is especially recommended for Websites with deep hierarchy levels. The breadcrumb trail displays the path of the currently displayed page within a Website, schematically visualized in Figure 5.8, whereat the separation element of the single link items is up to the designer. In this example the ">" symbol is used. The placement of the breadcrumb path is ideally on top of the main content.



Home > Sightseeing > Vienna > Schönbrunn Palace

Figure 5.8: Sample Breadcrumb path

For the implementation the `<nav>` element with an aria-indication is used:

```
<nav aria-label="Breadcrumb">
```

Within the `<nav>` element the breadcrumb path is again implemented as an unordered list containing the link elements to each page of the hierarchical tree [Con08]. The current active page link uses the `aria-current="page"` indication.

Navigating via keyboard

There are various reasons for users not to use a mouse while interacting with a Website and some other commonly used input options are keyboards or any specifically modified input device. These devices are used by users with limited fine-motor skills who have trouble using a mouse, blind users, or by any other person who is (temporarily) handicap and prevented from using a mouse. Typical control values for navigation on a Website via the keyboard are the *TAB* respectively the *SHIFT + TAB* keys. These are the most common and important ones to navigate back and forth single elements. The *Enter*, *Space*, *Esc* and the *Arrow* keys are needed for navigation as well as to access features. Further information on how the keyboard is used in given in Section 5.1.4.

For an efficient use, the user also needs orientation, meaning she/he needs to know where she/he is. Thus, a keyboard user needs a visual indication on where she/he is respectively and which element is active or better where the focus is on. Browsers are offering these indications by default, when a user navigates interactive elements like *links, buttons, input-fields, ...*. These indications are called `outline` and are visualized as border around the element. Unfortunately, the default browser settings can be disabled via CSS, using `outline:none` or `outline:0`, which a lot of Web designers do, probably without knowing the implications for some users. Due to the fact, that the style of the outline can be modified, there is also an option for developers to implement their own focus-styling. So, a style according to the design can be implemented and the visual appearance of the elements can still be supported. The creativity on how to style the elements on focus is up to the designer and should just be a visual highlighting of the focused element, like a border or background color. The implementation is done via the `:focus` pseudo class in CSS and can be applied to interactive elements like `a:focus, input:focus, button:focus`.[4]

The same applies to blind users, they just don't orientate on the visible indications, but use screen readers. Thus, proper naming of the elements is essential.

By default the interactive elements can also be navigated via keyboard. If custom elements are used, the possibility to navigate them has to be provided by using the `tabindex="0"` attribute on the elements [fPwD17c].

tabindex="0" Elements which are usually not focusable are added to the logical tab order based on the placement within the source code.

tabindex="-1" A negative value means, that the element is not included in the tab order, but programmatic focus is given.

tabindex="1+" Any positive integer may be used and thus defines the tab order explicitly. It is bad practise to use this argument, because it usually leads to confusion and illogical tab order.

Keyboard users navigate through the structure of a Website until they find what they are looking for. Thus, as already mentioned before, it is important that the Website follows an intuitive flow - this regards the source code structure. Everything has to be in a logical and intuitive order, typically from the top to the bottom and from left to right.

[4] A full list of interactive HTML elements is available at `http://w3c.github.io/html/dom.html#interactive-content`, accessed Jun 2017

Due to that fact, that it is very inconvenient to always navigate through the entire page before reaching the desired content, a bypassing mechanism would be in favour of these users. Looking, for example, at a popular Austrian news site[5], it can be seen, that it is not optimal for keyboard users to get to the main content - the news articles - while running-through the navigation menu. While testing this Website, it is noticeable how important a focus styling and a tab order is. During navigation with Firefox, which only uses a very light outline, it is very hard to keep orientation.

A solution for quickly bypassing other sections and jumping to the main content are so called skip links [fPwD17c]. These links are typically provided at the top of the page and enable directly jumping to a defined area on a page. Since the links don't need to be visible, they don't conflict with the design or lead to irritation. The basic idea is to add links on top of the page, which are `visually-hidden` via CSS, and show the links on focus in the left upper corner. The placement of the link is up to the developer, but placing it somewhere at the beginning of the site makes the most sense.

An example for such links is shown in Listing 5.5.

```
<--! anchor links to the navigation and the main content, references
   the id of <nav> resp. <main> -->
<a href="#navigation" class="visually-hidden">Skip to main navigation
   </a>
<a href="#main-content" class="visually-hidden">Skip to content</a>

<--! style.css -->
a.visually-hidden:active,
a.visually-hidden:focus,
a.visually-hidden:hover {
   position: static;
   left: 0;
   top: 0;
   width: auto;
   height: auto;
   overflow: visible;
}
```

Listing 5.5: Implementing a skip-link navigation

Document Outline

Although the Website is structured well based on landmark and sectioning elements, a logical and precise hierarchy of headings should still be implemented by the proper usage of the corresponding tags. `<h1>` defines the top level heading and `<h6>` defines the heading of least importance.

Each page should have exactly one main or first level heading `<h1>`, while the other heading tags could be used more often. Screen reader software like NVDIA and Jaws

[5] http://www.derstandard.at as of Jun 2017

provide the function to directly jump to the <h1> with the “1”-key on the keyboard. It is recommended to not leave out a level heading and build a clear hierarchical structure. Headings on the same level should also be tagged accordingly.

For assistive technology and also for SEO reasons, it is important to use the heading tags and not just add different heading styles to the CSS. Screen reader users can use defined keys to quickly navigate through the page from heading to heading by simply using the “h” key on the keyboard. Search engines like Google rank headings higher, which is a good marketing argument for having a proper document outline with correct syntax and semantic.

Links & Buttons

Links on Websites are basic elements and considerations, only in terms of an accessible implementation, are of minor relevance. A separate CSS class to identify external links is necessary and opening external links in a new window is accomplished by adding the `target=_blank` argument to a link element.

It is important to provide meaningful text to all links and buttons, because this is what the screen reader software reads. However, this is an editorial task and described in detail in Section 5.2.3.

Integrate Multimedia

Multimedia includes all different content forms like text, audio, images, video, et cetera. Knowing how to correctly include them in a Website, so that also people with different kind of disabilities can profit from them, is a very basic rule. Still, there are a lot of Websites, which don’t even provide basic implementation. A commonly missing Accessibility feature is descriptive alternative text in the `alt` attribute of images.

The alternative text should act as a textual description of the image and give the reader an impression of what the image is about [fPwD17c]. But there are more arguments for choosing the alternative text wisely. If, for any reason, images cannot be load by the browser, the alternative text is shown to all users. And furthermore, the Google image search is also using the alt-attribute as a ranking factor and classify the images accordingly.

When choosing the correct text for the alt-attribute the following issues should be considered:

- The basic idea is to provide a text alternative presenting the same information as the non-text content - a substitution should be possible, and still communicate the same information.
- The alt attribute should be accurate and describe the same content and function which is visible on the image.

- It should have an appropriate length - as short as possible, but also as informative as needed. In general a few words are sufficient to describe the image in a way that one can imagine what the image depicts.
- The alt-text should not duplicate the content around the image. If the surrounding content describes the image accordingly, the alt-attribute might be very short or even empty.
- Screen reader software tells the user, that the element is an image. Thus terms like "image of ..." or something similar in the alt-description should be avoided.
- Questions like "Why is this image here?", "What information is it trying to communicate?" or "How would I describe the same information with words" could help to define the alternative text.

Although the implementation of alternative text for images is easy to add programmatically, there are still many wrong realizations, which is one of the most frustrating facts for screen reader users [LAKM07]. Deciding whether an image should get an alternative text or not and then describing the image in an appropriate way, is not an easy task [PHD05]. Knowing that an alternative text is necessary, but not knowing the details about how and why it is used, leads to useless information. Still, very often the file name of the image is used as an alternative text (e.g., *IMG_0815.jpg*, *image1.jpg*, ...). In other cases the alternative text just duplicates information which is already present around the image anyway. Hearing the same information several times is very frustrating for screen reader users. Also the distinction on how long an alternative text should be is misunderstood by a lot of Websites [LAKM07]. CNN[6] for example has either empty alt-attributes or very long descriptions, which often don't represent the content of the image at all. Figure 5.9 shows an example for such an alternative text.

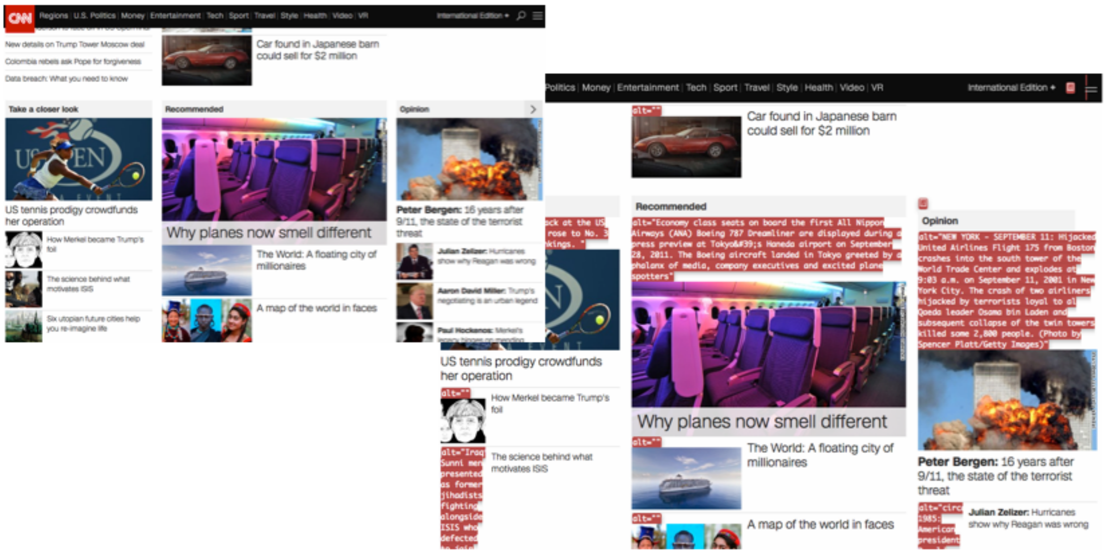
The figure shows a comparison of what is actually seen and the corresponding alt-attributes. For example, an image displaying the burning twin-towers of the World Trade Center uses the alternative text show in Listing 5.6.

```
alt="NEW YORK - SEPTEMBER 11: Hijacked United Airlines Flight 175
   from Boston crashes into the south tower of the World Trade Center
    and explodes at 9:03 a.m. on September 11, 2001 in New York City.
    The crash of two airliners hijacked by terrorists loyal to al
   Qaeda leader Osama bin Laden and subsequent collapse of the twin
   towers killed some 2,800 people. (Photo by Spencer Platt/Getty
   Images)"
```

Listing 5.6: Example for a too long unrelated alt-attribute

For sighted users this text is not seen anywhere - also not in the article itself - and the text does not reflect or represent what is depicted by the image. Additionally, the length is inappropriate.

[6] `http://edition.cnn.com/`, accessed Sep 2017


CNN
Regions | U.S. Politics | Money | Entertainment | Tech | Sport | Travel | Style | Health | Video | VR
International Edition
Take a closer look
Recommended
Opinion

New details on Trump Tower Moscow deal

Colombia rebels ask Pope for forgiveness

Data breach: What you need to know

Car found in Japanese barn could sell for $2 million

US tennis prodigy crowdfunds her operation

How Merkel became Trump's foil

The science behind what motivates ISIS

Six utopian future cities help you re-imagine life

Why planes now smell different

The World: A floating city of millionaires

A map of the world in faces

Peter Bergen: 16 years after 9/11, the state of the terrorist threat

Julian Zelizer: Hurricanes show why Reagan was wrong

Aaron David Miller: Trump's negotiating is an urban legend

Paul Hockenos: Merkel's legacy hinges on mending

alt=""

alt="Economy class seats on board the first All Nippon Airways (ANA) Boeing 787 Dreamliner are displayed during a press preview at Tokyo&#39;s Haneda airport on September 28, 2011. The Boeing aircraft landed in Tokyo greeted by a phalanx of media, company executives and excited plane spotters"

alt="NEW YORK - SEPTEMBER 11: Hijacked United Airlines Flight 175 from Boston crashes into the south tower of the World Trade Center and explodes at 9:03 a.m. on September 11, 2001 in New York City. The crash of two airliners hijacked by terrorists loyal to al Qaeda leader Osama bin Laden and subsequent collapse of the twin towers killed some 2,800 people. (Photo by Spencer Platt/Getty Images)"

alt="Iraq Sunni men presented as former jihadists fighting alongside ISIS who defected

alt="circ 1985: American president

ck at the US rose to No. 3 kings. "

Figure 5.9: Example for alternative text from CNN as of Sep 2017

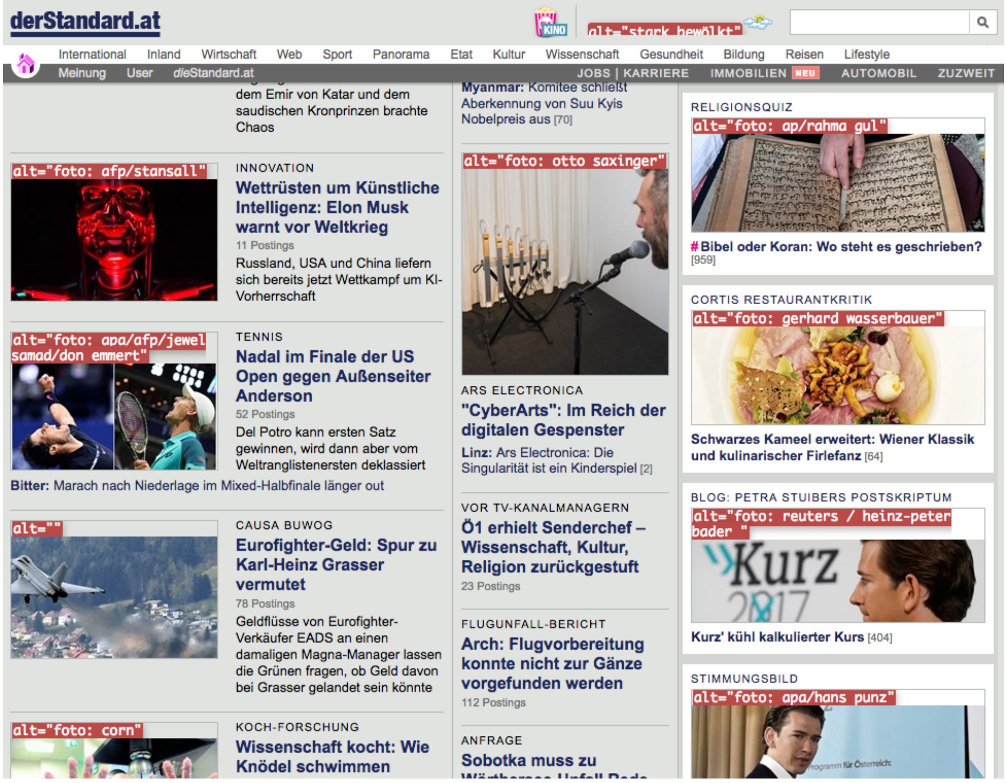
Some basic rules for images adjusted from [fPwD17c]:

- Every image must have an `alt` attribute!
- Decorative images also need an empty (`alt=""`) alt-attribute.
- The alternative text should describe the content and the function of the image.
- The alternative text depends on the context the image is placed in and redundancies should be avoided.
- The surrounding context could also act as alternative text.

A different example, again from the Austrian news site derstandard.at[7], is shown in Figure 5.10.

As of September 2017 the copyright information of the image is the alternative text on derstandard.at's Website. This might help in terms of legal copyright issues, but it is not the correct place to add this information. It can easily be conceived how frustrating it is to hear the copyright information for every image. Though this seems to be a common practice along news sites, it is still not good practice. The fact that derstandard.at even duplicates the copyright information in the alt- and the title-attribute is even worse. Some screen reader software might even read the same information twice, and still the user does not get the information she/he wants to obtain. Another practice commonly used is to just copy the headline of the article into the alt-attribute of the corresponding image

[7] `http://www.derstandard.at/`, accessed Sep 2017



dem Emir von Katar und dem saudischen Kronprinzen brachte Chaos

alt="foto: afp/stansall"

INNOVATION

Wettrüsten um Künstliche Intelligenz: Elon Musk warnt vor Weltkrieg

11 Postings

Russland, USA und China liefern sich bereits jetzt Wettkampf um KI-Vorherrschaft

alt="foto: apa/afp/jewel samad/don emmert"

TENNIS

Nadal im Finale der US Open gegen Außenseiter Anderson

52 Postings

Del Potro kann ersten Satz gewinnen, wird dann aber vom Weltranglistenersten deklassiert

Bitter: Marach nach Niederlage im Mixed-Halbfinale länger out

alt=""

CAUSA BUWOG

Eurofighter-Geld: Spur zu Karl-Heinz Grasser vermutet

78 Postings

Geldflüsse von Eurofighter-Verkäufer EADS an einen damaligen Magna-Manager lassen die Grünen fragen, ob Geld davon bei Grasser gelandet sein könnte

alt="foto: corn"

KOCH-FORSCHUNG

Wissenschaft kocht: Wie Knödel schwimmen

Myanmar: Komitee schließt Aberkennung von Suu Kyis Nobelpreis aus [70]

alt="foto: otto saxinger"

ARS ELECTRONICA

"CyberArts": Im Reich der digitalen Gespenster

Linz: Ars Electronica: Die Singularität ist ein Kinderspiel [2]

VOR TV-KANALMANAGERN

Ö1 erhielt Senderchef – Wissenschaft, Kultur, Religion zurückgestuft

23 Postings

FLUGUNFALL-BERICHT

Arch: Flugvorbereitung konnte nicht zur Gänze vorgefunden werden

112 Postings

ANFRAGE

Sobotka muss zu

RELIGIONSQUIZ

alt="foto: ap/rahma gul"

# Bibel oder Koran: Wo steht es geschrieben? [959]

CORTIS RESTAURANTKRITIK

alt="foto: gerhard wasserbauer"

Schwarzes Kameel erweitert: Wiener Klassik und kulinarischer Firlefanz [64]

BLOG: PETRA STUIBERS POSTSKRIPTUM

alt="foto: reuters / heinz-peter bader "

Kurz' kühl kalkulierter Kurs [404]

STIMMUNGSBILD

alt="foto: apa/hans punz"

Figure 5.10: Alt Attribute sample from derstandard.at, as of Sep 2017

- which again does not make any sense at all. In this case even an empty alt-attribute would be more appropriate.

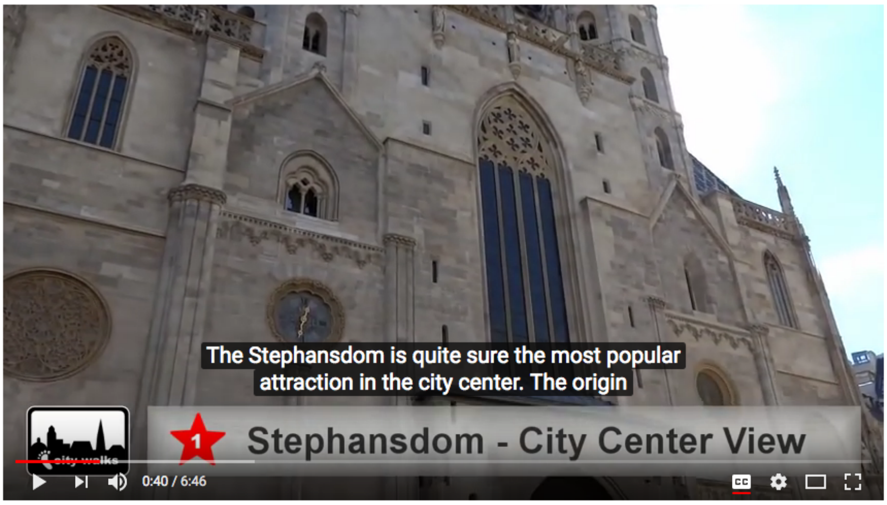
As demonstrated, alternative attributes are not an easy task and a lot depends on the subjective opinion of the user or respectively the one who is testing the site. But with some background information on images and their alternative text, their application could be improved and make a difference.

Audio or video is another type of multimedia content that is commonly integrated on Websites. The implementation of the players and also the accessibility requirements are similar. To fully comply with the WCAG guidelines for video and audio content is not easy, since some of the guidelines can be tricky to fulfill. It is up to the customer's exigencies which level of the guidelines she/he wants to satisfy.

For any audio or video content a transcript is required. A transcript builds a textual version of the spoken content and is accessible by anyone. A transcript is not necessarily literal, but also contains additional information on noises, which can either only be heard or are visually noticeable, as for example laughter, gun shots or waves at a sea. The transcript also has the advantage, that it is searchable for both humans and machines. That is, not disabled people can also use the transcript to search for special content.

However, it is particularly helpful for machines as screen readers can access and read it for the user and search-engines have SEO advantages.

Closed Captions (CC) should be provided additionally to the transcript captions. Captions are text displayed in an overlay on the video content and show what is spoken in the video. Usually the text is shown against a dark background so that it has better reading quality. Captions are always visible, whereas Closed Captions can be toggled on and off. Figure 5.11 shows an example of a video player with captions.



The Stephansdom is quite sure the most popular attraction in the city center. The origin

Figure 5.11: Screenshot from a youtube video, showing Closed Captions[8], about the Top City Attractions in Vienna

In general embedding the youtube player[9] is a good idea, since the youtube player offers great Accessibility support. As seen in the menu band in Figure 5.11, the youtube player offers all the necessary controls. The primarily needed controls concerning Accessibility support are the *play/pause* button, *volume control and mute option* as well as the *Closed Captions* button. These are the basic options a video and audio player should provide. Nice add-ons are the options to jump forward to scenes at will, to view the video in *full screen* and to change the general *settings*.

For full Accessible support an audio description, though this is optional if good captions and a transcript is offered, should be provided as well. The audio description describes what is seen in the video. For videos produced with Accessibility in mind and if a good transcript is available, then the audio description can be neglected.

[8] Video provided on `https://www.youtube.com/` by the Vienna Travel Agency: `https://youtu.be/9B9pO2Yqeb8`, accessed Jun 2017

[9] `https://developers.google.com/youtube/player_parameters`, as of Jul 2017

Form Implementation

Forms are a pervasive and prominent used feature on Websites. Almost every Website has some sort of form implemented, even if it is just a search or newsletter form. Although forms are so commonly integrated, they still trouble many developers and there are many implementations that are not ideal out there. With increasing complexity of the form the challenge to implement them accessible rises.

Primarily the form implementation concerns people using a screen reader or keyboard navigation, but also people with cognitive disabilities can benefit from an improved form. Making a form accessible improves not only how it is presented or used via screen reader software, but also improves the layout structure and provides better feedback to users [SHBA+14]. According to the research of Lazar et al. [LAKM07], using forms, especially the incorrect labeled of forms, is one of the most frustrating online tasks.

First of all, labeling the form controls, for example *input fields, textbox, checkbox, radiobox,...*, correctly is the first step on creating an accessible form. Thus, the `<label>` element should be related to the corresponding form control. This text label describes the form control and has to be placed in context and visual correlation of the form control. But also a non-visual assignment has to be realized using the appropriate markup. That means, that each form control needs an analogous `<label>`, which is associated with the `id` attribute of an form control. Listening 5.7 shows an example for a labeled form control.

```html
<label for="name">Name</label>
<input type="text" name="name" id="name">

<input type="checkbox" name="subscribe" id="subscribe">
<label for="subscribe">I have read and accept the Policy.</label>
```

Listing 5.7: Sample associations between labels and controls in forms

Adding this proposed basic markup guarantees, that the form is keyboard accessible and also that a screen reader software will process the form properly. A correct labeling is also a valuable Usability feature for everyone, since most of the advanced users use the keyboard navigating while working on a form. Of course, the label should be concisely, descriptive and easy to understand, meaning there should not be any confusion for the user about what to enter in the field. Also, mandatory fields should be marked accordingly. Not just by placing an asterisks (*) next to the label, but also by explaining that fields marked with an asterisks are required. Furthermore, the indication for mandatory fields needs to be visible for the screen reader software. This can be done with the HTML5 element `required` and also with the ARIA label `aria-required="true"`. Since not all screen readers recognize HTML5 elements, it is saver to add both elements to a required input field.

Having form elements which belong semantically together should be grouped by the `<fieldset>` and `<legend>` elements. The `<fieldset>` groups the elements and the

<legend> adds a description to it. This is typically used to group related radio buttons as shown in Listing 5.8. Other form controls, for example the shipping or billing address fields, can be grouped in the same way.

```
<fieldset>
<legend>Select your payment method</legend>
  <input id="visa" type="radio" name="payment" value="visa">
  <label for="visa">Visa</label><br>
  <input id="mastercard" type="radio" name="payment" value="mastercard
      ">
  <label for="mastercard">MasterCard</label><br>
  <input id="paypal" type="radio" name="payment" value="paypal">
  <label for="paypal">PayPal</label>
</fieldset>
```

Listing 5.8: Semantically related elements grouped in HTML

For descriptive or additional displayed information, which would be too much for a label, the already mentioned aria-describedby attribute can be used. An example would be providing additional information on the password policy, meaning how a valid password should look like when the it is set. Listing 5.9 shows an example for adding an additional label to a password field. Furthermore, for hidden labels the aria-label or even the title elements could be used.

```
<label for="enterpassword">Enter Password</label>
<input type="password" name="enterpassword" id="enterpassword" aria-
    describedby="passpolicy">
<br>
<span id="passpolicy">Your password may be any combination of minimum
     8 letters, numbers and special characters.</span>
```

Listing 5.9: Adding an additional label to an field

In recent times, the placeholder attribute is often used instead of a label. This is not only not an Accessible solution, but is also not recommended in terms of Usability [She14]. Users, who quickly run through a form, may not remember all the placeholders after filling out the form. The users forget the context and cannot properly double-check the inputs since they are missing the related information belonging to the field. Furthermore, in case of appearing errors in the form, the original information, which was given in the placeholder, is missing and thus makes it harder to fix mistake.

Generally, there are rules to follow in terms of accessible error handling. No matter if the error is detected via client-side or server-side validation, a proper error handling is necessary. First of all, the user needs to be informed about the occurrence of an error. Moreover, a proper indication on where the error arouse and a possibility to resubmit the corrected form has to be available. The specific implementation of this procedure does not greatly matter, as long as the rules are applied. The error message should be displayed in context of the incorrect field, either directly before, next or after the field.

Another option is to display all errors on top of the form and link each error to the corresponding form field. Either way, it has to be clear which field is incorrect and the error message has to be descriptive, meaning it should give clear instructions on how to correct the mistake. Furthermore, the `aria-invalid="true"` element should be added to the corresponding field.

Not only errors should be made clear, but the user should also be informed about a successful completion of a form. A short note, that the form has been submitted successfully, is sufficient. The note should include information on the next steps, e.g. that a confirmation email has been sent.

Beyond that, an accessible CAPTCHA, as already defined in the *Design* phase and stated in Section 5.1.2, has to be implemented accordingly.

The importance of the search box, which is usually implemented as a form, has already been discussed in Section 5.1.2. However, it is essential that a correct implementation is carried out. There are many examples of poorly integrated search boxes and keyboard users, alongside users utilizing assistive technologies, get stuck in forms or cannot reach them at all. It is highly important to use native elements and style them via CSS accordingly. Self made markup, which assistive technology is not able to read, should not be used. Applying the correct ARIA element `role="search"` to the form indicates the proper landmark. A simple example on how such a markup looks like is given in Listing 5.10. As already mentioned, the CSS styling should be applied as well as the optional visually hiding of the label.

```html
<form role="search">
  <label for="search">
   Search the site <p id="error" role="alert"></p>
  </label>
  <input id="search" type="text" placeholder="Searchterm">
  <button type="submit" value="Search">Search</button>
</form>
```

Listing 5.10: Accessible Search Form

Lists

Studies [Nie14] show how little information of the overall content users extract and what patterns they use. Thus, a solid structure, where much information is shown in a compact matter, is a plus concerning the Usability of the Website. Overall, one of the main goals is, that the content is received by the user. Lists are an appropriate method to clearly structure and present information. Using lists also has the advantage, that screen reader software can easily access the elements if they are implemented with proper markup. This is similar to the approach used for menus shown in Section 5.1.3.

One important note on lists is to use the proper HTML elements `<ul>` for unordered lists and `<ol>` for ordered lists. The single list items are marked with the `<li>` element.

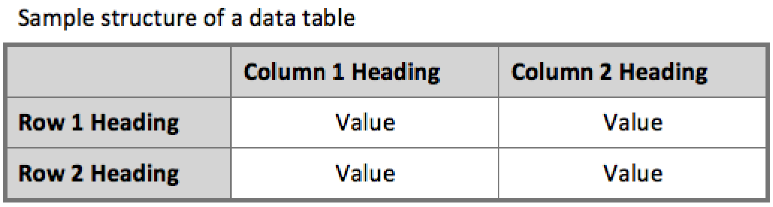
Using Tables

Basically, the use of tables can be split into two dimensions: data tables on the one hand and layout tables on the other hand. The main intention of HTML tables was to show data in a tabular presentation. Due to easier styling of the layout, the table markup was gradually adopted for layout purposes. In terms of Accessibility a table layout is not the best solution, though, if properly implemented, and with correct reading order ensured, it is less of a problem for assistive technologies. However, with the recent responsive design alternatives and the problem of table layouts not being adaptive, they becomes less important anyway. New JavaScript libraries and CSS technologies replace the table layout and provide flexibility with a clean markup, excellent responsive behaviour and less Accessibility issues.

In contrast, data tables are still commonly used for presenting information organized in data grids. Adding the correct markup indicating the structure of the table and the content relationships enables assistive technologies to properly present the table content to the user.

For developers it is important to know that the screen reader software reads the content of tables in a linear matter. The source code of tables is read line by line, including additional information about headers. Thus, it is important to structure the code in a logical order. By doing so, the content is presented in a reasonable way and can also be recognized by blind users. Tools can help a developer to visualize the linear reading order (see Section 5.5).

An example of a basic data table is shown in Figure 5.12.



Sample structure of a data table

	Column 1 Heading	Column 2 Heading
Row 1 Heading	Value	Value
Row 2 Heading	Value	Value

Figure 5.12: Basic presentation of a HTML table

A HTML markup presentation of this data table is shown in Listing 5.11. Note that no styling of the table is included as the markup only illustrates the basics.

The basic HTML elements that are used for proper Accessibility support are <tr> indicating a row, <th> elements for table headers and <td> elements representing a cell. The <caption> element is used to add a short description to the table. Distinguishing between row headers and column headers can be accomplished with the scope attribute. For a correct markup of more complex tables please refer to the Web Accessibility Tutorials on tables[10].

[10] https://www.w3.org/WAI/tutorials/tables/, accessed Jun 2017

```
<table>
  <caption>Sample structure of a data table</caption>

  <tr>
    <th scope="col">Column 1 Heading</th>
    <th scope="col">Column Heading</th>
  </tr>

  <tr>
    <th scope="row">Row 1 Heading</th>
    <td>Value</td>
    <td>Value</td>
  </tr>

  <tr>
    <th scope="row">Row 1 Heading</th>
    <td>Value</td>
    <td>Value</td>
  </tr>
</table>
```

Listing 5.11: Data Table

5.1.4 Testing

Accessibility testing, as a subset of Usability testing, means to test all requirements and user stories that where defined at the beginning of the project and their corresponding implementations. An expert user, i.e. an Accessibility expert, typically performs or supervises this step. Due to the fact, that this thesis focuses on the integration of Usability and Accessibility, only the parts covering these areas are included in the *Testing* phase.

Web Accessibility testing means to check the ease of use of a Website or discover potential user problems. Its goal is to formulate new requirement specifications for the identified issues and thus improve the product with each process-cycle. Ideally, the requirements from the *Concept, Specifications & Requirements* phase of Section 5.1.1 already include corresponding test cases focusing on Accessibility. Although definitions and specifications made in the *Concept, Specifications & Requirements* phase probably do not cover all issues, they provide a first indication on what Accessibility standards should be tested against. In addition, being able to test with “real” (disabled) users, in comparison to expert tests, might uncover additional issues.

Since user tests are very expensive, the focus in many test cycles is on expert tests and user tests are often only performed near the end of the development-cycle of a product. However, expert tests can already reveal a lot of problems and save money if done frequently.

Expert testing can be divided in three components:

Tool-based evaluation Testing a Website with tool support (see Section 5.5) can reveal a lot of issues. However, if this evaluation method is used standalone, it is not sufficient for Accessibility testing.

Simulation Simulation means, that the expert user emulates the experience of a disabled person, e.g., using assisitive technology. As an example, an expert user might turn of the screen and uses screen reader software to test a Website.

Code review In code reviews the source code is directly checked for potential problems. Generally this is done with tool assistance or in combination with the tool-based evaluation.

Most of the time, though not in each cycle of the process, testing uses a combination of all three components. Especially at the beginning of a project the simulation with using assistive technologies would be too much effort for the tester. Using assistive technology is not an easy task and requires extensive training and experience. It has to be noted, that it makes a difference if a screen reader software is used while being able to see the site versus not having a visual representation of what is going on. For that purpose, or rather if the target group includes blind users, it is highly recommended to do user tests with blind users.

Testing should generally cover the the four WCAG principles: *Perceivable*, *Operable*, *Understandable* and *Robust*. Most automatic tools perform checks based on the *POUR* principle and they are recommended for manual testing as well.

Perceivability All information and components has to be presented in a way that allows users to recognize it. Problems concerning the perceivability are commonly related to media content, like providing proper text alternatives to any kind of integrated media, or the use of color. In addition, the semantic structure and proper markup of the HTML code is covered in this principle.

Operability An interaction with all components and navigation elements has to be possible, including access via keyboard functionality. The navigation options are covered in this principle as well. Furthermore, it is checked whether disallowed time limitations are utilized or if flashing content is in operation.

Understandability All information and operations with the UI have to be understandable. This principle checks the consistency of elements and also includes tests on forms.

Robustness The content has to be robust to be interpreted reliably by various assistive technologies. This principle involves validation tools for technologies as well as manual review code. Overall, the proper usage of native controls is checked.

User testing is not as easy as it appears to be and a proper preparation is indispensable. User tests cannot cover all features, they typically do not include the entire Website, because the test situation is extraordinary and exhaustive for the test users. The following steps should be taken into consideration:

- A test protocol should be prepared.
- Specific tasks for the test user to accomplish, should be formulated.
- The "Thinking aloud" approach should be applied. The Accessibility tester has to listen carefully and take notes on what the test user states.
- It has to be precisely observed what the test user does and where she/he has problems.
- The test user could be asked on her/his first impression, what she/he thinks about the page and what she/he thinks the intention of the Website is.

This phase does not cover particular rules and guidelines as they would be a repetition of the previously presented design and implementation practises. The main focal point is to show the importance of testing the user stories and requirements defined in previous phases and to cover the implemented features. Manual and tool supported tests are combined to check for abnormalities and critical parts.

Keyboard Use

Keyboard-only tests are a fundamental activity during an Accessibility evaluation. In such tests the keyboard is used as the sole input device. Important to know is that the forward navigation works with the *Tab* key and moving back is accomplished by selecting the *Shift + Tab* keys. Selections are made with the *Enter*-key or sometimes with the *Spacebar*. These basic keyboard commands enable everyone to perform basic tests to determine whether a Website is keyboard accessible or not.

Using the keyboard to evaluate a Website, allows to check the existence of *skiplinks*. In a correct implementation the hidden skiplinks should be visible on the first tab-interactions. Following the outline of the tabbed elements, the logical tab-order can be checked. By doing so, it can be quickly observed, that someone can get lost quite easily if the `outline` is not presented or `focus` styles are missing. If a sighted user gets lost on a Website, a user using assistive technology probably does have an even smaller chance to logically navigate through the site. Another point to check is whether all elements can be reached via the keyboard or if there are any traps a user gets caught in.

Table 5.2 shows the most common keyboard interactions on a Website. The commands shown in this table are needed to test the basic functionality. A full list of keyboard commands is provided by the W3C[11].

Keystroke	**Interaction**
Tab / Shift + Tab	Navigate the elements forward and backwards
Enter / Spacebar	• Follow a link • Confirm a button • Select / De-select checkboxes • Expand dropdown
Arrow keys left / right	• Horizontal scrolling • Select radio-buttons • Browse through slideshow
Arrow keys up / down	• Vertical scrolling • Access menu items • Approach items of a dropdown menu
Esc	Close dialog box
Alt + left arrow key	Back to previous page

Table 5.2: Basic keyboard interactions on a Website

5.2 Accessibility Considerations for Content

When entering the content, for example via a CMS, there is a lot to consider for the *Content Manager* stakeholder. Thus, it is important to train the *Content Managers* accordingly. All the technical infrastructure and proper implementation is meaningless if the content is added inadequately. Therefore, to keep a high level of Usability and Accessibility, it is not sufficient to have a clear and comprehensive concept with an adequate implementation, but the people providing the content for the Website need to follow particular rules as well.

[11] `https://www.w3.org/TR/wai-aria-practices/`, accessed Jul 2017

Often a CMS provides a so-called What you see is what you get (WYSIWYG)-editor which enables the *Content Manager* to enter content. Typically, these editors do not have special Accessibility support built-in and additional support for Accessibility can be implemented via plug-ins by developers. TinyMCE[12] is a commonly used What you see is what you get (WYSIWYG)-editor that provides several options to insert content. Figure 5.13 shows the formatting panel, including its options, of the TinyMCE editor.

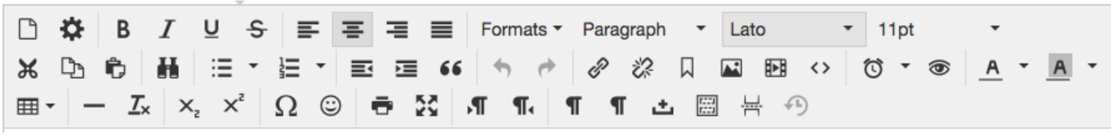


Figure 5.13: Format options of the TinyMCE Editor

5.2.1 Formatting Content

Although formatting content/text seems straightforward, there are some guidelines to consider that help to make the content easier to read and understand. A common error among editors is, that the content and its formatting is done in Microsoft Word or a similar text editing tool. Per se this is not bad, but when the text is copied and pasted into the WYSIWYG-editor of the CMS often times the formatting is lost. If this is not recognized by the editor, then the text will not be fully accessible for the users. Alternatively, the text could be entered directly in the CMS or the editor needs to be able to convert the content and its format when it is copy-pasted.

Another problem with content preparation in external tools is, that the text is often not separated in different paragraphs. Commonly paragraphs are created artificially by multiple line breaks. Not using paragraph style is not only poor design, but also could disturb screen reader users.

The code samples of Listing 5.12 and Listing 5.13 show the differentiation between a paragraph styling and an artificial paragraph with line breaks. The visual result is in both cases the same, but for screen readers it makes a huge difference. A further advantage of using paragraphs is easy styling adaptions via adding CSS classes.

```
Some text.<br/>
Text in a new line.
```

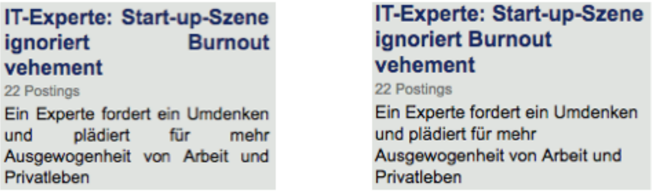
Listing 5.12: Artificial “paragraph” via line-breaks

```
<p>Some text.</p>
<p>Text in a new line.</p>
```

Listing 5.13: Use of paragraph style

[12] https://www.tinymce.com/

Text alignment has also to be considered when entering content. If text is not justified it is much easier to read. Justification fills the gaps with white-spaces and requires more concentration to read it [Pic16, page 53ff]. For Web content it is recommended to use a left to right alignment. Figure 5.14 gives a comparison of text alignments.



IT-Experte: Start-up-Szene ignoriert Burnout vehement

22 Postings

Ein Experte fordert ein Umdenken und plädiert für mehr Ausgewogenheit von Arbeit und Privatleben

IT-Experte: Start-up-Szene ignoriert Burnout vehement

22 Postings

Ein Experte fordert ein Umdenken und plädiert für mehr Ausgewogenheit von Arbeit und Privatleben

Figure 5.14: Comparison of text alignment justified versus flush left

5.2.2 Actual Content

The content should not just be formatted properly, but should also be meaningful and understandable. Therefore the use of abbreviations should be limited and the full form should be used instead. Not all abbreviations are clear to everybody, so, if using them is necessary, an explanation should be provided as well.

Moreover, the use of local sayings and insider terms should be avoided. For example, outside of Vienna a *"Grätzeltour"* might not be obvious to everyone and *"Wien vom Sattel aus entdecken"* could be misinterpreted as well, e.g, it could mean by bike, by horse buggy or something completely different. Informational content should leave as little room as possible for interpretation and state the facts.

A change of language within the content should be declared accordingly. Multi language sites should give the option to mark a passage within a text as different language content. If a paragraph, or maybe the whole content within a page, has a different language, this part of the site needs to be declared with that language. Listening 5.14 shows how a paragraph is marked accordingly.

```
<div lang="de">
   Deutscher Inhalt innerhalb eines englischen Textes.
</div>
```

Listing 5.14: Language Tag for changing language

The reason is similar to the one mentioned for the language tag of the HTML declaration. The screen reader software checks the language tags and sets the language accordingly. This leads to terrible understandings if a German screen reader reads, for example, French text. The pronunciation is off and it is hard to follow and sometimes even impossible to understand.

When using quotes they should be marked with the `<q>` tag for inline quotations or the `<blockquote>` tag for more elaborate ones. By doing so, the screen reader reads it properly as a quote.

Referring to items on a page should not be based upon its color or position alone. A common, but not sufficient, way to express this, for example, *"Find further information on the product in the red box on the right."*. With responsive and adaptive design, it cannot be guaranteed where the "box" containing this item is and a color blind person will not be able to identify this box, especially if it is not the only box on the page. A better solution is to use proper references such as a title or a caption text.

5.2.3 Links and Buttons in the Content

Adding links to the content requires a proper marking and the link target should be clear. Therefore, a differentiation between external and internal links has to be considered. As the implementation allows a distinction between links to internal sites and a links to external content, it is the *Content Managers* duty to classify the links correctly and to make the differentiation visible according to the specified design. The link-text describes the target of the link and editors are admonished to clearly indicate what a link links to. Links like *more, read more, click here, ...* should be avoided and replaced by meaningful terms, representing the content of the target page.

If the editor has the option to add editorial button, then the same rules apply. A clear description of the buttons function has to be provided. Usually the length of the button text should not be too long, as it negatively affects Usability, but still should describe clearly what the button is for. For example, the button to approve the payment in a Web shop, should rather say *Confirm Payment* than just *Confirm*.

5.2.4 Headings

Using a consistent hierarchy of headings does not only help the Accessibility of a Website, but also improves the SEO [Goo10] and thus the ranking of the Website in search engines. Research shows that people consume Websites differently from traditional content like Magazines or Newspapers. They do not fully read the content of Websites and generally scan Websites for specific content and only read the content they are really interested in [Nie14]. Headings and well structured content is the key in skimming trough a Website [Kru14]. The same is true for users with assistive technologies. Screen reader software assists jumping from one heading to the next and enables the user to decide whether she/he reads the whole text or not. Having a clear hierarchy, including the usage of the heading tags `<h1>` to `<h6>`, has a lot of advantages. Hence, instead of structuring the text with fake headings, bold text and various sizes, it makes sense to use the correct markup.

5.2.5 Lists

The use of correct markup for ordered and unordered lists has similar reasons as described above. In general, it can be argued with users scanning a text and lists catch a higher attention and are easier to grasp [Nie14]. For screen reader software it helps with navigating a Website as well. The list elements are identified by the software and handled accordingly. In contrast, fake lists with, for example, just the "–" signs upfront, confuse users with assistive technologies.

5.2.6 Color

Taking care of the correct contrast ratio between text and background is of high importance as described in Section 5.1.2. In general this is less of a task for the *Content Manager* since the CMS should limit the available colors based on the definitions in the phases *Concept, Specifications & Requirements* and *Design* (see Section 4.3.2 and Section 4.3.3 respectively. The CI/CD specifications are in place to limit the color palette. Still it is beneficial if the editor is aware of the basics on Accessibility and colors.

The editor has to make sure, that color is not the only way to convey information. For example, when using graphs on a Website, a legend is a common way to describe the elements, but for color blind users the information could get lost. Phrases like *"Click on the red button"* or *"Find information in the blue box"* are not helpful for color blind users.

5.2.7 Multimedia

Adding multimedia content to a page involves Accessibility considerations as well. All images that are uploaded to the CMS and used within the Website, must have an alternative text specified. The alternative text is read to a user by the screen reader software and represents the image in textual form. It should give the user a notion on what is seen on the picture. It should not be too long, but still outline what is displayed on the image. If, for any reason, a longer description is needed, it is better to refer to a separate long-description and indicate how it can be accessed, than having the full alternative description in the *alt*-attribute. For decorative images, or images which do not convey information, the alternative text can be left empty.

Integrating an accessible video means that at least captions, which should be accessible for any software, have to be provided. For a higher level of Accessibility, a transcript of the video and an audio description have to be included. The same applies to audio content on a Website.

Any other added multimedia content, which is only understandable if it is perceived visually, needs a proper description, e.g., charts, info-graphics or maps.

Media content which can be downloaded from a Website needs to be marked appropriately. Best practice is to display the name of the element that can be downloaded, the format of the element and the file size. Listing 5.15 shows an example for an accessible download link.

```
<a href="thesis.pdf">Download the diploma thesis (PDF, 5 MB)</a>
```

Listing 5.15: Accessible download link

5.3 WCAG 2.0 Compliance

This thesis cannot address the full stack of WCAG guidelines as they are very comprehensive. A resume on the complied criteria is given in this section. The guidelines, that are covered in the *Accessibility Toolbox* are marked with a checkmark (✓) in the following tables. Left out are mostly the AAA level criteria, which are rarely - only for customers with a very high Accessibility demand - applicable.

5.3.1 Perceivable

Information and UI components must be presented to users in ways they can perceive it.

Guideline	Level	Description	
1.1.1 Non-text Content	A	Text alternatives for non-text content	✓
1.2.1 Audio-only and Video-only (Prerecorded)	A	Alternative to video-only and audio-only content	✓
1.2.2 Captions (Prerecorded)	A	Captions for video with audio	✓
1.2.3 Audio Description or Media Alternative (Prerecorded)	A	Alternative for media or audio	✓
1.2.4 Captions (Live)	AA	Live videos should have captions	✓
1.2.5 Audio Description (Prerecorded)	AA	Access to audio description for video content	✓
1.2.6 Sign Language (Prerecorded)	AAA	Sign language translations for videos	
1.2.7 Extended Audio Description (Prerecorded)	AAA	Extended audio description for videos	
1.2.8 Media Alternative (Prerecorded)	AAA	Text alternative to videos	
1.2.9 Audio-only (Live)	AAA	Alternatives for live audio	
1.3.1 Info and Relationships	A	Logical structure	✓
1.3.2 Meaningful Sequence	A	Content in a meaningful order	✓
1.3.3 Sensory Characteristics	A	Use more than one sense for instructions	✓
1.4.1 Use of Color	A	Not solely rely on color	✓
1.4.2 Audio Control	A	No automatic audio playing	✓
1.4.3 Contrast (Minimum)	AA	Contrast ratio is at least 4.5:1	✓
1.4.4 Resize text	AA	Text can be resized to 200%	✓
1.4.5 Images of Text	AA	Images of text should not be used	✓
1.4.6 Contrast (Enhanced)	AAA	Contrast ratio is at least 7:1	
1.4.7 Low or No Background Audio	AAA		
1.4.8 Visual Presentation	AAA	Offer a range of presentation options	
1.4.9 Images of Text (No Exception)	AAA		

Table 5.3: Overview of *WCAG Principle 1: Perceivable* including compliance mark

5.3.2 Operable

All users have to be able to interact and operate with the UI components and elements, no matter how they access it.

Guideline	Level	Description	
2.1.1 Keyboard	A	Accessible by keyboard	✓
2.1.2 No Keyboard Trap	A		✓
2.1.3 Keyboard (No Exception	AAA	No traps for keyboard users	✓
2.2.1 Timing Adjustable	A	Time limits have user controls	✓
2.2.2 Pause, Stop, Hide	A	User controls for moving content	✓
2.2.3 No Timing	AAA	No time limits	
2.2.4 Interruptions	AAA		
2.2.5 Re-authenticating	AAA	Save user data when re-authenticating	
2.3.1 Three Flashes or Below Threshold	A	No content flashes more than three times per second	✓
2.3.2 Three Flashes	AAA	No content flashes more than three times per second	
2.4.1 Bypass Blocks	A	Skiplinks	✓
2.4.2 Page Titled	A	Page title for every page	✓
2.4.3 Focus Order	A	Logical order	✓
2.4.4 Link Purpose (in Context)	A	The purpose of a link is clear	✓
2.4.5 Multiple Ways	AA	Several ways of navigating the site should be provided	✓
2.4.6 Headings and Labels	AA	Clear headings and labels	✓
2.4.7 Focus Visible	AA	Ensure keyboard focus is visible	✓
2.4.8 Location	AAA	Let users know where they are	✓
2.4.9 Link Purpose (Link Only)	AAA	Purpose of link is clear from its text	✓
2.4.10Section Headings	AAA	Structure content with headings	✓

Table 5.4: Overview of *WCAG Principle 2: Operable* including compliance mark

5.3.3 Understandable

Information and operations of a Website must be understandable.

Guideline	Level	Description	
3.1.1 Language of Page	A	Page has a language assigned	✓
3.1.2 Language of Parts	AA	Changes of language within the page have to be marked	✓
3.1.3 Unusual Words	AAA	Explain any uncommon words	
3.1.4 Abbreviations	AAA	Explain any abbreviation	
3.1.5 Reading Level	AAA		
3.1.6 Pronunciation	AAA		
3.2.1 On Focus	A	Elements do not change on focus	✓
3.2.2 On Input	A	Elements do not change when they receive content	✓
3.2.3 Consistent Navigation	AA	Navigation menu has to be consistent on entire Website	✓
3.2.4 Consistent Identification	AA	Consistency of elements	✓
3.2.5 Change on Request	AAA	No changes of elements unless with user interaction	
3.3.1 Error Identification	A	Identify and describe occurred errors	✓
3.3.2 Labels or Instructions	A	Element labeling and instructions if user input is required	✓
3.3.3 Error Suggestion	AA	Suggestions in case of user errors	✓
3.3.4 Error Prevention (Legal, Financial, Data)	AA	Risk of input errors for sensitive data should be reduced	✓
3.3.5 Help	AAA	Provide help and instructions	✓
3.3.6 Error Prevention (All)	AAA	Reduce risk of input errors	✓

Table 5.5: Overview of *WCAG Principle 3: Understandable* including compliance mark

5.3.4 Robust

Content must be robust enough for reliable interpretation by a wide variety of user agents, including assistive technologies. The code has to be as clean as possible and meet standards so that browsers are able to render the page.

Guideline	Level	Description	
4.1.1 Parsing	A	Validate code via tools or manual code-reviews and minimize errors	✓
4.1.2 Name, Role, Value	A	Use of standard HTML markup for adding new elements. All attributes must be set accordingly. If such a new element changes a notification is made available to user-agents.	✓

Table 5.6: Overview of *WCAG Principle 4: Robust* including compliance mark

5.4 Continuous Quality Assurance & Monitoring

Quality Assurance & Monitoring is a task that is continuously performed throughout the process life-cycle. In Section 4.3.9 a detailed description is given and its influence on the *Testing* phase is discussed.

This section focuses on the aspect of quality assurance once a Website is launched. Typically further implementations and improvements have to be implemented once a Website is in production. Changing environments and technologies make it necessary to monitor the product and adapt and improve it if desired. To keep a good Accessibility level it is important to maintain the Website.

However, each new feature, change-request or bugfix makes it necessary to reevaluate the Accessibility. This means, new features have to be planned and specified already with Accessibility in mind and also require a re-testing of parts of the Website. All newly implemented requirements have to follow the process described in Chapter 4 and are subject to the same quality standards.

5.5 Tools Supporting Accessibility Testing

Trying to evaluate the Accessibility of a Website raises particular challenges as it puts oneself into an unfamiliar situations. Experiences from a disabled person can not easily be reproduced or simulated and an expert has to somehow try to emulate the various situations as good as possible. Accessibility guidelines and tools help to bridge these experience gaps. However, it should be kept in mind, that theses applications are only a supplement, and not a replacement for including (disabled) people to gain experience.

In addition, the tools have various limitations and a solely automated test situation is hardly possible or rather not sufficient. For example, a tool only check of alternative texts in attributes would always be successful as long as any text in the attribute is given. The actual content and how informative the text is, cannot be analyzed automatically. However, tools are a great assistance and make manual testing much easier by producing a list of possible errors and/or areas that have to be investigated.

This section presents a selection of tools, but has no intend to be an exhaustive list. For a more extensive list please refer to the *Web Accessibility Evaluation Tools List*[13] of the W3C. Common practice, at least along the interview partners, is to use more than just one specific tool. Generally a mixture of the (subjectively) most attractive tools are used.

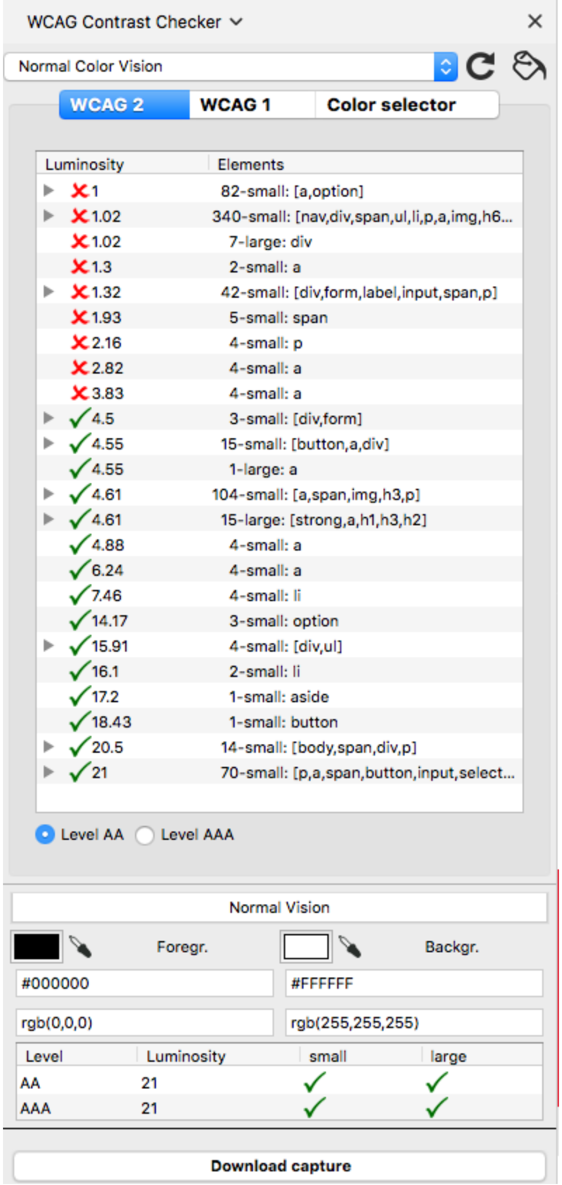
Some of these tools are implemented as *add-ons* or *extensions* for specific browsers. Not all of them are available for all browsers, but using Chrome, Firefox or Edge is usually a good choice. A proper evaluation should always consider more than one browser anyway. That does not mean, that the entire evaluation needs to be done in every browser, but at least a short check, confirming existing failures and checks on a random basis should be done in more than one browser. Moreover, it has to be mentioned, that this thesis is neutral and does not recommend or privilege any browser or extension.

5.5.1 WCAG Color Contrast Checker

A quick examination of the color contrasts used on a page can be done with the *WCAG Contrast Checker*[14] seen in Figure 5.15. An overview on which elements fail the color contrast check is provided and details, e.g., the exact color code, can be seen at the bottom of the widget by selecting the corresponding item. Additionally, the selection highlights the according elements on the page and allow an easy identification of the elements violating the guideline.

[13] https://www.w3.org/WAI/ER/tools/, accessed Aug 2017

[14] https://addons.mozilla.org/en-US/firefox/addon/wcag-contrast-checker/, accessed Aug 2017



Luminosity	Elements
▶ ✗1	82-small: [a,option]
▶ ✗1.02	340-small: [nav,div,span,ul,li,p,a,img,h6...
✗1.02	7-large: div
✗1.3	2-small: a
▶ ✗1.32	42-small: [div,form,label,input,span,p]
✗1.93	5-small: span
✗2.16	4-small: p
✗2.82	4-small: a
✗3.83	4-small: a
▶ ✓4.5	3-small: [div,form]
▶ ✓4.55	15-small: [button,a,div]
✓4.55	1-large: a
▶ ✓4.61	104-small: [a,span,img,h3,p]
▶ ✓4.61	15-large: [strong,a,h1,h3,h2]
✓4.88	4-small: a
✓6.24	4-small: a
✓7.46	4-small: li
✓14.17	3-small: option
▶ ✓15.91	4-small: [div,ul]
✓16.1	2-small: li
✓17.2	1-small: aside
✓18.43	1-small: button
▶ ✓20.5	14-small: [body,span,div,p]
▶ ✓21	70-small: [p,a,span,button,input,select...

● Level AA ○ Level AAA

Normal Vision

Foregr.	Backgr.
#000000	#FFFFFF
rgb(0,0,0)	rgb(255,255,255)

Level	Luminosity	small	large
AA	21	✓	✓
AAA	21	✓	✓

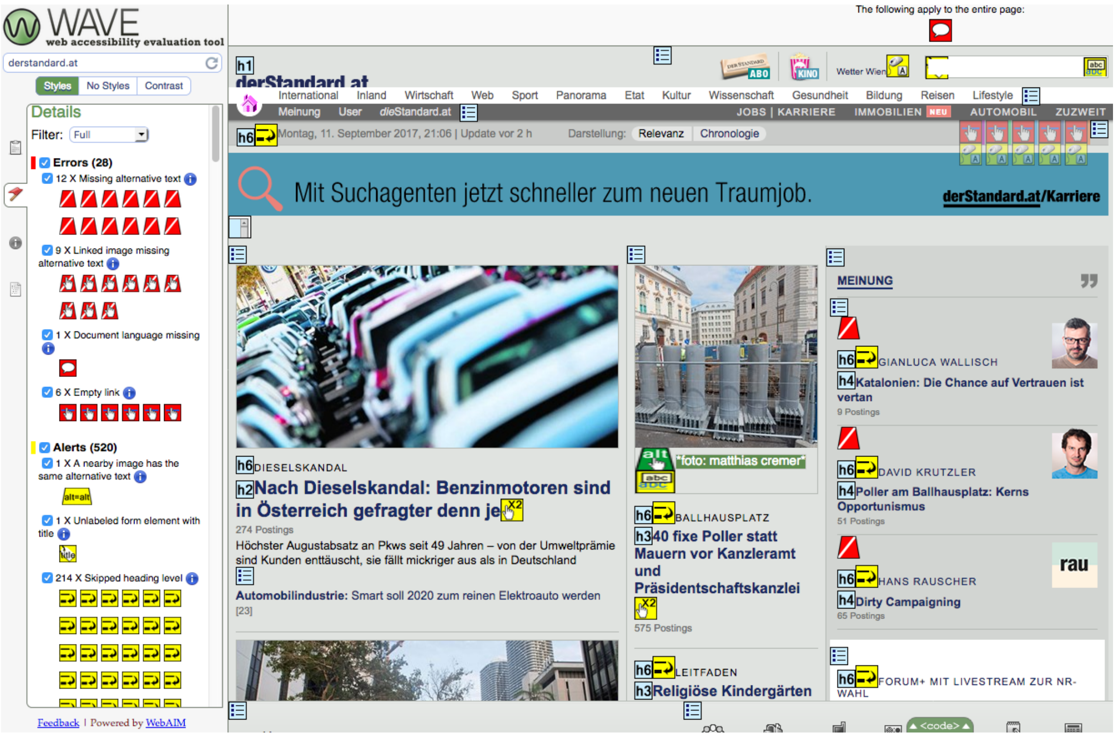
Figure 5.15: The WCAG Contrast Checker

5.5.2 Web Accessibility Evaluation Tool - WAVE

WAVE[15] is a very popular tool with a large community. *WAVE* is one of the first tools that the experts of the interviews use when starting an evaluation. It allows to check a page by providing the URL to an online tool (Website). Hence, it is not a add-on to a browser, but rather an online service.

[15] http://wave.webaim.org/, accessed Sep 2017

It provides a good overview on what WCAG criteria are violated on a page. Selecting the error icon in the *WAVE* explorer highlights the corresponding elements on the page. Choosing the tab *No Styles* displays the site without CSS support and hence offers a quick check of the logical structure. Figure 5.16 shows the *WAVE* tool report for *derstandard.at*.



WAVE
web accessibility evaluation tool

derstandard.at

Styles | No Styles | Contrast

Details

Filter: Full

☑ Errors (28)

☑ 12 X Missing alternative text

☑ 9 X Linked image missing alternative text

☑ 1 X Document language missing

☑ 6 X Empty link

☑ Alerts (520)

☑ 1 X A nearby image has the same alternative text

☑ 1 X Unlabeled form element with title

☑ 214 X Skipped heading level

alt=alt
title
Feedback | Powered by WebAIM

The following apply to the entire page:

h1 derStandard.at

h6 Montag, 11. September 2017, 21:06 | Update vor 2 h

Darstellung: Relevanz | Chronologie

Mit Suchagenten jetzt schneller zum neuen Traumjob. derStandard.at/Karriere

h6 DIESELSKANDAL

h2 Nach Dieselskandal: Benzinmotoren sind in Österreich gefragter denn je X2

274 Postings

Höchster Augustabsatz an Pkws seit 49 Jahren – von der Umweltprämie sind Kunden enttäuscht, sie fällt mickriger aus als in Deutschland

Automobilindustrie: Smart soll 2020 zum reinen Elektroauto werden [23]

alt *foto: matthias cremer*

h6 BALLHAUSPLATZ

h3 40 fixe Poller statt Mauern vor Kanzleramt und Präsidentschaftskanzlei X2

575 Postings

h6 LEITFADEN

h3 Religiöse Kindergärten

MEINUNG

h6 GIANLUCA WALLISCH

h4 Katalonien: Die Chance auf Vertrauen ist vertan

9 Postings

h6 DAVID KRUTZLER

h4 Poller am Ballhausplatz: Kerns Opportunismus

51 Postings

h6 HANS RAUSCHER

h4 Dirty Campaigning

65 Postings

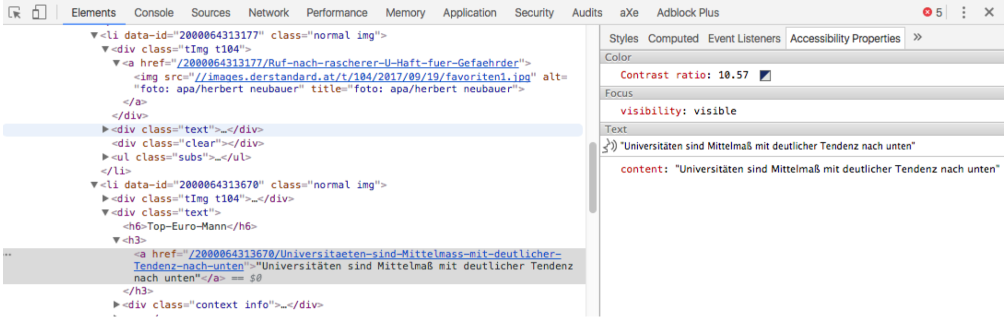
h6 FORUM+ MIT LIVESTREAM ZUR NR-WAHL

Figure 5.16: Exemplary WAVE report of derstandard.at, accessed Sep 2017

5.5.3 Web Developer Tools

In general, the *Web Developer Tools* are a great tool-suite for developers, testers and evaluators. They allow to debug code and are a helpful instrument for inspecting single elements and checking the structure of the source code. The *Web Developer Tools* can be accessed via right mouse click choosing *inspect element* or by pressing the *F12*-key on Windows and Linux or *Command+Option+I* on Mac in the Firefox and Chrome browser.

They allow to make quick and easy (temporary) changes on the Website to show a customer how a specific setting in the code affects the Website. Often this helps to reduce time and effort for wireframes and allows the customer to get a clear image his vision. The *Web Developer Tools* offer much more practical features, which are not part of the topic of the thesis. The *Chrome DevTools* already provide a native Accessibility inspection. This information is limited, but a good starting point. *Chrome DevTools* are shown in Figure 5.17.



```
▼<li data-id="2000064313177" class="normal img">
  ▼<div class="tImg t104">
    ▼<a href="/2000064313177/Ruf-nach-rascherer-U-Haft-fuer-Gefaehrder">
        <img src="//images.derstandard.at/t/104/2017/09/19/favoriten1.jpg" alt=
        "foto: apa/herbert neubauer" title="foto: apa/herbert neubauer">
      </a>
    </div>
  ▶<div class="text">…</div>
    <div class="clear"></div>
  ▶<ul class="subs">…</ul>
  </li>
▼<li data-id="2000064313670" class="normal img">
  ▶<div class="tImg t104">…</div>
  ▼<div class="text">
      <h6>Top-Euro-Mann</h6>
    ▼<h3>
        <a href="/2000064313670/Universitaeten-sind-Mittelmass-mit-deutlicher-
        Tendenz-nach-unten">"Universitäten sind Mittelmaß mit deutlicher Tendenz
        nach unten"</a> == $0
      </h3>
    ▶<div class="context info">…</div>
```

Color

```
Contrast ratio: 10.57
```

Focus

```
visibility: visible
```

Text

"Universitäten sind Mittelmaß mit deutlicher Tendenz nach unten"

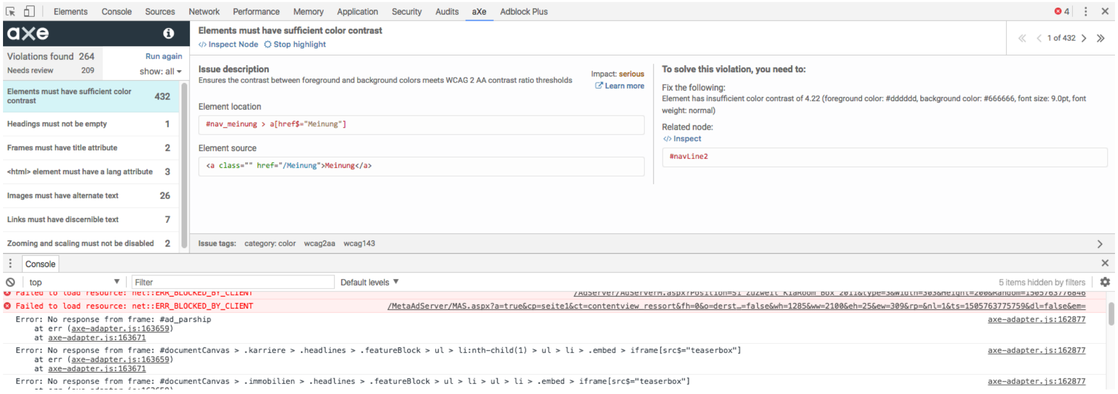
```
content: "Universitäten sind Mittelmaß mit deutlicher Tendenz nach unten"
```

Figure 5.17: Native Accessibility Inspection in Chrome Developer Tools

5.5.4 aXe - The Accessibility Engine

The *aXe- The Accessibility Engine*[16] is an automated tool to evaluate Accessibility defects on a Website. It is available as a plugin for the *Web Developer Tools* for Firefox and Chrome. Once installed it can be located in the *Web Developer Tools* or *Site Inspector* menu.

Figure 5.18 shows the tool in action. It is implemented as a separate tab in the *Web Developer Tools* and allows to check specific Accessibility rules based on the WCAG.



Violations found	264
Needs review	209
Elements must have sufficient color contrast	432
Headings must not be empty	1
Frames must have title attribute	2
<html> element must have a lang attribute	3
Images must have alternate text	26
Links must have discernible text	7
Zooming and scaling must not be disabled	2

Elements must have sufficient color contrast

Issue description
Ensures the contrast between foreground and background colors meets WCAG 2 AA contrast ratio thresholds

Impact: serious

Element location

```
#nav_meinung > a[href$="Meinung"]
```

Element source

```
<a class="" href="/Meinung">Meinung</a>
```

To solve this violation, you need to:

Fix the following:
Element has insufficient color contrast of 4.22 (foreground color: #dddddd, background color: #666666, font size: 9.0pt, font weight: normal)

Related node:

```
#navLine2
```

Issue tags: category: color wcag2aa wcag143

```
Failed to load resource: net::ERR_BLOCKED_BY_CLIENT        /AdServer/AdServerM.aspx?Position=SI_ZuZweit_KiaRoom_Box_2011&type=3&width=303&height=200&Random=1505763776846
Failed to load resource: net::ERR_BLOCKED_BY_CLIENT        /MetaAdServer/MAS.aspx?a=true&cp=seite1&ct=contentview_ressort&fh=0&o=derst…=false&wh=1285&ww=2100&eh=25&ew=309&rp=&nl=1&ts=1505763775759&dl=false&em=
  Error: No response from frame: #ad_parship                                                                         axe-adapter.js:162877
      at err (axe-adapter.js:163659)
      at axe-adapter.js:163671
  Error: No response from frame: #documentCanvas > .karriere > .headlines > .featureBlock > ul > li:nth-child(1) > ul > li > .embed > iframe[src$="teaserbox"]   axe-adapter.js:162877
      at err (axe-adapter.js:163659)
      at axe-adapter.js:163671
  Error: No response from frame: #documentCanvas > .immobilien > .headlines > .featureBlock > ul > li > ul > li > .embed > iframe[src$="teaserbox"]   axe-adapter.js:162877
```

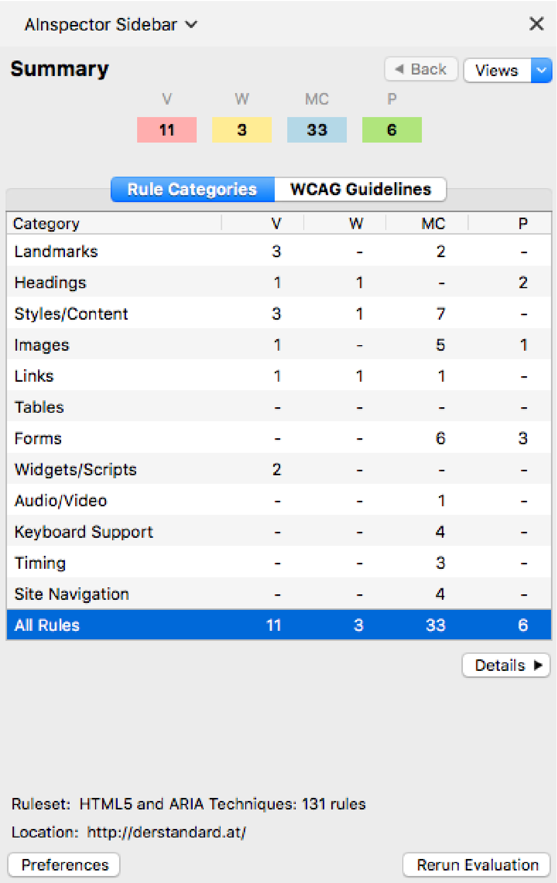
Figure 5.18: Screenshot of the aXe Developer Tools Report from derstandard.at, accessed Sep 2017

On the left side the report summarizes the violations and divides them into groups. By selecting single group items on the right side - *the main part* - a description of the violation is presented. Links to inspect the source code or to highlight the element are given. In addition, the tool suggests possible solutions to the identified problems.

[16] https://www.deque.com/products/axe/, accessed Sep 2017

5.5.5 AInspector Sidebar

Another tool for automatic evaluation according to the WCAG guidelines is the AInspector Sidebar[17]. Once the tool is run it loads on the side in the current tab and shows a summary report categorized by Rule Categories or WCAG guidelines, see as example Figure 5.19. Digging deeper into the single violated categories more information gets available. On the lowest level a description of the violation is presented as well as an option to highlight the affected elements. The details of the description includes a definition, an action on what to do to fix the violation as well as the purpose explaining the reasons for that rule. In addition, techniques for a proper implementation and compliance information are provided, including links to the violated WCAG success criteria.



V	W	MC	P
11	3	33	6

Category	V	W	MC	P
Landmarks	3	-	2	-
Headings	1	1	-	2
Styles/Content	3	1	7	-
Images	1	-	5	1
Links	1	1	1	-
Tables	-	-	-	-
Forms	-	-	6	3
Widgets/Scripts	2	-	-	-
Audio/Video	-	-	1	-
Keyboard Support	-	-	4	-
Timing	-	-	3	-
Site Navigation	-	-	4	-
All Rules	11	3	33	6

Ruleset: HTML5 and ARIA Techniques: 131 rules

Location: http://derstandard.at/

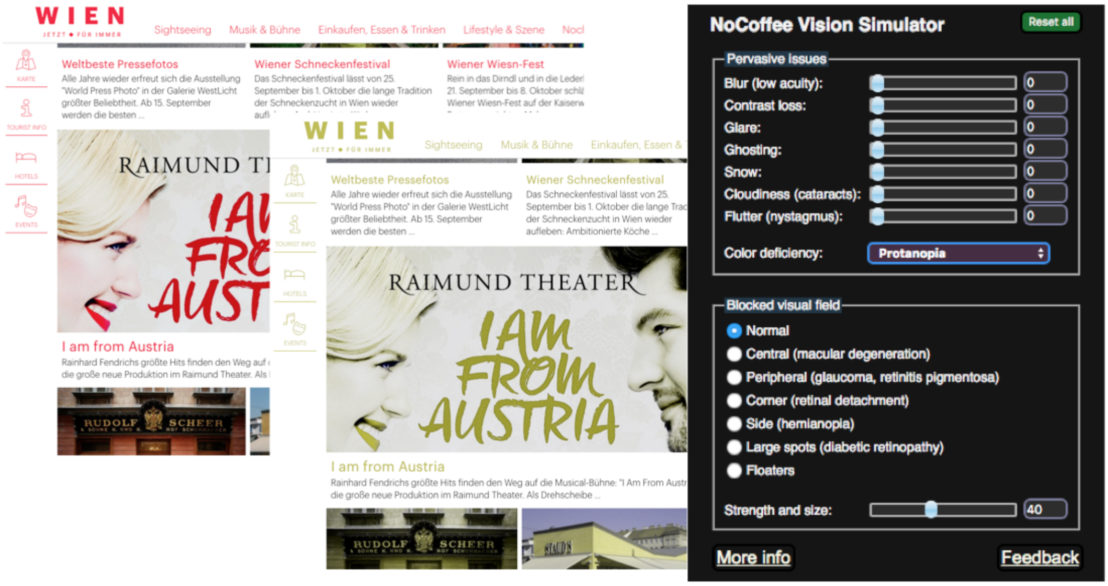
Figure 5.19: AInspector Sidebar

[17] https://addons.mozilla.org/en-US/firefox/addon/ainspector-sidebar/, accessed Sep 2017

5.5.6 NoCoffee vision simulator

The *NoCoffee vision simulator*[18] tries to simulate the effects of various vision impairments on Websites. It provides the means to experience and understand how users with visual impairments see and interact with a Website.

Once installed it can be activated on any Website and various vision impairments can be simulated. Figure 5.20 shows how the tool creates a Website of the Vienna Tourist Board `www.vienna.info` with simulated red-green vision impairment. The original Website is included as well for comparison.



NoCoffee Vision Simulator

Reset all

Pervasive issues

Blur (low acuity): 0

Contrast loss: 0

Glare: 0

Ghosting: 0

Snow: 0

Cloudiness (cataracts): 0

Flutter (nystagmus): 0

Color deficiency: Protanopia

Blocked visual field

- ● Normal
- ○ Central (macular degeneration)
- ○ Peripheral (glaucoma, retinitis pigmentosa)
- ○ Corner (retinal detachment)
- ○ Side (hemianopia)
- ○ Large spots (diabetic retinopathy)
- ○ Floaters

Strength and size: 40

More info

Feedback

Figure 5.20: Website of the Vienna Tourist Board with a simulated Red-Green Color Blindness, accessed Sep 2017

5.5.7 Web Developer Toolbar

The *Web Developer Toolbar*[19] is browser plugin that adds a wide variety of web developer tools. It allows to check a number of Accessibility related features and provides the possibility to generate reports for Websites. Available items are *Disable (CSS, Javascript...), Cookies, CSS, Forms, Images, Information, Miscellaneous, Outline, Resize, Tools* and *Options* and are shown in Figure 5.21.

One option is to turn off JavaScript and to thus to identify if a page is still functional. By disabling the CSS the logical structure of the page can easily be observed and the reading flow without styles can be verified. Deactivating all images and instead showing only the

[18] `https://chrome.google.com/webstore/detail/nocoffee/jjeeggmbnhckmgdhmgdckeigabjfbddl`, accessed Sep 2017

[19] `https://addons.mozilla.org/en-US/firefox/addon/web-developer/`, Aug 2017

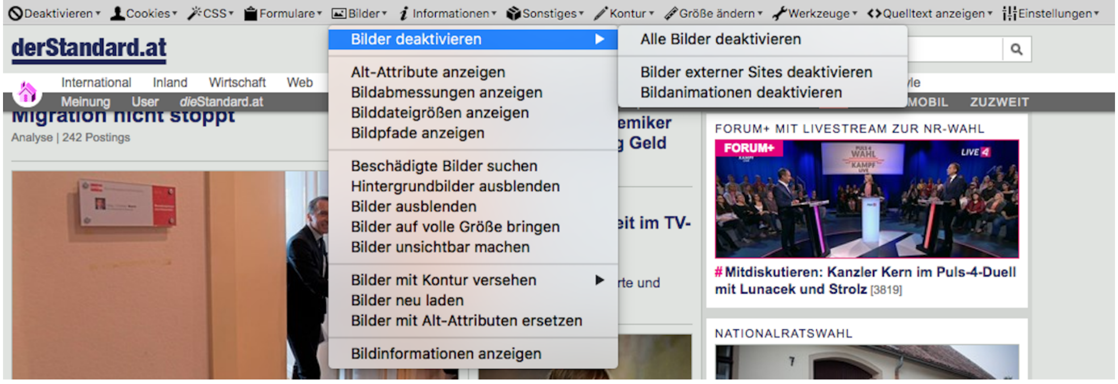


Figure 5.21: Web Developer Toolbar

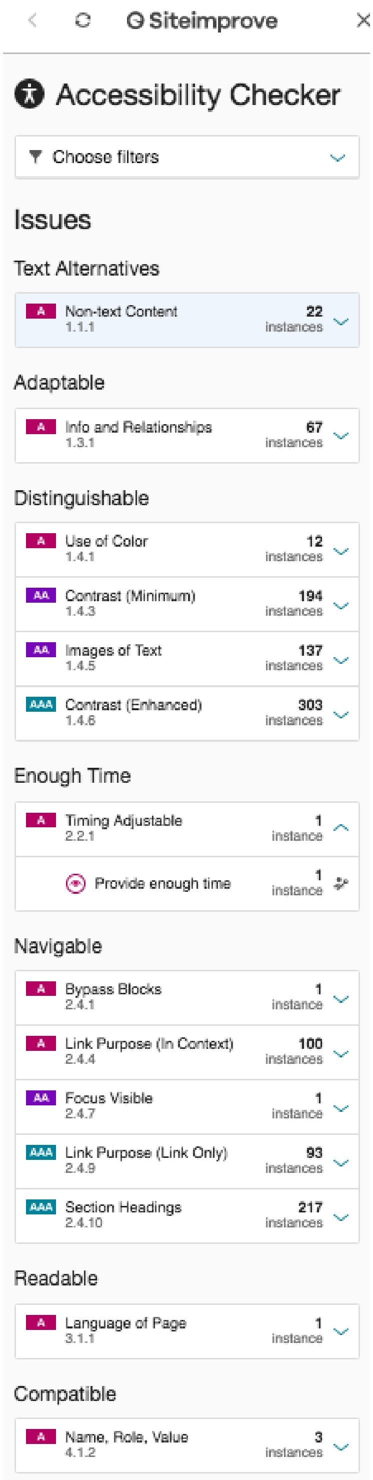
alternative text is another helpful option. The Information option provides identifications like *tabindex*, *ARIA roles*, *ids*, the *document outline*,... for various elements on a page. In additon, the *tools tab* offers validation options for HTML and CSS source-cde. This is just a summary on some of the more important options as the extension offers a lot more to be explored.

5.5.8 Siteimprove Accessibility Checker

Similar to the previously described tools the *Siteimprove Accessibility Checker*[20] is also embedded in a browser. It is only available for Chrome.

Figure 5.22 shows the *Siteimprove Accessibility Checker*. The tool identifies issues and groups them according to specific WCAG rules. All instances (elements) that related to a rule are listed. Each instance can be further investigated and, if a violation occurs, be highlighted. Additionally, the tool also offers information on how the elements that violated the success criteria can be improved.

[20] https://chrome.google.com/webstore/detail/siteimprove-accessibility/efcfolpjihicnikpmhnmphjhhpiclljc?hl, accessed Sep 2017



Siteimprove

Accessibility Checker

Choose filters

Issues

Text Alternatives

A Non-text Content 1.1.1 — 22 instances

Adaptable

A Info and Relationships 1.3.1 — 67 instances

Distinguishable

A Use of Color 1.4.1 — 12 instances

AA Contrast (Minimum) 1.4.3 — 194 instances

AA Images of Text 1.4.5 — 137 instances

AAA Contrast (Enhanced) 1.4.6 — 303 instances

Enough Time

A Timing Adjustable 2.2.1 — 1 instance

Provide enough time — 1 instance

Navigable

A Bypass Blocks 2.4.1 — 1 instance

A Link Purpose (In Context) 2.4.4 — 100 instances

AA Focus Visible 2.4.7 — 1 instance

AAA Link Purpose (Link Only) 2.4.9 — 93 instances

AAA Section Headings 2.4.10 — 217 instances

Readable

A Language of Page 3.1.1 — 1 instance

Compatible

A Name, Role, Value 4.1.2 — 3 instances

Figure 5.22: Siteimprove Accessibility Checker

CHAPTER 6

Conclusion and Outlook

The Internet and its user base are rapidly growing. A broader range of people have access to the Internet, the variety of devices connecting to it is rising and, of course, also the offered services increase constantly. Thus, as a result the requirements of a Website gain complexity due to the fact, that the Web has to be made usable under multiple circumstances. The expanded and more diverse user base makes Usability and Accessibility considerations indispensable. Another factor is that with higher life expectancy more and more old people get attracted to the Internet. In addition also less educated and moreover also more disabled people get the chance or have the need to use the Internet.

After an introduction and outlining the motivation for the thesis, a view on the theoretical basics is given. These fundamentals include a definition of Usability and Accessibility as well as an overview on the legal situation on Web Accessibility in Austria. Moreover, a detailed overview on Accessibility is provided. Included in the fundamentals is also a description of the target groups of accessible Websites and possible disabilities which affect people using the Internet are analyzed in detail.

The goal of the thesis was to provide a structured process on how to integrate Usability and Accessibility in a standardized way into the project cycle. Based on existing agile software engineering methodologies, the proposed process describes the considerations that have to be taken into account in each project phase. Furthermore, a framework supporting the project team while working with the process is proposed - the *Framework for Integrating Accessibility (FIA)*. For each phase of the process predefined tasks can be generated according to a specified template structure. Related tasks are automatically generated according to the defined process workflow. A visualization of the tasks and a representation of the workflow, is provided via a for the process adjusted Kanban Board. This custom Kanban Board labeled *FIA Board* contains pre-defined tasks that facilitate working with the proposed process.

Since the team is an essential part in every software project, a detailed overview on all involved stakeholders is given. Alongside this overview also a specification of the demands that each stakeholder has to fulfill is provided.

Performing interviews with Austrian Accessibility experts showed on the one hand the need for a structured approach, but also revealed how little importance is attached to the topic by the general public. The main reasons for customers to demand an accessible Website are the legal regulations. And even these legal regulations are often ignored. Another implication given by the interviews is, that, initially the workload of a project with an emphasis on Accessibility appears higher and so do the costs. However, once the team is well-trained on the basics about what to keep in mind and how to implement it properly, the additional work effort is minor. This is also in strong contrast to considering Accessibility as an afterthought, meaning a proper accessible integration into an existing Website, as the costs in such a case are way higher. None of the experts was willing to name a number, but they all agreed on definite extra complexity and effort if Accessibility is not considered from the beginning.

Most importantly though, the interviews showed that there is no well-structured process available that they proceed on. All employ their own strategies and approaches gained by experience. A clear path on how to proceed to tackle the problem of widely incomplete Accessibility implementations is missing. In addition, the interviewees also agreed and emphasized the need for clear and comprehensive specifications. The interviews clearly showed, that the need for according requirement specifications for the Accessibility demands would relieve the overall work effort. Partly the interview partners already try to define such specifications, but are not able to utilize them in all their projects. In fact, most of them only create formal requirement specifications in a minority of their projects. The conclusion that has been drawn was, that a standardized process including a supporting framework with predefined repetitive tasks would be an advantage.

The input from the expert interviews helped to refine the proposed process. The interviews provided insights into the work of practitioners and how they address common Accessibility issues. In addition, the interviews also brought some tools that can be used as a support for various stakeholders to light. All these information was incorporated into this thesis and help to sharpen the proposed process and the supporting framework.

In essence, the proposed process specifies a way how a project team can build accessible Websites efficiently. The basic idea is taken from the common agile development cycle, which is enriched with continuous Usability and Accessibility considerations. That is, each phase has defined special requirements concerning Accessibility which have to be fulfilled. The proposed framework supports this process by putting an emphasis on these requirements and highlighting them for the development team. The requirements are preprocessed in template based tasks and are automatically created and presented to the responsible stakeholders in each phase. In addition, these tasks have dependencies respectively successor tasks that are automatically created as soon as a task is fulfilled. This leads to an automatic creation of a task based workflow.

A concrete characterization of these requirements and how they can be handled was given a dedicated section in this thesis. This so called *Accessibility Toolbox* provides practical description on rules based on the WCAG and how they should be addressed. For each phase of the proposed process a detailed overview was given. Moreover, some Accessibility evaluation advice and tool descriptions were provided as well. The implementation methodologies described in the *Accessibility Toolbox* can also be used as a means to specify the requirements for the process, e.g. in the *Concept, Specifications & Requirements* phase. Additionally, a detailed description of common issues and how to prevent them is given. Here it is shown how the *Accessibility Toolbox* can help to identify the inputs for the task template definitions of the proposed framework.

Combining all these parts lead to a well-structured and comprehensive requirement specification that enables the developers to get a better understanding on how to implement the Accessibility demands.

Some of the key questions this thesis tried to answer were:

1. How is Usability and Accessibility defined? What standards do exist?
2. Why is there a lack of awareness for Usability and Accessibility in software projects?
3. How can the methodologies of Usability, Accessibility and Software Engineering be integrated and merged?
4. Is it possible to measure the integrated approach and its advantages?
5. What tools and methods, concerning Usability and Accessibility, can be used to support software developers and project managers in the entire project life-cycle?

Question number one was extensively targeted and covered in the Chapter Theoretical Basics & Fundamentals. Different definitions of Usability and Accessibility were discussed and the legal situation in Austria regarding Accessibility was depicted. Additionally, an overview on one of the most important and commonly used standards, the Web Content Accessibility Guidelines (WCAG), was given.

The lack of Accessibility awareness is covered in Chapter Empirical Research: Interviews. The discussion of the awareness issues has been covered in different chapters as there are many factors for it. One of the common factors, also verified by the interviews, is that many users don't recognize issues if they are not affected themselves. Another reason is that features that are well programmed and target Accessibility are mostly not visible for the typical user.

The third question is also the main focal point of this thesis. Chapter 4 presents the proposed process and the integrated methodologies. The process alongside the involved stakeholders are described in detail. In addition, a framework is presented that supports the stakeholders during their work with the process. The framework is theoretically characterized and depicts how an implementation could be accomplished. The framework is based on a Kanban Board and visualizes tasks specifically created for Accessibility requirements. Special pre-defined Accessibility tasks are generated automatically, to assure, that Accessibility is covered in each of the process phases and that all stakeholders always have an overview of the project status. In essence, this thesis proposes to integrate agile methods based on Kanban to integrate special Accessibility focused tasks into a project life-cycle.

The investigation in the measurement of the approach was left open as there is no implementation for the framework provided. Thus, this provides possible future research. Is it even feasible to measure the savings of an integrated approach? During the development of the thesis, the considerations of the author were riven. On the one hand it would be great to present a way to measure the success of the process, but the interviews showed, that a measurement concerning Accessibility implementations is generally difficult. It involves many factors and a valid comparison of different projects is nearly impossible. The necessity of a setting where similar stakeholders, with similar previous knowledge are implementing two similar projects, is a challenge in itself. This was beyond the scope of this thesis.

The last posed question was addressed in Chapter 5, which presents implementation and tools regarding WCAG. The chapter includes a description of typical issues concerning Accessibility and a declaration on how to properly avoid the problems. This Toolbox can act as basis for the requirement specifications in the process and provides input for the FIA task templates.

Some limitations of this thesis should be mentioned as well. First of all the thesis and its research has a strong focus on the Austrian market. All expert interviews have been conducted with Austrian companies. Although the research is not solely based on Austria, it does have a strong focus on Austria and Austrian law regulations.

Another limitation of the approach is that the proposed framework is only of theoretical nature. Although the framework is backed up by expert interviews an actual implementation is desirable. Since this was out of scope for this thesis it is left open for future work.

Continuing investigations could put the proposed framework into practice. This further research should also consider affected users and include them into a study where the proposed approach is applied in a real life Web project. Based on the findings of the users the provided *Accessibility Toolbox* could be expanded to cover additional standards and tools. In addition, the process and the framework could be revised and a new version addressing the issues could be created.

Overall, this thesis provides a blueprint for a project development process with an emphasis on Usability and Accessibility. It proposes a framework based on agile software development methods, specifically a custom Kanban Board with specific, automatically created tasks, to support all stakeholders during the project. Altogether, this proposed approach should lead to more efficient development of accessible Websites.

List of Figures

List of Tables

Acronyms

ADD Attention Deficit Disorder. 19

ADHD Attention Deficit Hyperactivity Disorder. 19

ARIA Accessible Rich Internet Applications. 83, 95, 97

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart. 80, 97

CC Closed Captions. 78, 93, 94

CD Corporate Design. 105

CI Corporate Identity. 50, 105

CMS Content Management System. 52, 101, 103, 105

CSS Cascading Style Sheets. 80, 85–90, 97, 98, 103, 111, 118

EU European Union. 22, 24

FIA Framework for Integrating Accessibility. 56, 57, 59–62, 70, 74, 119, 122

HTML Hypertext Markup Language. 48, 81–86, 88, 95–99, 104, 118

ICT Information and Communications Technology. 15, 20

ISO International Organization for Standardization. 4, 5, 8, 14

SEO Search Engine Optimization. 12, 35, 36, 89, 93, 105

UI User Interface. 2, 7, 13, 74

UNESCO The United Nations Educational, Scientific and Cultural Organization. 19

UX User Experience. 77, 85

W3C World Wide Web Consortium. 9–12, 14, 15, 22, 81, 101, 111

WAI Web Accessibility Initiative. 10–12, 20, 22, 23

WCAG Web Content Accessibility Guidelines. ix, xi, 4, 12–14, 22, 31, 35, 37, 57, 58, 70, 73, 74, 76, 87, 93, 100, 106, 111–113, 121, 122

WIP Work In Progress. 24, 25, 57

WWW World Wide Web. 10, 14

WYSIWYG What you see is what you get. 102, 103

Bibliography

[AB16] Jali Mohd Zalisham Abuaddous, Hayfa.Y. and Nurlida Basir. Web Accessibility Challenges. *International Journal of Advanced Computer Science and Applications (IJACSA), Volume 7*, 7(10):172–181, October 2016.

[AUS] STATISTIK AUSTRIA. Bundesanstalt Statistik Österreich. Website. `https://www.statistik.at/web_de/statistiken/index.html`; accessed 2017-06-27.

[AUS13] STATISTIK AUSTRIA. Schlüsselompetenzen von Erwachsenen - Vertiefende Analysen der PIAAC-Erhebung 2011/12. Website, 2013. `http://www.statistik.at/web_de/services/publikationen/5/index.html?includePage=detailedView&sectionName=Bildung%2C+Kultur&pubId=690`; accessed 2016-12-21.

[BMMW15] Manuel Brhel, Hendrik Meth, Alexander Maedche, and Karl Werder. Exploring Principles of User-centered Agile Software Development. *Information and Software Technology*, 61(C):163–181, May 2015.

[Bra08] Giorgio Brajnik. Beyond Conformance: The Role of Accessibility Evaluation Methods. In *Proceedings of the 2008 International Workshops on Web Information Systems Engineering*, WISE '08, pages 63–80, Berlin, Heidelberg, 2008. Springer-Verlag.

[Con99] World Wide Web Consortium. Web Content Accessibility Guidelines (WCAG) 1.0. Website, 1999. `https://www.w3.org/TR/WCAG10/`; accessed 2016-12-27.

[Con08] World Wide Web Consortium. Web Content Accessibility Guidelines (WCAG) 2.0. Website, 2008. `https://www.w3.org/TR/WCAG20/`; accessed 2016-12-27.

[Con16a] World Wide Web Consortium. Web Accessibility Initiative (WAI). Website, 2016. `https://www.w3.org/WAI/`; accessed 2016-12-27.

[Con16b] World Wide Web Consortium. World Wide Web Consortium (W3C). Website, 2016. `https://www.w3.org/`; accessed 2016-12-27.

[Con17] World Wide Web Consortium. WAI-ARIA Authoring Practices 1.1. Website, 2017. https://www.w3.org/TR/wai-aria-practices-1.1/; accessed 2017-08-27.

[CRC07] Alan Cooper, Robert Reimann, and Dave Cronin. *About Face 3: The Essentials of Interaction Design.* John Wiley & Sons, Inc., New York, NY, USA, 2007.

[DR99] Joseph S. Dumas and Janice C. Redish. *A Practical Guide to Usability Testing.* Intellect Books, Exeter, UK, UK, 1st edition, 1999.

[Dre95] Eric Drever. *Using Semi-Structured Interviews in Small-Scale Research. A Teacher's Guide.* ERIC, 1995.

[EN 14] Accessibility requirements suitable for public procurement of ICT products and services in Europe. Website, Feb 2014. http://www.etsi.org/deliver/etsi_en/301500_301599/301549/01.01.01_60/en_301549v010101p.pdf; accessed 2016-12-27.

[FB07] Mark G Friedman and Diane Nelson Bryen. Web accessibility design recommendations for people with cognitive disabilities. *Technology and Disability*, 19(4):205–212, 2007.

[Fis12] Holger Fischer. Integrating Usability Engineering in the Software Development Lifecycle Based on International Standards. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '12, pages 321–324, New York, NY, USA, 2012. ACM.

[FMS14] Brian Fitzgerald, Mariusz Musiał, and Klaas-Jan Stol. Evidence-based Decision Making in Lean Software Project Management. In *Companion Proceedings of the 36th International Conference on Software Engineering*, ICSE Companion 2014, pages 93–102, New York, NY, USA, 2014. ACM.

[fPwD16] WebAIM Center for Persons with Disabilities. Constructing a POUR Website. Website, 2016. http://webaim.org/articles/pour; accessed 2016-12-27.

[fPwD17a] WebAIM Center for Persons with Disabilities. Cognitive Disabilities. Website, 2017. http://webaim.org/articles/cognitive/; accessed 2017-01-29.

[fPwD17b] WebAIM Center for Persons with Disabilities. Visual Disabilities: Colorblindness. Website, 2017. http://webaim.org/articles/visual/colorblind; accessed 2017-01-05.

[fPwD17c] WebAIM Center for Persons with Disabilities. Web Accessibility in Mind. Website, 2017. http://webaim.org/; accessed 2017-04-23.

[fSIa] International Organization for Standardization (ISO). ISO 9241-11.2 Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. https://www.iso.org; accessed 2017-07-30.

[fSIb] International Organization for Standardization (ISO). ISO/IEC 40500:2012 Information technology – W3C Web Content Accessibility Guidelines (WCAG) 2.0. https://www.iso.org/standard/58625.html; accessed 2017-07-29.

[Glo13] B. Gloger. *SCRUM Produkte Zuverlässig Und Schnell Entwickeln.* Carl Hanser Verlag München, 2013.

[Goo10] Google. Google's Search Engine Optimization Starter Guide, September 2010. http://static.googleusercontent.com/media/www.google.com/en//webmasters/docs/search-engine-optimization-starter-guide.pdf.

[Has11] Behzad Joze Hashemian. Analyzing Web Accessibility in Finnish Higher Education. *SIGACCESS Access. Comput.*, 101:8–16, September 2011.

[Hir07] Kenji Hiranabe. Visualizing Agile Projects using Kanban Boards. Website, 2007. https://www.infoq.com/articles/agile-kanban-boards; accessed 2017-10-01.

[Hir08] Kenji Hiranabe. Kanban Applied to Software Development: from Agile to Lean. Website, 2008. https://www.infoq.com/articles/hiranabe-lean-agile-kanban; accessed 2017-10-01.

[HM16] Jonathan Harris and Ottavio Marzocch. The European Accessibility Act - IN DEPTH ANALYSIS. Website, 2016. http://www.europarl.europa.eu/RegData/etudes/IDAN/2016/571382/IPOL_IDA(2016)571382_EN.pdf; accessed 2016-12-27.

[Hou14] Dan X. Houston. Agility Beyond Software Development. In *Proceedings of the 2014 International Conference on Software and System Process*, ICSSP 2014, pages 65–69, New York, NY, USA, 2014. ACM.

[HP11] J.E. Hellbusch and K. Probiesch. *Barrierefreiheit verstehen und umsetzen: Webstandards für ein zugängliches und nutzbares Internet.* dpunkt-Verlag, 2011.

[IPF+11] Marko Ikonen, Elena Pirinen, Fabian Fagerholm, Petri Kettunen, and Pekka Abrahamsson. On the Impact of Kanban on Software Project Work: An Empirical Case Study Investigation. In *ICECCS*, pages 305–314. IEEE Computer Society, 2011.

[Kru14] Steve Krug. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability.* New Riders Publishing, Thousand Oaks, CA, USA, 3rd edition, 2014.

[LAKM07] Jonathan Lazar, Aaron Allen, Jason Kleinman, and Chris Malarkey. What Frustrates Screen Reader Users on the Web: A Study of 100 Blind Users. *International Journal of Human-Computer Interaction*, 22(3):247–269, 2007.

[LCG12] Marta Kristín Lárusdóttir, Åsa Cajander, and Jan Gulliksen. The big picture of UX is missing in Scrum projects. In *Interplay between User Experience and Software Development : I-UxSED 2012*, number 922 in CEUR Workshop Proceedings, pages 43–48. RWTH Aachen University, 2012.

[LGC10] Rui Lopes, Daniel Gomes, and Luís Carriço. Web Not for All: A Large Scale Study of Web Accessibility. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A)*, W4A '10, pages 10:1–10:4, New York, NY, USA, 2010. ACM.

[LSFM14] Olga Liskin, Kurt Schneider, Fabian Fagerholm, and Jürgen Münch. Understanding the Role of Requirements Artifacts in Kanban. In *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE 2014, pages 56–63, New York, NY, USA, 2014. ACM.

[May14] Philipp Mayring. *Qualitative content analysis: theoretical foundation, basic procedures and software solution.* SSOAR, 2014.

[MS10] Jacob E. McCarthy and Sarah J. Swierenga. What We Know About Dyslexia and Web Accessibility: A Research Review. *Univers. Access Inf. Soc.*, 9(2):147–152, June 2010.

[NBT13] Julius T. Nganji, Mike Brayshaw, and Brian Tompsett. *Describing and Assessing Image Descriptions for Visually Impaired Web Users with IDAT*, pages 27–37. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Nie93] Jakob Nielsen. *Usability Engineering.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[Nie14] Jakob Nielsen. *How People Read on the Web: The Eyetracking Evidence.* Nielsen Norman Group, 2014.

[NZP08] Karsten Nebe, Dirk Zimmermann, and Volker Paelke. *Integrating Software Engineering and Usability Engineering.* INTECH Open Access Publisher, 2008.

[oLSAP08] Austrian Ministry of Labour, Social Affairs and Consumer Protection. Behindertenbericht 2008 - Bericht der Bundesregierung über die Lage von Menschen mit Behinderung in Österreich 2008. Website, 2008. https://broschuerenservice.sozialministerium.at/Home/Download?publicationId=68; accessed 2016-12-28.

[oLSAP13] Austrian Ministry of Labour, Social Affairs and Consumer Protection. BEGLEITGRUPPE ZUM NATIONALEN AKTIONSPLAN BEHINDERUNG 2012 - 2020, STATISTIKEN. Website, 2013. https://www.sozialministerium.at/cms/site/attachments/5/1/5/CH3434/CMS1450699435356/statistik_-_menschen_mit_behinderung_20131.pdf; accessed 2016-12-28.

[Öst04] Bundeskanzleramt Österreich. E-Government-Gesetz. Website, 2004. https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20003230; accessed 2016-07-24.

[Öst05] Bundeskanzleramt Österreich. Bundes-Behindertengleichstellungsgesetz. Website, 2005. https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20004228; accessed 2016-07-24.

[Öst13] Bundeskanzleramt Österreich. Bundes-Verfassungsgesetz Art. 7 no 10000138. Website, 2013. https://www.ris.bka.gv.at/NormDokument.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10000138&Artikel=7&Paragraf=&Anlage=&Uebergangsrecht=; accessed 2016-07-24.

[Pam14] Jason Pamental. *Responsive Typography: Using Type Well on the Web*. O'Reilly Media, Inc., 1st edition, 2014.

[Par16a] European Parliament. DIRECTIVE (EU) 2016 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the accessibility of the websites and mobile applications of public sector bodies. Website, 2016. http://data.consilium.europa.eu/doc/document/ST-9389-2016-REV-1/en/pdf; accessed 2016-12-27.

[Par16b] European Parliament. European Accessibility Act. Website, 2016. http://ec.europa.eu/social/main.jsp?catId=1202; accessed 2016-12-27.

[PFPS12] 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*, CHI '12, pages 433–442, New York, NY, USA, 2012. ACM.

[PHD05] Helen Petrie, Ra Harrison, and Sundeep Dev. Describing images on the web: a survey of current practice and prospects for the future. In *Proceedings* *of 3rd International Conference on Universal Access in Human-Computer Interaction*, 2005.

[Pic16] Heydon Pickering. *Inclusive Design Patterns*. Smashing Magazine GmbH, Freiburg, Germany, 2016.

[PN12] K. Pernice and J. Nielsen. *Web Usability for Senior Citizens: Design Guidelines Based on Usability Studies with People Age 65 and Older*. Nielsen Norman Group, 2012.

[PWNG97] K. Pernice, K. Whitenton, J. Nielsen, and Nielsen Norman Group. Search and You May Find. Website, 1997. `https://www.nngroup.com/articles/search-and-you-may-find/`; accessed 2017-01-31.

[RBY12] Luz Rello and Ricardo Baeza-Yates. Optimal Colors to Improve Readability for People with Dyslexia. Website, 2012. `https://www.w3.org/WAI/RD/2012/text-customization/r11`; accessed 2016-12-29.

[RBY13] Luz Rello and Ricardo Baeza-Yates. Good Fonts for Dyslexia. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '13, pages 14:1–14:8, New York, NY, USA, 2013. ACM.

[RBY15] Luz Rello and Ricardo Baeza-Yates. How to present more readable text for people with dyslexia. *Universal Access in the Information Society*, pages 1–21, 2015.

[SB09] Sergio Sayago and Josep Blat. About the Relevance of Accessibility Barriers in the Everyday Interactions of Older People with the Web. In *Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibililty (W4A)*, W4A '09, pages 104–113, New York, NY, USA, 2009. ACM.

[SHBA+14] Mirjam Seckler, Silvia Heinz, Javier A. Bargas-Avila, Klaus Opwis, and Alexandre N. Tuch. Designing Usable Web Forms: Empirical Evaluation of Web Form Improvement Guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 1275–1284, New York, NY, USA, 2014. ACM.

[She14] K. Sherwin. Placeholders in Form Fields Are Harmful. Website, 2014. `https://www.nngroup.com/articles/form-design-placeholders/`; accessed 2017-06-23.

[SK10] O. Sohaib and K. Khan. Integrating usability engineering and agile software development: A literature review. In *Computer Design and Applications (ICCDA), 2010 International Conference on*, volume 2, pages V2–32–V2–38, June 2010.

[TN05] A Thatcher and M Ndabeni. HCI accessibility guidelines and illiteracy: developing a model of illiteracy and engagement with technology. *ergonomics SA*, 17:13–24, 2005.

[TR03] Mary Frances Theofanos and Janice (Ginny) Redish. Bridging the Gap: Between Accessibility and Usability. *interactions*, 10(6):36–51, November 2003.

[UBP$^+$15] Falk Uebernickel, Walter Brenner, Britta Pukall, Therese Naef, and Bernhard Schindlholzer. *Design Thinking: Das Handbuch.* Frankfurter Allgemeine Buch, Frankfurt am Main, 2015.

[Wik15a] BKA Wiki. Barrierefrei:EN 301 549. Website, 2015. `https://www.ag.bka.gv.at/at.gv.bka.wiki-bka/index.php/Barrierefrei:EN_301_549`; accessed 2016-12-27.

[Wik15b] BKA Wiki. Barrierefrei:OENORM EN ISO 9241-171:2008 11 01. Website, 2015. `https://www.ag.bka.gv.at/at.gv.bka.wiki-bka/index.php/Barrierefrei:OENORM_EN_ISO_9241-171:2008_11_01`; accessed 2016-12-27.

[Wik15c] BKA Wiki. Bundes-Verfassungsgesetz Art. 7 no 10000138. Website, 2015. `https://www.ag.bka.gv.at/at.gv.bka.wiki-bka/index.php/Barrierefrei:Rechtliche_Rahmenbedingungen`; accessed 2016-07-24.

Appendix

Transcript

Interviewee C

Was für ein Rolle hast du?

Projektmanager, was heisst das bei uns in der Firma? Das ist ja je nach Firma und Struktur der Organisation ein bisschen anders. Wir sind einerseits das was man Account Manager nennt, d.h. für die aktive Kundenbetreuung und Weiterentwicklung zuständig, wir machen operatives Projektmanagement im organisatorischen und koordinatorischen Sinn und machen auch Konzeptionsarbeiten in fachlicher und technischer Hinsicht. Das ist so das Spektrum was sich hinter meine Rolle an Arbeit befindet.

Du machst also auch Spezifikationen?

Ja.

Und arbeitest von Null an, also schon mit dem Designer usw. zusammen?

Ja, genau. Also Gesamtprojektverantwortlicher - qualitative Verantwortung für das was hinten rauskommt liegt beim Projektmanager.

Ihr seid ja in der Webbranche tätig. Achtet ihr grundsätzlich auf Barrierefreiheit?

Wir haben ja einige Kunden aus dem Behörden-Umfeld und einer meiner Accounts, die ich auch direkt betreue ist ein Landtag. Und da ist das BITV - die barrierefreie Internet und Technik Verordnung - so heisst das Gesetz in Deutschland. Das österreichische Pendant dazu ist das Behindertengleichstellungsgesetz. Also da ist das BITV 2.0 und die sind gebunden daran, dass sie sich an gewisse Barrierefreiheits-Richtlinien halten. Insofern, bei dem Kunden ist es ein Muss, da wird die Frage gar nicht gestellt. Bei unseren privaten Kunden überlassen wir es dem Kunden, ob er es für seine Zielgruppe relevant findet. In der Regel ist es dort eher kein grosses Thema. Es kommt natürlich immer auf die Zielgruppen an, aber bei, zum Beispiel Corporate and Finance, die juckt das eher nicht, das Geld wird lieber in Features gesteckt als in Barrierfreiheit.

Wer ist bei euch zuständig für das Ganze? Also wenn wir von einem Kunden ausgehen, für den die Barrierefreiheit wichtig ist, wer überwacht diese Barrierfreiheit? Wer garantiert das? Bei wem liegt die Verantwortung?

Wir haben eine gewisse Richtlinie, dass wenn ein Behördenprojekt oder ein Projekt das barrierefrei sein muss, bevor das Projekt freigegeben werden darf seitens der Projektsteuerung, also wir Projektleiter. Das Ganze funktioniert nach dem Prince II Prinzip, wo der Projektleiter immer einem Steuerungskreis quasi Rechenschaft ablegen muss. Der kann aus dem Kunden bestehen, oder bei uns ist es in der Regel intern bei der Abteilungsleitung und Geschäftsleitung vorzustellen. Man muss also intern erst Rechenschaft ablegen, bevor die sagen, ok, du darfst mit deinem Projekt live gehen, muss es einen Barrierefreiheitstest geben. Also den BIK Tests, das ist ein Verein in Deutschland, die einen Online Test, so einen Selbsttest, entwickelt haben. Den klickt man durch und dann bekommt man Punkte. Gewisse Punktezahlen müssen erreicht werden und nur dann darf so eine Site live gehen. Das heisst, ich bin als Projektmanager verantwortlich. Den Test führt eine eigene Mitarbeiterin bei uns, das ist eine Barrierfreiheits-Expertin quasi, und die macht diese BIK Tests. Wenn wir mit Externen zusammen arbeiten, was gerade bei Frontend Entwicklung (HTML, CSS Erstellung) immer wieder vorkommt, dann bekommen die das schon als Vorgabe und müssen uns selbst getestetes liefern. Und unsere Internen werden in der Regel schon direkt nach Fertigstellung der ersten Prototypen getestet und dann wird bevor die Website live geht nochmals getestet. Dass das passiert, bin ich als Projektmanager verantwortlich, ich hab aber jemanden der die Tests durchführt. Ich muss nicht selber auf Barrierefreit prüfen.

Du selber hast also keine Kenntnis, bzw. wie groß ist dein Wissen über Barrierefreiheit?

Das Wissen ist eigentlich schon ganz gut ausgeprägt, weil wir ja auch mit dem Kunden sprechen müssen und man spricht über gewisse Dinge und stellt fest, die wollen das und das haben und man muss halt dann gerade bei so Kunden, bei denen Barrierfreiheit ein muss ist, auch mal sagen "aber moment... das könnte Probleme geben, weil.." oder "da müssen wir berücksichtigen, dass...". Also da geht es eher darum, dass wir im Beratungsaspekt im Hinterkopf haben, was das denn tatsächlich bedeutet und was für Implikationen das hat. Oder auch bei der Technologieauswahl. Da hatten wir zum Beispiel ein riesiges Problem mit einem Videomanagement System, das wir bei einem behördlichen Kunden eingesetzt haben. Die haben uns die neueste Version von deren Videoplayer gegeben und in der Version war das Untertitelfeature noch nicht fertig. Das war eigentlich crazy, dass die das bereits für die Produktion freigegeben hatten. Da mussten dann wir ganz kurzfristig ein Feature für Untertitel draufbauen. Also ich muss schon ein Grundverständnis haben, aber eher für Risikovermeidung und zur Beratung des Kunden und nicht weil ich bewerten muss, ob das was wir liefern auch tatsächlich barrierefrei ist.

Du hast vorher gesagt, dass ihr Externen einfach die entsprechenden Spezifikationen gebt, das es barrierfrei sein muss. Sagt ihr da einfach nur die Seite muss beispielsweise AA Konformität aufweisen oder listet ihr auf was genau sie berücksichtigen müssen? Wie schauen diese Specs genau aus?

Wir sagen denen, du musst BITV 2.0 machen, das entspricht in etwa AA, bietet in gewissen Dingen aber ein wenig mehr Interpretationsspielraum und ist nicht ganz so technisch. Und die bekommen von uns einfach die Richtlinie und vom BIK gibts dann so eine Seite mit Erläuterungen und die Selbsttests. Diese Inputs bekommen die und damit müssen sie dann zurecht kommen. Wobei, wir haben externe Partner mit denen wir immer wieder zusammenarbeiten und die kennen das Prozedere bereits und sind auch vertraut damit. Wir hatten aber auch schon Probleme mit anderen Dienstleistern, wo das nicht der Fall war. Das waren Dienstleister aus dem südasiatischen Raum, und die hatten keine Ahnung von Barrierefreiheit. Denen hatten wir zwar auch Inputs geschickt, was wir wollen, aber das hat gar nichts gebracht. Das war visuell erfüllt, aber sonst nichts.

Macht ihr im Unternehmen für eure Entwickler spezielle Schulungen, dass die dann auf Barrierefreiheit achten?

Nein. Das ist die Verantwortlichkeit von unseren jeweiligen Teamleitern. Also unser Teamleiter für die Frontend-Entwicklung der hat bereits mehr als 15 barreirefreie Websites gebaut, mit teilweise auch - es gibt beim BIK so ein Punkterating - mit so +90 oder +95 Club - also wie die barrierefreien Websites sind. Und es kommt dann oft in den Ausschreibungen von den Behörden, dass sie sagen man muss mindestens 90 Punkte für die gesamte Website haben, oder 95 Punkte für die Startseite und so weiter. Also der kennt sich da sehr gut aus und der schaut darauf, was seine Entwickler machen und das die das entsprechend umsetzen. Aber Schulungen gibt es nicht, das ist mehr so ein on-the-job Training. Es fällt dann in der Regel beim Test auf, dass was falsch gemacht wurde und dann kommt es wieder zurück zum Entwickler und so lernen das die Entwickler. Aber dezidierte Schulungen haben wir nicht, nein.

Macht ihr da in dem Fall kontinuierliche Tests? Also auch die Barrierefreiheit wird kontinuierlich mitgetestet?

Nein. In der Regel, wenn wir von einem klassischen Entwicklungsprojekt ausgehen, im Sinn von neuen Webseiten, dann haben wir zwei Phasen. Das Erste ist, wenn du den Klickdummy hast, also das HTML in seiner Struktur steht, dann einmal. Und dann wenn die Zielinterpretation fertig ist. Also nur an den zwei Punkten.

Hast du ein Gefühl dafür wie weit Barrierefreiheit im Web allgemein schon umgesetzt ist? Achtest du überhaupt auf das?

Ich glaub, dass es durch responsive wieder einen ziemlichen Dämpfer bekommen hat. Wie wir noch Webseiten entwickelt haben in einem starren Layout waren alle schon so eingespielt, dass sich alle diesem nicht sofort sichbaren Qualitätskriterium angenähert hatten. Aber das Ganze responsive und multi-Client-Thema ist so ein großer Faktor dazu gekommen, für den man Kompromisse eingehen muss. Also meiner Erfahrung nach, man muss also für responsive schon Kompromisse eingehen und deswegen ist die Kompromissbereitschaft beim Thema Barrierefreiheit auf den zweiten Rang zurückgestuft worden. Im Web an sich, das ist doch standardisierter geworden. Also mit HMTL5 und die Browser haben jetzt auch gut nachgezogen und die Entwicklungszyklen sind viel kürzer. Man kann neuere Technologien einsetzen. Ich glaub das Web insgesamt ist schon standardisierter, aber die Barrierefreiheit... ich glaub der Hype hat wieder ein bisschen abgenommen, vor allem auch in den Köpfen der Kunden primär. Ich hab keinerlei statistische Daten dazu, das ist lediglich mein Eindruck.

Du hast ja gesagt bei deinen Kunden sind das eher nur die Behörden, die sich dafür interessieren. Glaubst du, dass also nur die gesetzlichen Verbindlichkeiten ausschlaggebend sind?

Wir haben das bei den Kunden total unterschiedlich ausgeprägt. Manche Kunden sagen, das ist absolut wichtig und da gibts keine Kompromisse. Und ich hab auch schon von Kunden gehört, die gesagt haben, solange mir die Interessensvertretung der Barrierevereine quasi, nicht auf den Socken geht, habe ich kein Problem. Also den stören Defizite erst wenn ihn deswegen jemand anklagt. Also nicht zwingend im juristischen Sinn, sondern generell. Also das ist sehr unterschiedlich. Aber zurückkommend auf die Frage, der Tenor ist ganz klar, weil sie müssen. Ich glaub, wenn sie nicht müssten, würde mindestens die Hälfte meiner Kunden das Geld in irgendwelche visuell hübsche Features investieren, als barrierefrei zu sein.

Glaubst du das Barrierefreiheit so ein grosser Mehraufwand ist? Also tatsächlich ein großer Kostenfaktor?

Nein, aber man muss Kompromisse eingehen an manchen Stellen und es schränkt einen schon manchmal an manchen Stellen ein bisschen ein. Ich glaub das ist das, wo gewisse Kunden dann die Barrierefreiheit nicht vermissen würden, wenn sie nicht mehr da wäre. Weil dann könnten sie irgendwelche superfancy Animationen machen. Konkret war das Thema bei einem Kunden. Die wollten eine Website haben, die gerade junge Leute ansprechen sollte und deswegen war das Thema Animationen und optischer Schnickschnack sehr stark, aber wir mussten immer auf Kontextwechsel achten. Wir durften hier nicht viele DOM Manipulationen machen. Das es nachher für Leute, die die Animationen nicht sehen, weil da gabs schon eigentlich auch Inhalte für Leute, die das betrifft. Wir mussten also alle abdecken, nur halt doch den Großteil nicht. Und da war unsere Herangehensweise so, dass wir gesagt haben, wir können es nicht so extem machen wie ihr es eigentlich schön fändet, weil ihr sonst nicht barrierefrei seid. Also es ist halt schon ein Einschränkungsfaktor. Weniger die Kosten, das ist ja, wenn man das von vornherein mitdenkt, dann ist das ja nicht viel mehr. Das bißchen mehr an Testing und die eine Korrrekturschleife. Kosten ist eigentlich nicht das Problem, eher die Einschänkungen in der Spielfreiheit. Kommt halt auch darauf an, wenn man jetzt zum Beispiel eine Ministeriumswebseite hernimmt, die ja eh einen seriösen Look haben sollte, hat man damit auch weniger Probleme. Aber grad wenns darum geht eine junge Zielgruppe zu erreichen, die mit Snapchat und Co ein ganz anderes Level an visuellen Gimmicks gewohnt ist und dann muss man die mit einer barrierefreien Webseite ansprechen, kanns schon schwierig werden. Oder auch wenn man an so Sachen wie Infografiken beziehungsweise animierte Informationen denkt. Das ist ja dann ein informationstragendes Element. Und entweder ich hab dann hier eine Audiobeschreibung und da gehts dann schon um einen Kostenfaktor. Wenns darum geht, wir brauchen eine externe Dienstleistung für Audiotranskriptionen oder die Tonspuren erstellen oder wie auch immer da würde es dann teuer werden. Da wollen die Kunden dann schon lieber schummeln oder halt eben die Barrieren in Kauf nehmen und dafür dann halt an anderen Stellen dafür sauberer sein, sodass sich das von dem Punkteranking, das benötigt wird, ausgeht.

Das bringt mich gerade zum Thema Designer. Die sagen ja auch des öfteren, dass sich barrierefreie Webseiten oder besser Barrierefreieheit und schönes Design stehen sich gegenüber.

Zustimmendes Ja vom Gesprächspartner.

Glaubst du das barrierefreie oder besser gesagt funktionierende Webseiten - Usability ist hier ja auch ein besonders wichtiger Faktor - auch schön sein können?

Auf jeden Fall, ja. Es war jetzt so ein Designtrend - oder ich weiss nicht so genau, ob man das als Trend bezeichnen kann. Ich hab auf jeden Fall viele Designs gesehen, die viel mit zarten Farbtönen, also so Pastellfarben oder helle Grautöne und so, sind mir in letzter Zeit oft über den Weg gelaufen und da ist man kontrastmäßig halt schnell am Ende. Was wir jetzt einfach gemacht haben, für einen unserer Kunden, der unbedingt so einen Look wollte, Kontrastlösungen gebaut. Die dann das CSS, dadurch dass wir mittlerweile Iconfonts und so was haben ist es ja eh einfacher geworden das zu machen. Und dann haben mir mit CSS wirklich harte Kontraste eingebaut, also mit schwarz und weiss. Es kommt halt wie gesagt immer darauf an, was ist gerade Trendy, was will der Kunde, was findet er schön. Vor allem was ist schön, das ist ja immer so ein bißchen subjektiv. Also es geht auf jeden Fall, und sonst muss man sich halt behelfen. Ganz ehrlich bei dem Kunden finde ich sogar die Kontrastversion schöner.

Nochwas, das du vorher schon kurz angesprochen hast - hier geht es jetzt wieder um den Aufwand. Kannst du das irgendwie einschätzen, wie weit die Kosten auseinander gehen, wenn man Barrierefreitheit von Begin an mitberücksichtigt versus einer nachträglichen Integration? Macht ihr das überhaupt? Dass ein Kunde einfach nur sagt er will eine Überarbeitung, rein wegen der Barrierefreiheit?

Ich hab genau für einen Kunden, dem es bisher egal war, habe ich gerade eine Anfrage vorliegen. Die habe ich noch nicht beantwortet. Bei denen geht es konkret um die Bedienung mit Screenreadern. Ich weiss jetzt ehrlich gesagt nicht genau wo bei denen der Wind da herweht, also was der Beweggrund dafür ist. Aber da hab ich jetzt die Anforderung und ich hab das mal im Kopf überschlagen, also mir das einfach mal durchgedacht und gedacht, wo könnten wir Problemstellen haben. Und frag mich nicht wie viele wir haben könnten. Wenns einfach konzeptionell von vorne nicht mitbedacht ist, man muss ja dann ganz vorne anfangen. Also nicht nur welche technischen Lösungen funktioniern vom Konzept her nicht, was haben wir für Alternativen und dann halt wirklich vom Design beginnend alles durchgehen. Also die Kosten sind, sicherlich um ein vielfaches höher, als wenn man es von vornherein bedenkt. Wobei da auch wieder die - wir haben zum Beispiel auch einen Kunden von uns, der viel Wert darauf legt. Die haben auch alle PDFs, die sie auf der Webseite haben, barrierefrei ausgezeichnet. Für alle Videos, die sie haben gibts eine Audiodeskription, sogar für die Livestreams die sie haben gibts Gebärdendolmetscher. Also denen ist das thematisch einfach wichtig. Und da, angenommen, das wäre erst später wichtig geworden, sind hier zum Beispiel glaube ich die technischen Kosten kleiner, als die inhaltlichen, also die Kosten, die die haben um das zu überarbeiten und so weiter. Aber wenn man jetzt nur den technischen Part der Website betrachtet, spart man sich halt wenn man es von vorne herein mitbedenkt diese eine Konzeptionsphase und nochmal umsetzen und Testen und so.

Was glaubst du, was sind die Gründe warum eigentlich so wenig auf Barrierefreiheit geachtet wird?

Ich glaube viele sehen die Zielgruppe einfach nicht. Also, die Frage ist ja auch immer, was ist der Unternehmenszweck. Wie viel verlier ich, oder was kostet es mich, wenn ich diese Zielgruppe nicht beachte? Bei Ministerien, wenn sie es nicht machen, haben sie ein rechtliches Problem und ein Reputationsproblem, weil das wird in der Regel ja in den Medien sofort ausgebreitet. Aber nehmen wir jetzt zum Beispiel wieder einen Kunden von uns, DAX30 Unternehmen, der hat Millionen von Kunden, 15.000 Mitarbeiter - es interessiert die aber in Wahrheit nicht. Weil, was sind die Kosten dafür, wenn sie es nicht machen? Wenn ein Verein ihnen irgendeine Email schreibt, der das auf seinem Portal veröffentlicht. Die Finanzwelt, die sich für deren Presse interessiert, die juckt das nicht. Die allgemeine Öffentlichkeit, die deren Produkte kauft, die kriegt das auch nicht so mit. Ich glaub das Thema ist einfach, ich muss schon irgendwie in meiner Zielgruppe mit der Gruppe die von Barrierefreiheit profitieren zu tun haben, um das auch als wichtig zu erachten. Und im Umkehrschluss, der Reputationsverlust oder die Einbußen müssen für mich so erheblich sein, dass sich das lohnt. Ich glaube es ist irgendwo einfach auch eine wirtschaftliche Entscheidung, ob man das macht oder nicht. Schon auch eine ethische, sollts eigentlich auch sein, oder sagt Gleichstellung - die ist ja auch gesetzlich festgesetzt, aber meine Erfahrung gerade bei der Privatwirtschaft. Wo kein Kläger, da kein Richter. Da ist der Verlust, auch der Reputationsverlust, größer, wenn der Aktienkurs mal drei Tage runterfällt. Da haben die mehr schlechte Presse, als wenn sich jemand über die Barrierefreiheit ihrer Webseite beschwert.

Wie schauts bei solchen Kunden mit der Usability aus? Ist denen dann grundsätzlich egal, ob ihre Seite benutzbar ist?

Also bei dem Kunden, an dem ich jetzt konkret hier gedacht hatte, wir habens einmal versucht sie zu Usabilitytests zu bringen. Wir haben da auch eine Zielgruppenanalyse gemacht - das fanden die dann ganz toll. Seitdem sagen sie auch, wir kennen ja unsere Zielgruppen und entscheiden nun quasi eigenmächtig für ihre Zielgruppe. Und um das zu verifizieren, ob das was sie für ihre Zielgruppe entscheiden richtig ist oder nicht, versucht man über das Tracking so ein bißchen machen, aber in Wahrheit passierts nicht. Also wirklich UX betreiben und regelmäßig irgendwelche Datenanalysen erheben, Daten messen, Befragungen und Tests machen geht so. So was ist immer schwierig zu verkaufen. Wenn die Kunden zu uns kommen, dann sagen wir denen natürlich, dass man vorab die Zielgruppe definieren muss, man muss die Probleme und Nutzungsszenarien erstmal wissen. Das ist erstmal schon immer schwierig, weil das sind ja die Online Redaktion oder die Kommunikationsabteilungen, Presse oder wer auch immer - die sind ja, oder denken es zumindest, Experten auf ihrem Gebiet. Und dann sagt man ihnen, das man noch so viel in Erfahrung bringen muss, um überhaupt eine gute Website machen zu können. Dann können die das natürlich nicht immer so gut verstehen. Die sagen dann, wir brauchen ja nur eine Webseite. Ich glaub aber schon, dass das ein Trend ist, der gerade im kommen ist, dieses UX Thema. Um zu prüfen, ob die Webseite gut funktioniert. Aber weniger in der Bedienbarkeit, sondern eher in der Auswirkung für den Kunden. Also nach dem Motto, wie schnell können Informationen gefunden werden, wie gut ist mein Interface bedienbar oder hab ich auch tatsächlich alle Dinge beispielsweise ein wichtiges pdf oder so, wo ich weiss, dass es der User benötigt auf der Startseite verlinkt. Vom Ansatz her also eher, dass man versucht sich Gedanken zu machen was will der User als ich mach eine Webseite, die so gut funktioniert, dass jeder sich zurecht findet. Gerade auch wenn man das Thema Personalisierung und Targetting, das kommt bei uns gerade recht stark und in dem Zug muss man sich auch mehr Gedanken darüber machen, was wollen die User, die einzelnen, eigentlich wirklich von meiner Website? Und das ist ja eigentlich schon ein ganz anderer Zugang, als wenn ich sage ich will ein gut bedienbares Interface, sondern ich versuche die Bedürfnisse individuell dann abzubilden.

Inwieweit ist UX grundsätzlich bei euch ein Thema?

Ist gerade allgemein groß im kommen. Also thmatisch alles, was Fortbildungen, Tools und so weiter betrifft. Und man merkt, dass immer mehr Kunden auch bereit sind, da Geld in die Hand zu nehmen. Dadurch, dass das Thema insgesamt größer geworden ist, also Tools sind besser, da gibts ja auch so Sachen wie Crowd-Usability-Testing, was die Kosten für Usability einfach drastisch minimiert hat, als wenn man immer Leute einladen muss und schauen muss was machen die denn alles usw. Kostet nun halt alles nur ein Bruchteil davon. Das ist also schon alles stark im kommen. Es ist noch nicht so, dass Kunden uns Anforderungskataloge und so schicken, da stehts noch nicht drinnen - so weit sind wir noch nicht, aber wenn man Kunden hat, die am Booth der Zeit sind, dann kennen die das Thema schon, haben schon was davon gehört und sagen, ja das Thema müssen wir jetzt auch mal angehen.

Das heisst, wenn es vom Kunden kommt, dann wird das auch bei euch von Anfang an mitberücksichtigt? Also zum Beispiel schon beim Wireframing.

Ja genau. Von Anfang an, regelmäßig verifiziert. Also wir haben uns jetzt bei einem Projekt wirklich Zielgruppen-Tester besorgt, die haben schon die Wireframe Prototypen bekommen und getestet, dann die HTML Prototypen getestet, dann die fertige Website wieder getestet. Also wirklich den ganzen Entwicklungszyklus der Webseite begleitet, um auch ständiges Feedback einzuarbeiten. War sehr spannend. Teilweise auch schwierig, als Dienstleister, wenn man dann darauf kommt, dass das was man sich überlegt hat eigentlich gar nicht funktioniert und dann wieder zurück muss. Und dann muss man das dem Kunden erklären, dem man ja gesagt hat, wir machen da jetzt Usability Tests und dann merkt, vieles was man sich übelegt hat wird wieder über den Haufen geworfen. Das darf nicht zu oft passieren, sonst ist die Frustration zu groß.

Wo sollte deiner Meinung nach innerhalb eines Projektteams die Verantwortung liegen für Barrierefreiheit (und auch Usability)? Beim Projektmanager, beim Entwickler oder

Externe?

Ich glaube es ist, so wie das bei uns grundsätzlich organisiert ist, schon richtig. Der Projektmanager trägt die Verantwortung dafür, dass es im Projekt gemacht wird, also bedacht wird. Das muss irgendwo auf eine Checkliste oder irgendwo muss das stehen, das "Ja, hab ich bedacht". Die Expertise, sehe ich eigentlich falsch, wenn sie beim Projektmanager liegt. Die muss irgendwo - es gibt da Schnittmengen zu Frontend. Also bei uns macht es eigentlich jemand der im First-Level-Support arbeitet. Was total schräg ist, weil dort gehört es eigentlich nicht hin, aber historisch einfach so entstanden. Aber eigentlich sehe ich es eher im Frontend-Bereich. Also in der Regel - ich habe gerade überlegt, wenn ich ein Problem habe, wo gehe ich jetzt hin? Also nicht, wer sollte diese Rolle jetzt inne haben, sonder nur, es ist fünf vor zwölf, du hast ein Problem bei der Barrierefreiheit und zu wem gehst du nun? Und ich glaube, ich würde zu meinem Teamleiter Frontend gehen. Weil der sich einfach fachlich am Besten damit auskennt. Jetzt nicht zum analysieren und rausfinden was nicht in Ordnung ist, sondern was halt die Lösung dafür ist. Und das so ist eigentlich eine ganz gute Kombination.

Wie können deiner Meinung nach entsprechende Spezifikationen ausschauen, um die Accessibility Vorgaben zu erfüllen?

Jetzt gerade habe ich darüber nachgedacht, du hast mich hier auf was gebracht. Ich schreib nämlich gerade eine Spezifikation. Bei uns gibt es in der Regel oben im Dokument die Referenz auf die Anforderung im Lastenheft. Also das ist jetzt der Idealfall, du hast ein Lastenheft vom Kunden bekommen, da stehen die Anforderungen drinnen, was der will und dann gibts die Referenz zum Layout dazu und jetzt geht es also um CMS Spezifikationen für die Umsetzung. Und da wär eigentlich genau der richtige Punkt, das man ansetzt, das man in der Anforderungsspezifikation nicht nur einen Punkt hat in dem dann Barrierefreiheit drinnen steht, sondern das man aus der Verordnung oder den Guidelines, die es ja auch von den WCAG zum Beispiel gibt. Das man hier sagt, wir haben zum Beispiel das Video-Modul in dem der Player und alles mögliche drinnen ist. Und wenn ich mir dann eigentlich hier das Snippet aus der Guideline in meine Anforderungsdokumentation hineingeben würde - das wäre eigentlich der richtige Punkt. Quasi. Also zum Beispiel, sag ich ja, hier steht der Fließtext drinnen. Dann sag ich was der Fließtext alles können soll und dann schreib ich da auch rein, was gibts hier zu berücksichtigen, also Abkürzungen, Sprachwechsel. Also aus Barrierfreiheit - was muss berücksichtigt werden, das müsst ich hier jetzt eigentlich reinschreiben. Diese nicht-funktionalen Geschichten pro Umsetzungsmodul in die Anforderungsdokumentation mit aufnehmen. Das wär eigentlich geil - machen wir nicht, aber das fände ich gut. Also es gibt bei uns die nicht funktionalen Anforderungen, da steht dann die Barrierefreiheit drinnen und aus. Und bei der Barrierfreiheit gibts dann halt auch immer die entsprechenden Referenzen aus dem Standard, die eingehalten werden müssen und so. Aber schöner wärs eigentlich, also für die Entwickler wäre es einfacher, wenn er einfach die Spezifikation bekommt, liest sich die Anforderungen und er sieht gleich was vom technischen Konzept als Lösungsweg gedacht ist und sieht dann auch schon, ok ich muss mir also überlegen, wie ich einen Sprachwechsel mache oder darauf achten, dass das funktioniert. Ja, das wär eigentlich schon schön.

Die WCAG Richtlinien kennst du ja.

Kennen schon, aber ich arbeite eigentlich nur mit der BITV.

Inwiefern unterscheiden sich die?

Die sind sehr ähnlich ja. Also die BITV sagt halt, also definiert für gewisse Bereiche oder Anwendungsfälle welches Level akzeptabel ist, um insgesamt der Gleichstellung gerecht zu werden. Also, weil wenn man überall AAA haben will, ist das schon heavy. Und die definieren quasi, für bestimmte Bereiche, was denn akzeptabel ist.

Und sind die verständlich geschrieben?

Die BITV selbst nicht. Deswegen arbeiten wir auch hauptsächlich mit bik-online.de. Das ist dieser Verein für Barrierefreiheit in der Kommunikation und die haben eben so Guidelines und diesen Selbsttest und da gibts auch schon einen entwicklungsbegleitenden Selbsttest und das ist so unser Hauptarbeitsmittel. Da ist quasi interpretiert, was in der Verordnung halt nicht so genau herauslesbar ist.

Das Interview wäre eigentlich um, vielleicht möchtest du mir noch etwas auf den Weg mitgeben?

Ja, also ich hab selbst eigentlich so ein gespaltenes Verhältnis zu Barrierefreiheit, weil einerseits denk ich mir schon, es ist notwendig, dass man das macht, aber ich hab auch schon oft darunter gelitten. Gerade auch bei so knappen Projektlaufzeiten, war das halt teilweise so ein Übel, das noch on-top gekommen ist. Und was ich mich auch gerade frage, unsere Entwickler, die also neu anfangen. Kennen die das? Haben die schon von BITV gehört, wenn die grad frisch von der Uni kommen? Ich glaub eigentlich, die haben das noch gar nie gehört. Also vielleicht sollte man hier beginnen und schauen, dass Barrierefreiheit in den Lehrplan mit aufgenommen wird? Weil die Gesetzeslage ist mittlerweile ja eindeutig. Ja, aber ich glaube der Hype ist jetzt ja eigentlich leider vorbei, bzw. hat sich gesetzt.

Da gibt es kontroverse Aussagen. Die Entwickler, die ich interviewt habe, haben beide gesagt, dass sie das Gefühl haben, dass die Thematik immer mehr in Mode kommt. Und zwar mit dem Hintergrund, dass nun immer mehr große Firmen wie Microsoft und Google auch tatsächlich dahinterstehen und schauen, dass sie barrierefrei sind und es dadurch in aller Munde ist. Zusätzlich bringt ja Barrierefreiheit auch im SEO Bereich Vorteile und auch Machine Learning ist immer stärker im kommen.

Beim SEO Bereich... ich glaub das befruchtet so gegenseitig - je besser du dich an moderne Standards hältst, desto besser finden dich die SEO Leute und nebenbei hat das auch einen positiven Einfluss auf die Barrierefreiheit. Wenn man überlegt, allein im Vergleich - ich habe gerade eine alte Website, ein Frontend, auf eine neue umgestellt, von einem ein paar Jahre alten XHTML 1.0 Transitional basierten auf ein HTML5 basiertes. Wie einfacher und schöner es jetzt möglich ist semantisch korrekte Inhalte abzubilden, als das was sie sind. Und auch innerhalb des Codes - mit section und innerhalb noch ne section. Die Möglichkeiten sind hier einfach viel schöner geworden. Insofern glaube ich, die Verbreitung der Standardisierung und die schneller werdenden Entwicklungszyklen der Browser, weil das ist das was uns... stell dir vor wir müssten responsive machen und hätten nicht Browser, die sich alle 2 Wochen mit neuen Versionen updaten - und die mobile Browser würden sich nicht updaten, sondern immmer nur mit dem Betriebssystem updaten, so wie das beim IE früher war .. ich würde sterben. Keiner würde mehr im Web arbeiten wollen.

Arbeitet ihr agil?

Nicht per Definition - sondern eher nur so ein bisschen. Von den Prozessen her, verwenden wir das Prince2 Prozessmodell, das geht von Projektphasen aus und man definiert sich so viele Phasen wie man benötigt und kann diese quasi eigentlich auch als Sprints aufsetzen. Und wann welche Anforderungen erfüllt werden oder wie man mit den Anforderungen vorgehen will, ist diesem Modell egal. Gerade bei Behörden-Kunden mit öffentlichen Ausschreibungen, Lastenheften und Festpreisen geht das nicht. Sondern da gibt es eine big specification upfront und danach eine change-request Orgie. Bei der öffentlichen Hand, wo man auch Aufwände-Budgets hat, da würde ich es als agil bezeichnen, auch wenn wir kein klassisches System wie Scrum oder Kanban in unseren Prozessen festgeschrieben haben. Aber wir überlegen uns alle 2 Wochen was aus unserer Roadmap wir als nächstes entwickeln. Das ist dann quasi agil. Glaubst du, macht es einen Unterschied, ob man .. also gerade bei agil... man schreibt ja hier am Anfang nicht so große Konzepte. Sondern es gibt use-cases und einen backlog und man orientiert sich immer mehr danach, was das können soll. Aber gerade das Thema Anforderungsverfolgung, das wenn du ein Wasserfallmodell betrachtest, da heisst es ja deine Testfälle müssen deine Anforderungen verifizieren. Wenn man das betrachtet, müsste eigentlich Agilität schlechter sein für Barrierefreiheit. Weil steht in einem use case, bzw. user story oder wie man es nennen will .. natürlich kannst du hier definieren "Ich kann mit meiner Tastatur die Seite bedienen..." oder "Ich kann mir die Seite vorlesen lassen und verstehe die Struktur" - ja, dann geht es. Aber ich habe noch nie gesehen, dass das gemacht wird. Aber es wäre eigenltich schon gut. Aber es geht ja auch niemand so strikt nach PM Modell vor. Wobei wir bei einem Kunden, schon stringente Regeln gehabt haben - klassisch, nach Entscheidungen. Es muss eine Entscheidung geben, sonst wird nicht gehandelt. Deswegen war mit denen auch so ein V-Modell total easy und man musste nicht über change requests diskutieren. Sondern die haben sich strikt an die Spezifikation gehalten. Und da gab es dann tatsächlich Testfälle für Tastaturbedienbarkeit. Und der Tester musste tatsächlich Maus beiseite legen. Das war da wirklich alles als Testfälle dokumentiert und wir mussten dann auch Testprotokolle abliefern, dass wir wirklich die Tests gemacht haben. Aufwand insgesamt war natürlich enorm, aber nachdem das Projekt live ging hatten wir gerade einmal 6 gemeldete Bugs. Das hat schon gut funktioniert.

Interviewee E

Was für eine Rolle hast du?

Frontendentwickler.

Wie lange bist du schon FE, bzw. wie lange schon fokussiert auf A11Y?

Seit ca 1 Jahr spezialisiert, arbeite aber schon sehr lange in der Webentwicklung.

Hattest du spezielle Schulungen oder Trainings im Bezug auf A11Y? Wird das bei dir im Unternehmen irgendwie forsiert/unterstützt?

Ja, bei uns wird das auf jeden Fall unterstützt. Direkte Schulungen hatte ich zwar noch nicht, aber die Option besteht schon. Und ich war zb auch bei der Accessibility Tagung in Wien und wir haben eigene Experten inhouse, die legen auch wert darauf, dass all unsere Projekte entsprechend umgesetzt werden und dementsprechend steht die ganze Thematik hoch im Kurs bei jedem Projekt.

Und fühlst du dich schon als A11Y Entwickler - wenn man das so nennen darf?

In dem Sinne nicht, nein, aber ich sehe definitiv Änderungen im Vergleich zu früher, dass man halt einfach von vornherein schon dran denkt. Dass man bei gewissen Punkten einfach schon weiss, das sind die Standard - Main Points beim Entwickeln und da geht man schon von vornherein anders auf eine Aufgabe heran. zb Schriftgrößen beachten, Color-Contrast.... diese Punkte werden auf alle Fälle von mir nun mehr wahrgenommen. Dass das dann auch perfekt umgesetzt wird, ist wieder ein anderes Thema. Aber ich nehme es mehr wahr und versuche darauf zu achten. Ich glaube wir müssen weg von dem Punkt, dass A11Y explizit berücksichtigt werden muss, sondern es sollte schon von Haus aus so sein, dass es Best-Practises gibt, die das Ganze unterstützen.

Also dass du bereits entsprechende Spezifikationen und Anfoderungen bekommst? Oder Grundsätzlich?

Ja, gerade beim Frontend, sollte es Best-Practises geben, also Spezifikationen, die definieren - wenn ihr (Devs) schon so etwas macht, dann macht ihr es am Besten so und das sollte bereits eine Berücksichtigung der Barrierefreiheit beinhalten. Das sollte beinhalten - was sind die Standards, was kann ich wie umsetzen, so dass es eine Routine wird.

Würdest du also die Verantwortung für eine barrierefreie Umsetzung beim Entwickler sehen?

Ich glaub das ist eine geteilte Verantwortung - der Entwickler ist dafür verantwortlich was er entwickelt und darin ein Experte. Wenn dieser nun beim Design schon feststellt, dass es bei der Form der Umsetzung zu Problemen mit der Barrierefreiheit kommen könnte, dann sollte dieser das bereits in dem Prozessschritt einbringen. Aber Vollverantwortlich kann nicht der Entwickler sein, da dieser ja auch nur Anforderungen umsetzt. Grundsätzlich glaube ich, dass es eine geteilte Verantwortung aller Beteiligten ist.

Du glaubst also nicht, dass es spezielle Spezifikationen oder Vorgaben geben muss, sondern grundlegende Best-Practices, die jeder Stakeholder kennt und damit sollte es quasi klappen?

Nein. Ich glaube schon, dass es Standards, wie WCAG geben muss, um Probleme zu identifizieren und das Bewusstsein zu schaffen. Aber dennoch sollte es einen roten Faden geben, der zeigt an was man sich halten sollte. Um die Dinge anhand des Standards prüfen zu können.

Du erwartest dir also nicht, dass wenn du ein Projekt übernimmst, dass dann in den Spezifikationen konkret Dinge wie "Achte darauf, dass die Kontraste so und so sind" ...

Das kommt darauf an. Gerade für den Anfang wäre das schon praktisch, wenn es konkrete Anforderungsspezifikationen gäbe. Aber die grundlegendsten Dinge wie Kontrast zb finde ich sollte ein Entwickler einfach bereits standardmäßig berücksichtigen. Wobei es sicherlich hilfreich wäre, eine Erinnerung zu haben - wie eine Checkliste. Grundsätzlich ist es natürlich so, dass es drauf an kommt, auch ob es ein bestehendes Projekt ist, oder ein neues. Bei Neuen, hat man noch mehr Freiheiten und Möglichkeiten. Oder macht man nur eine schrittweise Verbesserung?

Und würdest du sagen, dass der Aufwand für eine nachträgliche Integration höher ist?

Kommt leider immer drauf an. Grundsätzlich sehe ich natürlich schon einen Vorteil darin, wenn man das Ganze von vornherein mitberücksichtigt, weil man dadurch natürlich von Anfang an darauf schauen kann, dass die Vorgaben erfüllt werden. Was auch viel leichter ist, wenn man bereits weiß was alles erfüllt sein muss, um ein gewisses Ziel zu erreichen. Wohingegen bei bestehenen Projekten, kann es natürlich sein, dass gewisse Dinge komplett wegfallen oder umgeschrieben werden müssen, weil es so gar nicht machbar ist. Oder es so ein hoher Aufwand ist es umzubauen. Also kann man schon auf alle Fälle sagen, dass eine Berücksichtigung von Anfang an auf alle Fälle hilfreich ist und weniger Aufwand.

Glaubst du grundsätzlich, dass es für dich mehr Aufwand bedeutet barrierefrei zu programmieren? Also wenn du von Anfang an weisst, dass es berücksichtigt werden muss?

Es ist sicherlich möglich ohne großen Mehraufwand. Anforderungen gibt es ja immer in Form von gewissen Spezifikationen und da kann man natürlich schauen, dass Accessibility bereits mitberücksichtigt bzw inkludiert ist, dann gibt es eigentlich quasi keinen Aufwand. Beim Testen ist es natürlich ein zusätzlicher Punkt, der mitberücksichtigt werden muss, aber ein großer Mehraufwand ist es bestimmt nicht. Vor allem nicht, wenn von Anfang an darauf geachtet wird. Wenn quasi jedes Rad im Wagen darauf achtet, dass es passt, dann sehe ich da keinen Mehraufwand.

Inwieweit hast du das Gefühl, dass Barrierefreiheit im Web ein Thema ist? Also das es umgesetzt wird, oder es berücksichtigt wird?

Früher habe ich das ganz wenig bis gar nicht mitbekommen, aber mittlerweile tut sich schon einiges - ich sehe hier schon Änderungen. Gerade Google zb hat hier einen großen Fokus auf A11Y, auch bei youtube - da gibt es einen eigenen Channel für Developer. Also ich sehe hier schon einen Wandel. Es wird bewusster, dass viele Menschen ausgeschlossen werden, und im Web wird nun überlegt, wie man dazu kommt, dass diese auch am Leben im Web teilnehmen können.

Und das ist nicht aufgrund deiner geänderten beruflichen Situation, da du jetzt ja in einem Unternehmen bist, in dem Barrierefreiheit einen großen Stellenwert hat?

Nein. Ich habe zB diesen erwähnten Channels auch früher schon angeschaut, aber der Fokus dort lag nie auf Barrierefreiheit, sondern Technologie usw. Aber hier sehe ich auf alle Fälle einen Wandel.

Warum glaubst du, dass das bisher von so vielen Entwicklern "ignoriert" worden ist?

Ich glaube, dass vor allem - man muss natürlich sehen, dass das Hauptziel dieser großen Firmen natürlich auch ist Geld zu verdienen - wenn man sich nun ansieht wie rasant diese Unternehmen gewachsen sind und bei welchen Userzahlen sie stehen, dann muss man sich halt einfach fragen, "Gibt es überhaupt noch mehr Menschen?" Und wahrscheinlich hat denen dann jemand gesagt, dass sie einige ausschließen. Also ich glaube eigentlich nicht, dass das aus gutem Willen von denen forsiert wird, sondern die wollen halt Geld verdienen und haben hier eine Lücke gefunden. Dabei glaube ich gar nicht, dass es primär nur um eingeschränkte Personen geht, sondern man erreicht dadurch ja noch viel mehr - die Seiten sind für ältere Geräte, oder sonstige Einschränkeungen ja auch leichter zu nutzen.

Also du glaubst es geht um die reine Geldgier der Menschen?

Nein, so negativ darf das nicht ausgelegt werden. Es gibt sicherlich auch Menschen, die aus Nächstenliebe handeln. Aber bei den Big Playern glaub ich schon, dass das eine große Rolle spielt. Aber das ist ja eigentlich auch egal, da es ja eh um die Sache geht und die Beweggründe ja eigentlich zweitrangig. Wichtig ist, dass diese Big Player insgesamt ein Umdenken herbeiführen.

Wie stehst du dem Statement, dass Accessibility und Schönheit sich gegenüber stehen? Kann eine barrierefreie Webseite auch schön sein?

Das halt ich ehrlich gesagt für kompletten Schwachsinn. Das halte ich eher für eine Ausrede, um die Defizite wohl auf etwas abzuwälzen. Man kann sicherlich Wege finden, bei dem sich alles vereinbaren lässt und ein Kompromiss zu finden.

Wie verständlich findest du die WCAG Richtlinien?

Ich glaub sie bringen schon das rüber, was sie sollen. Ein Spec ist ja meistens hauptsächlich etwas was Experten nutzen und Experten die drüber schauen. Ein Spec ist eigentlich ja immer sehr technisch, also Otto-Normal-Verbraucher können das vermutlich nicht ganz verstehen oder nachvollziehen. Auch einige Developer haben sicherlich Probleme damit - das kann ich gut nachvollziehen. Aber ich glaube, dass wenn man sich damit beschäftigt und mehr Docs und zusätzliches dazu liest, dann kann man das schon verstehen.

Interviewee B

Was für eine Rolle hast du? Wie würdest du deinen Job beschreiben?

Geschäftsführer und Gründer der Firma X, einer kleinen New Media Agentur, mit Schwerpunkt Accessibility aber nicht ausschließlich - hauptsächlich im Web-Bereich.

Wie lange bist du bereits in der Accessibility Branche tätig?

Seit 2003.

Woher kommen die Beweggründe deiner Kunden, dass sie dich kontaktieren und eine barrierefreie Website wollen?

Mehrere Gründe: öffentliche Institutionen haben den Beweggrund, dass sie rechtlich verpflichtet sind, die müssen es machen. Dann gibt es kleinere Vereine, die eine Förderung bekommen, bsp von einem Ministerium und in diesem Auftrag steht drinnen, dass sie barrierefrei sein müssen. Und dann machen sie es. Und bei anderen Kunden machen die es teilweise aus Image-gründen, und andere glauben, dass sie damit mehr Umsatz machen oder allgmein das Image aufpeppen können.

Siehst du da einen Trend hin zur Barrierefreiheit?

Ja, schon. V.a. seit 1.1.2016 haben wir schon viel mehr Anfragen in die Richtung. Sowohl an barrierefreien Webseiten, als auch Evaluierungen und Beratung.

Führst du bei den Entwicklern mit denen du zusammenarbeitest spezielle Schulungen durch? Oder nimmst du nur solche, bei denen du bereits weisst, dass sie die entsprechenden Qualifikationen haben?

Teilweise.. zum Teil sage ich ihnen schon vornab auf was sie achten müssen. Teilweise bauen wir aber auch die Sachen so zusagen im Prozess mit ein (nicht nachträglich, sondern fortlaufend)

Wie schaut das aus? Wie kann ich mir diesen Prozess vorstellen?

Zum Beispiel hatten wir gerade ein Projekt des ... und da habe ich mit 4 Entwicklern zu tun gehabt. Und ich habe einfach während sie entwickelt haben teilweise drüber geschaut und geschaut was noch fehlt oder was anders gemacht werden sollte und hab denen das dann gesagt, bzw. teilweise auch selber gemacht.

Das heisst, du machst dann also fast kontinuierliches Testing während dem gesammten Entwicklungsprozesses?

Das kommt darauf an, welche System eingesetzt werden. Manche Systeme sind eh bereits recht gut, die bringen die Basis mit und dann muss man nur darauf achten, dass gewisse Zusatzpunkte eingebaut werden. Das ist dann oftmals zu mühsam alles dem Entwickler zu erklären, dann bau ich das einfach selber schnell ein.

Was glaubst du, warum Barrierefreiheit gerade bei Entwicklern so wenig berücksichtigt wird?

Ich glaube einfach, dass die Entwickler teilweise eh schon mehr ihre Schwerpunkte haben, also die einen beschäftigen sich intensiv mit JavaScript, die anderen mehr mit Backend Entwicklung, Datenbanken oder Frontendentwicklung. Und die haben dann jeweils ihre Schwerpunkte. Also mit denen wo ich zusammenarbeite, die haben eigentlich alle ihre Schwerpunkte. Und sie kennen sich alle ein bisschen aus mit der Barrierefreiheit - sie wissen alle um was es geht, aber dass sie so richtig Experten in dem Gebiet werden, da fehlt ihnen der Anspruch und sie sehen da eigentlich auch gar keine Notwendigkeit dafür für ihre tägliche Arbeit.

Das heisst, bei wem würdest du die Verantwortung sehen für Barrierefreiheit, also offenslichtlich nicht beim Entwickler. Sondern eher beim Projektleiter?

Schon beim Projektleiter, ja. Also bei einem größeren Projekt muss auf alle Fälle der Auftraggeber dahinter sein - der muss klarerweise sagen, dass er es gerne barrierefrei hätte. Und dann gibt es einen PM, der das Projekt betreut und dieser ist dafür verantwortlich, dass es barrierefrei gemacht wird. Der muss seinen Entwicklern dann sagen.. das müssen wir so und so machen. Ich glaube nicht, dass die Entwickler selber da großen Wert darauf legen können. Es sei den man sagt es ihnen - das muss aber vom PM ausgehen - also von den Anforderungen.

Glaubst du man kann da entsprechende Spezifikationen bereitstellen. Also bereits in den Spezifikationen auf Barrierefreiheit Rücksicht nehmen?

Sicherlich. Das muss bereits in ein Pflichtenheft mitaufgenommen werden - das hilft auch den Entwicklern. Das machen wir auch so.

Und wie beschreibst du das dann? Sagst du einfach nur es muss barrierefrei sein, oder machst du bereits konkrete Umsetzungsvorschläge?

Da geben wir schon konkrete Anhaltspunkte. Wenn man beispielsweise ein Flyout, also eine Flyout-Navigation macht, die nur mit Maus bedienbar ist, da sagen wir dann, dass das Flyout schon mit Maus, aber auch mit Tastatur bedinbar sein muss. Dann wissen die Entwickler, wenn sie sich ein bisschen damit auskennen, was zu tun ist. Dann wissen die genau, ok, das js muss so und so sein und das stylesheet so und so aufbauen, damit das funktioniert.

Glaubst du das Barrierefreiheit ein großer Mehraufwand ist? Wenn man es von Anfang an berücksichtigt, vs wenn man es im Nachhinein integriert?

Ja das sicher, wenn man die Seite fertig hat und man dann darauf kommt, ups - jetzt haben wir die Barrierefreiheit vergessen und das müssen wir eigentlich auch noch tun. Dann ist das sicher ein höherer Aufwand, als wenn man das gleich von Anfang an berücksichtigt. Wie ich das vorher gesagt hab, wenn der PM das schon entsprechend in der Spezifikation mitberücksichtigt. Aber auch wenn die Entwickler gebrieft werden, dass ist sicher ein Mehraufwand. Das hängt aber auch ein wenig vom Projekt ab. Also, ob es jetzt 10% oder 20% sind, das kommt drauf an. Bei manchen Projekten ist es wahrscheinlich wirklich ein enormer Mehraufwand. Ich hab da zb einen anderen Kunden im öffentlichen Bereich und die machen JavaSript lastige Systeme und die haben dann lauter strukturelle Probleme, wenn sie es nicht gleich berücksichtigen. Z.B. wie die Seitenstruktur aufgebaut ist, ob es eine gescheite Navigation gibt, ob es andere Wege der Navigation gibt - eine Sitemap oder eine Suche. Und wenn sie das nicht gleich von Anfang an mitbedenken und die Systeme sind so gebaut, das eine Suche nicht möglich ist oder denkbar ist und die dann nachträglich einbauen müssen, dann ist es schon ein ziemlicher

Mehraufwand. Es gibt aber andere Seiten, wie zb kleine Wordpress Projekte da kann man relativ schnell mit JavaScript und ein paar Kleinigkeiten von Aussen noch die gröbsten Sachen abfangen. Da ist das dann nicht so viel Aufwand. Es hängt also davon ab, was es für ein Projekt ist. Aber als Mehraufwand würde ich es auf alle Fälle sehen. Auch wenn man es von Anfang an berücksichtigt. Das ist ganz klar ein Mehraufwand, gegenüber wenn man Barrierefreiheit komplett aussen vor lässt. Aber sicherlich wesentlich weniger als nachträglich.

Designer stehen sehr of Barrierefreiheit kritisch gegenüber und meinen, dass Barrierefreiheit zu Lasten des Designs gehen. Können barrierefreie Seiten auch schön sein?

Ja, das sind Designer, die sich halt nicht wirklich damit beschäftigen und halt auch nicht ausgekennen. Viele glauben halt noch - es gibt noch dieses alte Vorurteil, dass barrierefreie Seiten schiach sind, das kommt halt noch von 2003/2004 wo das angefangen hat. Als viele noch eine alternative w3c Version gemacht haben, die hat dann auch oft so geheissen - das ist natürlich völliger Blödsinn, der Name. Und diese w3c-Version war eigentlich nur eine Textversion der Webseite ohne jegliche Effekte oder dergeleichen und daher kommt das. Aber grundsätzlich, wir machen Seiten, die ... wir achten da am Anfang mit den Designern eigentlich gar nicht so drauf, sondern wir lassen sie einfach mal machen. Und das wichtigste was sie berücksichtigen müssen sind die Kontraste. Weil da können sie halt doch schon am Anfang recht viel verbocken. Aber ansonsten kenne ich jetzt kein Design, das nicht barrierefrei umsetzbar wäre, wenn die Kontraste passen.

Und die Kontrastvorgaben gibst du dem Designer aber von vornherein mit auf den Weg?

Ja, wir sagen ihnen halt, dass sie auf das achten müssen und es hier ein gewisses Verhältnis von Hintergrund und Schrift geben muss. Die Designer, mit denen wir zusammenarbeiten, die barrierefreie Seiten machen müssen, die wissen das aber eh schon.

Schaust du bei deinen anderen Projekten, wo Barrrierefreiheit keine Rolle spiel, zumindest aus Sicht des Kunden, gar nicht darauf?

Es kommt drauf an. Also grundsätzlich ist ja Barrierefreiheit sehr stark verwoben mit der Suchmaschinenoptimierung und mit der Usability - wie meine DA gezeigt hat, hängt die Barrierefreiheit auch mit der Usability zusammen, das heisst, eine Seite, die barrierefrei ist, ist auch automatisch besser nutzbar. Und eine Seite die barrierefrei ist, ist auch automatisch suchmaschinenfreundlicher. Das heisst, es gibt einfach gewisse Sachen, die du sowieso machst wenn du eine Seite baust. Weil SEO wollen ja eh alle, Usability ist auch gerade im Trend und wollen alle, sodass die Seite halt funktioniert und nicht zu Unzufriedenheit führt. Und dadurch bauen wir schon gewisse Sachen grundsätzlich ein. Aber so spezielle Sachen wie Skiplinks zb - so Abkürzungssprunglinks - würde ich jetzt nicht einfach so einbauen, wenn der Kunde es nicht so explizit wünscht. Also er nicht explizit eine Barrierefreie Webseite haben will.

Also du siehst den Trend zur Usability stärker als wie zur Barrierefreiheit?

Nicht unbedingt, nein. Es kommt darauf an, woher die Kunden kommen, bzw. was es für ein Projekt ist. Also derzeit zb haben wir wesentlich mehr Anfragen in Richtung

Barrierefreiheit. Aber wir sind natürlich auch dafür bekannt. Andere Firmen sagen vielleicht schon wir machen mehr Usability.

Wie hast du das Gefühl, ist Barrierefreiheit grundsächlich im Web ein Thema?

Ganz schwierig. Im öffentlichen Bereich glaube ich, dass wir schon bei 80% sind, bei den größeren Unternehmen vielleicht so 40% und bei den Kleinen vielleicht so 5-10%.

Woran glaubst du liegt der Trend zur Barrierefreit? Wird der überhaupt mehr?

Wird viel mehr, ja. Ich glaube das hängt hauptsächlich an den gesetzlichen Vorgaben.

Was verwendest du für Accessibility Methodiken. Bzw. gibt es welche? Was wendest du an und wie?

Naja, es gibt hier natürlich Methodiken. Es gibt zb die WCAG, das ist die Hauptgrundlage, die wir verwenden. Derzeit gibt es aber gerade einen spannenden Prozess - das hast du wahrscheinlich eh mitbekommen - den Zertifizierungsprozess. Da sind wir gerade dabei - mehrere Accessibilityexperten - eine gemeinsame Prüfmethode zu entwickeln für einen Zertifizierungsprozess. Wir überlegen uns wirklich für jeden von diesen WCAG Checkpunkten wie man den überprüfen kann, was gilt, was lassen wir durch und was gilt nicht. Wir lehnen uns da derzeit sehr stark auch an die BITV an, also die deutschen Richtlinien, weil die haben ja die Prüfschritte ziemlich ausführlich formuliert. Und wir schauen und diese Prüfschritte von der bitv an und schauen, ob das noch passt - ist ja auch schon ein paar Jahre alt - also wir schauen, ob es noch zeitgemäß ist oder ob wir dies anders formulieren würden. Aber ansonsten, für unserer Testberichte gelten ganz streng die WCAG Richtlinien.

Wie verständlich findest du die WCAG Richtlinien?

Ja, also wenn man sich nicht sehr damit beschäftigt sind sie unverständlich. Du kannst nichts fragen, keine Kommentare schreiben usw. Sie sind auch nur in Englisch erklärt, was für manche deutschen Entwickler und Seitenbetreiber vielleicht eine weitere Hürde ist. Aber andererseits ist es vielleicht eh nicht schlecht. Das es halt Experten gibt und geben muss, die sich damit beschäftigen. Weil das ist ja kein einfaches Thema - man muss eigentlich schon ziemlich viele Sachen beherrschen, um das zu verstehen. Man muss quasi Frontendentwickler sein, du musst dich auch mit Backendtools auskennen, wie kann man uas einem Alt-Text ein Pflichtfeld machen usw. und du musst auch wissen, wie die assistiven Technologien angewendet werden. Weil es ja auch wichtig ist, dass man mit denen prüft. Das ist insgesamt nicht so einfach. Und ich glaube nicht, dass es so einfach massentauglich wird. Das jemand sagt, ok ich schau mir das kurz an und dann überprüfe ich diese Seite. Es ist ein sehr komplexes Thema und ich glaube da wird es immer Experten brauchen, die beraten oder auch testen.

Aber für Entwickler wäre es schon besser, wenn es einfacher verständlich wäre - dann wäre vielleicht auch die Diskrepanz nicht so hoch.

Ja, das ist halt wie gesagt so eine Geschichte. Wenn man sich zb mit jQuery beschäftigt, also mit JavaScript, da gibt es auch einen gewissen Lernaufwand. Das ist ja auch nicht einfach alles so ganz einfach in Tutorials drinnen, sondern man muss sich schon auch damit beschäftigen. Sicher, man könnte sicherlich einiges machen, also besser machen bei den WCAG. Wir hatten ja auch im Verein Accessible Media bereits einmal überlegt, dass wir eine eigene Seite machen, quasi eine WCAG Erklärungspage auf deutsch, als Erklärungsseite. Mit best practice Beispielen und Code Schnipsel usw. ist aber leider nichts daraus geworden, weil wir die Förderung dafür nicht bekommen haben. Aber aufgeschoben ist nicht aufgehoben.

In der wissenschaftlichen Literatur stehen die WCAG Richtlinien ein bisschen in der Kritik - die sagen, das ist teilweise veraltet und berücksichtigt einiges nicht mit und es werden Dinge gefordert die eigentlich nicht mehr so relevant sind.

Ja, die Browser und auch die assistiven Technologien entwickeln sich weiter, daher stimmt es sicherlich, dass bei den WCAG nun ein paar Punkte dabei sind, die nicht mehr so relevant sind. Ein Beispiel dafür ist die Vergrößerung des Textes auf einer Seite. Die Browser sind hier alle schon so gut, dass man mit STRG + + oder mit den Scrollrad, die Seite wirklich angenehm vergrößern kann. Das können mittlerweile auch alle major Browser. Dieser Punkt ist also wahrscheinlich nicht mehr so relevant. Was dann aber schon doch wieder ab und zu passiert, ist, dass bei der Vergrößerung auch durch diesen Zoom-Faktor zu schnell in den mobilen Modus springt und dadurch dann die Dinge nicht ganz richtig funktionieren mit diesem vergrößerten mobilen Modus. Das ist schon ein Problem, dass wir häufig sehen. Von daher hat der Punkt jetzt schon noch ein bisschen seine Berechtigung. Aber das ist ein Bsp. Ein anderes Bsp wäre der Link, also dass ein Link gut erklärt werden muss. Und da gibt es die leeren links, aber wenn man dafür dann zb einen Link-Titel einführt, wird das von den Screenreadern perfekt vorgelesen. Wobei das streng genommen nicht WCAG konform ist. Aber im großen und ganzen enwickeln ja die W3C die WCAG noch weiter und die versuchen da schon ein bisschen aktuell zu sein. Was aktuell zb noch drinnen steht - alles was Flash betrifft - das ist völlig unsinnig, das braucht man wahrscheinlich wirklich nicht mehr. Also meiner Meinung nach ist es jetzt nicht so heillos veraltet, wie es vielleicht in der Literatur häufig dargestellt wird. Sondern es sind ein paar Punkte die man diskutieren muss.

Wie gehst du mit Redationen um? Machst du hier spezielle Schulungen?

Kommt auf den Kunden an. Bei ... haben wir einige Workshops gemacht, als Redaktions-Briefing. Wie sie sozusagen den Inhalt - wir nennen das Redaktionelle Barrierefreiheit - wie sie also den Inhalt barrierefrei aufbereiten können. Das ist schon wichtig. Gerade in größeren Redaktionen, die den Anspruch haben, dass die Seite barrierefrei sein muss. Die müssen gebrieft und eingeschult werden.

Interviewee F

Was hast du für eine Rolle?

Frontend Developer

Wie lange arbeitest du bereits als Dev, bzw. mit Fokus auf Barrierefreiheit?

6 Jahre, seit Jän 2015 barrierefrei

Und hast du da spezielle Schulungen oder Trainings gemacht?

Trainings in Form von youtube-Videos oder online Artikel, aber keine dezidierte Schulung, keine Zertifikate.

Wie viel Zeit hast du damit verbracht um dich zu einem barrierefrei-Entwickler zu entwickeln?

Das geht Hand-in-Hand. Wenn du guten Code hast und eine schöne HTML Struktur schreibst und alles so machst wie es gehört, dann ist das Ganze schon viel mehr accessible, als wenn du einfach nur so Code schreibst. Die Frage ist, wo fängt Accessibility an und wo hört HTML auf? Also wie gesagt, wenn ich form-element dort einbinde, wo auch form-element sein sollen, dann hat das eigentlich nichts mit Accessibility zu tun, sondern ist einfach nur standard-konforme programmierung.

Und warum sind dann doch so viele Seiten nicht barrierefrei, bzw. zb nicht keyboard tauglich?

Weil Entwickler halt doch die falschen Elemente benutzen, weil sie nicht wissen, dass es einfach gehen würde. Weil sie nicht wissen, dass es Elemente geben würde, die das einfach unterstützen. Ich hab früher zb auch divs verwendet statt dem button element. Das macht man halt so. Das ist einfach - also man verwendet halt die einfachere Variante. Aber wenn man sich eine gewisse Zeit damit beschäftigt, dann kennt man auch die richtigen Ressourcen.

Also du glaubst es liegt an der Unwissenheit der Entwickler?

Ja. Also ich sehe keinen Vorteil darin warum man divs einem button bevorzugen sollte, ausser, dass man halt die Hintergründe warum man die korrekten Elemente verwenden sollte nicht kennt.

Wie könnte man dem entgegenwirken? Glaubst du es gibt hier überhaupt eine Möglichkeit dem entgegenzuwirken?

Ja, ich finde es passiert auch derzeit ganz stark mit den ganzen - also Microsoft und Google pushen das derzeit ganz stark. Im Gegensatz zu früher, wo das überhaupt kein Thema war. Z.B. vor drei Jahren gab es kaum jemand der Interesse hatte und nun wird das von den großen Firmen schon ziemlich gepusht.

Kannst du dir vorstellen warum das nun so gepusht wird?

Ich bin nicht sicher - vielleicht weil Machine Learning ein immer größeres Thema wird. Und je barrierefreier Seiten sind, desto leichter ist es für Maschinen den Code zu interpretieren und die Seite zu verstehen bzw. zu verarbeiten. Also die großen sind vermutlich auch daran interessiert die Seiten besser indexieren zu können, was natürlich durch saubere Strukturen und gute Inhalte einfacher möglich ist. Da wirkt Accessibility natürlich unterstützend.

Deinem Gefühl nach - wie weit ist Barrierefreiheit im Web bereits umgesetzt?

Eigentlich unterdruchschnittlich. Selbst Seiten von Google - also irgendwelche Tutorial Seiten von ihnen, sind teilweise nicht durchtabbar, Bilder sind nicht anspringbar. Letztens habe ich auf youtube ein Tutorial gesehen, wo sie für font-icons Werbung machen, also fontawesome, und sowas geht halt eigentlich nicht, weil die nicht accessible sind. Es könnte insgesamt wesentlich besser sein, wenn nicht mal die Seiten von großen Firmen barrierefrei sind.

Kannst du dir vorstellen warum das so ist? Weil du ja auch vorhin selbst gesagt hast, dass es doch immer mehr im Kommen ist. Glaubst du die haben einfach immer noch nicht nachgerüstet?

Ich glaube sie sind interssiert daran, dass andere das machen, aber selber haben sie die Daten ja, also warum sollten sie die eigenen Services unbedingt accessible machen. Nur weil du etwas nach aussen trägst, heisst das ja nicht, dass du auch intern die Struktur so umsetzen kannst.

Aber schön wäre es ja schon, wenn man hinter seinen eigenen Aussagen stehen würde.

Ja, klar wärs schön - aber wer zahlt das?

Also du glaubst, dass es einfach zu teuer ist, Barrierefreiheit umzusetzen?

Um die Firmen entsprechend umzustrukturieren, ja.

Glaubst du grundsätzlich, dass barrierefreie Gestaltung mehr Aufwand ist? Wenn man es von Grund auf macht.

Kommt auf die Entwickler an. Wenn die Entwickler wissen was sie tun, dann nein. Aber wenn die nicht wissen was sie tun, dann auf alle Fälle. Weil du ihnen halt alles sagen muss.

Wie könnnte man diesen Prozess, wie könnte man die Entwickler darauf aufmerksam machen auf was sie schauen müssen? Entsprechende Anforderungen?

Kommt auf die Teamgröße an. Grundsätzlich sind Schulungen schon super, aber bei kleinen Teams halt auch nicht immer so intensiv machbar. Bei großen Firmen könnten Schulungen aber auf alle Fälle hilfreich sein. Weiters hilft es natürlich, wenn die Anforderungsbeschreibungen schon dementsprechend sind. Und man den Entwickler so aufmerksam macht auf semantisch richtige Strukturen. Also konkrete Anforderungen, sodass man weiss wie man es umsetzt und man auch nichts vergisst. Das würde schon helfen.

Wer hat deiner Meinung nach die Verantwortung für Barrierefreiheit? Der Entwickler oder PM?

Ja, der Entwickler legt die Grundlage, getestet muss es dann von jemand anderen werden. Aber es sind schon die Entwickler, die daran denken sollten.

Glaubst du, dass sich gutes (schönes) Design und Accessibility gegenüber stehen? Können barrierefreie Seiten auch schön sein?

Kommt darauf an, wie radikal man ist. Es gibt hier schon Umsetzungen, die sehr stringent sind, und das Ganze dadurch schon unschön werden lassen. Aber einen wirklichen Widerspruch würde ich nicht sehen. Man kann schon auch Dinge schön und gleichzeitig auch barrierefrei machen. Ich glaube sogar, dass wenn Dinge wirklich schön sind, sind sie auch automatisch accessible. Also ich zb würde eine Seite, die mit der Tastatur nicht bedienbar ist, nicht als schön identifizieren.

Wirklich? Ich spreche jetzt von ästhetisch schön, nicht von strukturell schön.

Ja, das ist natürlich sehr subjektiv. Ich achte auf so etwas, aber der Durchschnittsmensch wohl eher nicht. Ja, da haben wir zb Outlines. Die werden sehr gerne weggelassen, weil sie halt nicht schön sind. Aber es gibt Vorschläge dies in Zukunft zu optimieren bei den Browsern - sodass die Outline von den Entwicklern stylebar ist. Aber sonst.. ich glaube es gibt wenig Dinge, die aufgrund von Accessibility nicht schön sein können.

Glaubst du das man dem Problem, dass Webseiten nicht ausreichend barrierefrei umgesetzt sind, irgendwie entgegenwirken kann? Glaubst du es kann einen standardisierten Weg geben, den man vorgibt, und dann sind Seiten barrierefreier?

Kann ich mir jetzt nicht vorstellen. AMP (Accelerated Mobile Pages) von Google sowas könnte dann automatisch accessible sein, weil du immer ein Subset von irgendwelchen Funktionen und JavaScript Funktionen benutzen kannst. Aber sonst.... wie willst du Leute dazu zwingen barrierefrei zu entwickeln? Also, du kannst Gesetze machen und man kann schon von aussen versuchen das zu regulieren, aber ein Toolkit, um Dinge automatisch accessible zu machen, kann ich mir nicht vorstellen.

Wenn man nachträglich barrierefreiheit integriert, ist dies ein recht hoher Mehraufwand gegenüber einer Berücksichtigung von Anbeginn. Wie hoch schätzt du das tatsächlich ein?

Es kommt darauf an, ob das Testing schon passiert ist, also ob man noch mehr Freiheiten beim umbauen hat, oder ob es sich tatsächlich um eine Seite handelt, die so schon in Verwendung ist. Wenn es wirklich eine fix fertige Seite ist, ist es schon kritisch und echt viel Aufwand... weil möglicherweise die gesamte Struktur umgestellt werden muss und nochmal alles auch in Funktionalität getestet werden muss.

Also quasi fast doppelter Aufwand, weil du die gesammte Struktur, das ganze HTML überarbeiten musst.

Ja, mindestens, das doppelte.

Kennst du WCAG? Findest du diese sehr verständlich?

Überhaupt nicht, nein. Ist auch nicht meine favorisierte Stelle, um an Infos zu kommen. Wenn ich etwas brauche/suche, dann sicher nicht von WCAG Spec. Es ist schön, dass es sie gibt, und das man einen Standard festlegt wie bei einem Gesetz. Aber für Developer sind die nicht wirklich brauchbar - höchstens in irgendwelchen Edge-Cases, wo ich sonst nichts dazu finde.

Interviewee A

Was für eine Rolle hast du?

Geschäftsführer, Projektleiter und in der Konzeption von Projekten tätig, sowie Accessibility und auch Usability Consultent.

Wie lange arbeitest du bereits in der A11Y Branche?

2005 im Bereich A11Y, davor auch schon damit auseindandergesetzt aber eher nur am Rande.

Was sind die persönlichen Beweggründe für den Fokus auf Barrierefreiheit?

Weil es für mich ein wichtiges Grundprinzip des Webs ist und insbesondere von inhaltsreichen Webseits und funktionsstarken Webseites (content-rich). Bei meiner Firmengründung wollte ich mich abgrenzen von diesen teils sehr stark verbreiteten Websites, die aus der Werbebranche gekommen sind. Und standardkonforme Webentwicklung war immer ein Grundprinzip, wobei das nicht als Selbstzweck dient, sondern immer den Menschen, die es benutzen auch dienen sollte.

Was glaubst du ist die Motivation oder die Beweggründe deiner Kunden, dass sie sich für eine barrierefreie Webseite interessieren/bestellen?

Die ursprüngliche Motivation ist sicherlich häufig die gesetzlichen Anforderungen bzw. die hoheitliche Anforderung, also bei Websites der öffentlichen Hand - hier ist es schon auch gesetzlich festgelegt, aber dennoch zusätzlich hoheitlich, also von oben herab, sozusagen verordnet. Der Bund hat sich das ja selbst auferlegt. Wobei ich sagen muss, dass eigentlich mir kaum jemand begegnet ist, der das grunsätzlich kritisiert hätte, wenn dann kommt die Kritik eher aus der grafischen Ecke.

Das bringt mich gerade zu einem Punkt: Designer stehen Barrierefreiheit oft kritisch gegenüber und sagen, dass diese nicht schön sind. Können barrierefreie Webseiten auch schön sein?

Ich denke, ein Webdesigner, der keine Ahnung hat von sagen wir mal nur UX, nehmen wir das Thema A11Y raus, das ist für mich kein guter Webdesigner, weil dass das nur schön ausschaut, das hat sehr wenig Mehrweit. Also niemand geht auf eine Website und denkt sich einfach nur, wow - das schaut geil aus, ist dann happy und benutzt sie vielleicht wieder. Man ist vielleicht einmal beeindruckt, aber die Leute, die dann wieder kommen, wenn das nicht die entsprechende Usability hat oder den Nutzen für den User.

Inwieweit glaubst du das Barrierefreiheit im Web heute tatsächlich umgesetzt ist? Hast du ein Gefühl dafür was für eine Rolle A11Y tatsächlich spielt?

Es ist schon ein Thema - es hat so Zyklen gegeben. ZB vor 12 Jahren als wir begonnen haben, war es ein Thema, zwar nicht sehr gross, ist dann aber gewachsen. Dann ist dieses Buzzword Web 2.0 aufgekommen, wo es sehr viel auch um Content Sharing und neue dynamische Umsetzungen gegangen ist. Damals ist dann noch die schlechte Browserqualität dazu gekommen. Und da hat es dann eigentlich einen Backlash für die

A11Y gegeben. Da war zwar das Bewusstsein schon da, aber eigentlich war es den Leuten damals wichtiger coole Dinge umzusetzen - den neuesten heissen Scheiss und das war dann eigentlich ein ziemlicher Rückschlag in der Praxis. Und vor einigen Jahren ist dann aber die standardkonforme Webentwicklung eigentlich schon ein ethischer Grundsatz der Frontend-Devs geworden. Und jetzt gibt es wieder ein wenig einen Rückschlag, und der kommt von diesen großen Frameworks, wie Angular zum Beispiel. Wo jetzt keine echten Frontend Developer mehr Frontends bauen, sondern halt einfach Leute, die glauben, sie sind coole Developer und haben mit Web aber eigentlich wenig zu tun, sondern bauen einfach nur Webapplikationen und Websites, die eher wieder sehr grottig sind.

Grottig im Sinne der Accessibility?

Ja, und auch im Sinne der Standardkonformität, also semantische Grundstrukturen sind denen völlig fremd und man kann heute ja auch alles mit JavaScript und div und span Elementen machen und so wird das auch gemacht. Und auf dem aufbauend dann Accessibility umzusetzen ist total schwierig.

Du hast gesagt coole Websites haben der Barrierefreiheit entgegengewirkt? Das heisst, eine Website kann nicht cool und barrierefrei zugleich sein?

Das war eher ironisch gemeint. Es war damals sicher schwieriger gewisse dynamische Features und Effekte auch im Sinne der Barrierefreiheit umzusetzen. Aber inzwischen ist das alles sehr gut möglich und wir verdanken das sicher zu einem großen Teil den heutigen Browser Umsetzungen.

Wie kann man deiner Meinung nach Accessibility im Projekt positiv beeinflussen? Was sind die Maßnahmen, die du da in deinen Projekten setzt? Wo im Projektzyklus fängt Accessibility an und wo hört sie auf?

Also, sie muss sehr früh anfangen, in der Konzeption und eigentlich im Auftrag. Also es braucht von Anfang an ein klares Commitment und das sollte nicht in Frage gestellt werden, weil wenn das, gerade bei größeren und komplexeren Projekten irgendwie in Frage gestellt wird oder belächelt wird bzw. ständig zurückgestellt wird, dann wird es ungleich viel aufwendiger, wenn man dann tatsächlich barrierefrei sein will. Also es geht einfach nicht, dass man dann am Schluss des Projektes hergeht und sagt, jetzt schauen wir mal ob wir eh gut sind. Das ist schade ums Geld. Man kann hier nämlich sehr schnell an Niveau verlieren, wenn zb die Redaktion da nicht mitspielt. Wobei das Problem, gerade bei komplexeren Projekten, ist, dass hier Halbwissen über Accessibility nicht ausreicht. Da braucht es schon jemanden, der wirklich weiss wovon er spricht. Und ich würde vielen WebDevs oder manchen Agenturen auch nicht böse Absicht unterstellen, aber es braucht einfach schon einiges an Erfahrung um zu wissen was funktioniert und was nicht. Das braucht halt kontinuierlichen Know-How Aufbau und testen...

Wer, also welche Rolle, sollte innerhalb eines Projektes die Verantwortung für die Barrierefreiheit übernehmen?

Das hängt von der Komplexität ab. Das Management muss das Commitment von Anfang an vermitteln, sonst ist es schwierig, bzw. teurer auf ein gewisses Niveau zu kommen.

Die Person, die das konzipiert oder, sagen wir so... die die das konzipiert und designed... wenn diese Person eine Ahnung von Accessibility hat, ist das schon gut, aber das muss nicht unbedingt sein. Es geht auch, wenn hier jemand beratend begleitet, auch mit eingeschränktem Aufwand. Schwieriger ist es, wenn die Frontend Developer nicht über das entsprechende KnowHow verfügen. Zumindest grundlegend. Wenn das von Extern kommen muss, dann wird es schon sehr schwierig, weil dann dreht man sich schon recht lange im Kreis. Was Backend betrifft, sehe ich normalerweise keine großen Probleme, ausser die bauen die HTML-Strukturen, dann gilt das gleiche wie bei Frontendern. CMS Systeme sind inzwischen schon die meisten eh schon ganz gut ausgestattet. Also, dass sie sich nicht ins Frontend einmischen und das das Medienmgmt ermöglicht, dass man hier Bilder entsprechend einfach und ordentlich hochlädt. Oder Bilder und andere Medieninhalte einfach mit alternativ Texten versehen kann.

Führst du da in deinem Unternehmen für die Entwickler spezielle Schulungen durch? Oder wie stellst du sicher, dass die Entwickler dieses Basis-KnowHow bekommen?

Im Idealfall sollte es schon internen Informations- und Wissenstransfer geben, von den Leuten, die bereits Erfahrung damit haben, zb Schulungen oder gemeinsame Tests. Eine sehr gute Möglichkeit ist es auch, wenn die Entwickler bei User-Tests dabei sind, um die Entwickler zu motivieren und ihnen einen besseren Einblick geben zu können. Meiner Erfahrung nach lassen sich auch eigentlich Entwickler schon sehr gut für die Accessibility motivieren. Das hat mich in der Anfangszeit, als ich Schulungen durchgeführt habe, schon sehr überrascht. Die Entwickler sind die, die am ehesten aufspringen, wenn es darum geht gute barrierefreie Umsetzungen zu machen. Die Redaktionen sind hier meistens eher etwas zurückhaltender. Also ja, interner KnowHow Transfer und Aufbau - ich denke mir es ist halt auch wichtig, dass so wie bei der internen Qualitätskontrolle, dass immer jemand anderer der mindestens das selbe Fachverständnis hat drüber schaut, auch im Hinblick auf Barrierefreiheit. Es passieren hier einfach schnell mal Fehler und Schlampigkeiten.

Du hast vorher Mehraufwand angesprochen - kann man das tatsächlich irgendwie in Zahlen ausdrücken? Was schätzt du was für ein Mehraufwand es ist die Barrierefreiheit grundsätzlich zu integrieren, aber auch erst im Nachhinein im Vergleich dazu.

Einfache Websites, da ist der Mehraufwand für Barrierefreiheit realtiv gering, solang es von Anfang an mitberücksichtigt wird. Schwieriger wird es dann, wenn es um Applikationen geht, die ein bisschen komplexer sind. Und ich denke hier auch an Inhalte, die sich dynamisch aktualisieren zb, oder auch das Thema Lightboxes. Es gibt barrierefreie Lightbox Umsetzungen, oder auch auto-suggest oder so, aber wenn man hier was eigenes entwickeln möchte oder muss, dann wird es schnell mal ein bisschen komplizierter. Vor allem auch der Testaufwand erhöht sich da auch drastisch. Und wenn dann diese Dinge nicht von Anfang an berücksichtigt wurden, dann kann es schon wirklich sehr teuer werden. Wenn man zb HTML Strukturen umbauen muss, wenn dafür in JavaScript events oder dynamische js-Interventionen im Code gibt, dann kann das wirklich sehr aufwändig werden. Was ich auch erlebt hat, ist zb wenn mehrere Entwickler an einem Projekt arbeiten und sie keine Konventionen festgelegt haben, was die Struktur, die

HTML Struktur, betrifft und damit ist dann vielleicht das look-and-feel für Sehende gewährleistet, wobei auch das oftmals nicht, sondern nur das look. Aber gerade das look and feel für behinderte Menschen, die assistive Technologien, Screen Reader verwenden, das ist dann in jeder View anders und eigentlich so nicht tragbar.

Wie kann während des Projektes sichergestellt werden, das alle Accessibility Vorgaben erfüllt sind? Prüfst du auch schon während der Umsetzung oder gibt es erst am Ende einen Test?

Schwer zu sagen. Das hängt von der Größe und Komplexität des Projektes ab. Aber gerade bei komplexeren Projekten ist es schon wichtig, dass frühzeitig immer diese QA-Schleifen gezogen werden, um daraus zu lernen und den weiteren Entwicklungen Inputs zu geben und einfach, um Fehler zu vermeiden.

Und machst du da bereits gewisse Spezifikationen für jegliche Stakeholder, um gewisse Vorgaben zu machen?

Spezifikationen ist ein großes Wort. Es macht Sinn und ist wichtig, wenn beim Design Briefing das Thema angesprochen wird, wenn die Designer nicht A11Y-affin sind, muss man denen gewisse Kriterien kommunizieren. Das ist aber eh überschaubar. Mit den FE-Devs macht es schon Sinn eine, ich nenn das Semantische Landkarte, vorab zu zeichnen, und man anhand von Screen Designs oder Wireframes her geht und die Semantische Struktur definiert - Überschriftenstruktur, welche Elemente verwendet werden und gewisse Sachen, die immer wieder falsch umgesetzt werden gezielt anspricht, zb Tastaturbedienbarkeit. Es gibt hier klassische Elemente, wo es einfach Fallen gibt - lightboxes, datepicker, gewisse Widgets aus Bibliotheken - das man hier die richtigen Entscheidungen trifft, fokus und hover states, dass diese am Anfang definiert werden. Ein Punkt über den auch großes Unwissen herrscht ist die Unterscheidung wann Links und wann Buttons eingesetzt werden sollen.

Vielen fehlt das Grundverständins - was Barrierefreiheit im Web eigentlich ist und wie man das umsetzt usw. Hast du hier irgendwelche Tipps, wie du einem 0815 Website Menschen beibringst, wie wichtig das Thema ist, das es nicht viel Mehraufwand ist und trotzdem viel Nutzen mit sich bringt?

Das Beste Argument ist immer, wenn die Leute unmittelbar erkennen können, wie manche Leute einen Nutzen daraus ziehen. Also wenn sie zb jemandem zuschauen können. Blinde Personen, die Websites benutzen - und wenn es auch nur ein Video ist, finden sie das trotzdem immer sehr inspirierend. Ist meine Erfahrung. Man muss halt ein bisschen aufpassen, dass das nicht auf Blinde und Screen Reader alleine reduziert wird, wobei da viele Anforderungen damit für andere Gruppen auch mitberücksichtigt werden, aber es ist halt nicht das Einzige.

Ich habe festgestellt, dass in der wissenschaftlichen Literatur teilweise die WCAG Richtlinien kritisiert werden. Das sie veraltet sind, aber auch teilweise Menschen ausgrenzen. Was meinst du dazu?

Grundsätzlich sehe ich das ähnlich. Das Web hat sich seit WCAG 2.0 in Kraft getreten ist natürlich massiv verändert. Die Browserlandschaft hat sich verändert und die mobile Nutzung ist dazu gekommen. Früher war halt alles sehr auf klassische Desktop Browser ausgerichtet und viele klassische Anforderungen wie zb die Skalierbarkeit der Schriftgröße, war früher ein sehr wichtiges Kriterium und hat sicher an Bedeutung verloren, weil die Browser mittlerweile sehr gut stufenlos zoomen können. Auch wenn die ursprünglichen Anforderungen durchaus ihre Berechtigung haben oder hatten, aber halt vielleicht nicht mehr in dem Ausmaß. Und in manchen Punkten können Browser oder Screen Reader halt dem User helfen, auch wenn die Website schlecht ist. Was aber schon sehr wichtig ist, sind so Vorgaben, die mit WAI-ARIA hinzugekommen sind. Da sind die best practices, das ist ein extrem wichtiges Dokument, weil es vorgibt, wie gewisse Dinge funktionieren sollen, damit sie gut benutzbar sind. Und hier eine Einheitlichkeit, wie man die benutzt, das ist total wichtig, weil die Leute sich hier auf unterschiedlichsten Websites bewegen und sich zurecht finden können. Genauso wie sich eine gewisse Form, oder ein gewisses Aussehen einer Navigation etabliert hat im Laufe der Jahre, sollte es diese Patterns auch in zb in der Tastaturbenutzbarkeit geben. Und das wird in Zukunft sicher noch mehr. Grundsätzlich würde ich die WCAG nicht in Frage stellen. Das war bei der WCAG 1 viel ärger, weil die war am Schluss wirklich nicht mehr ok. Und WCAG 2.1 wird in einigen kleineren Punkten nachjustieren wie zb auch mobile Accessibility, hier stärker berücksichtigen.

Interviewee D

Was für eine Rolle hast du?

Leitung Portalmanagement

Wie lange arbeitest du bereits in der Accessibility Branche?

Meine Tätigkeit führe ich seit rund 17 Jahren aus, Accessibility war von Anfang an ein Thema.

Was sind die persönlichen Beweggründe? Warum hast du dich für einen Fokus auf Barrierefreiheit entschieden?

Bei Diensteintritt wurden wir von einem Kollegen sensibilisiert, der von Anfang an auf offene Ohren gestoßen ist. Allen schien die Barrierefreiheit eine logische Grundanforderung für ein öffentliches Portal. Später kam dann die rechtliche Verbindlichkeit dazu.

Führst du in deinem Unternehmen spezielle Schulungen durch? Wie gewährleistest du, dass jeder Stakeholder Barrierefreiheit berücksichtigt?

Die Grundzüge, vor allem im redaktionellen Bereich, sind allen Team-MitarbeiterInnen bekannt: LINK zu einer internen Seite. Je nach Aufgabengebiet ist die Expertise unterschiedlich bzw. auf Detailaspekte beschränkt. Etwa für PDF oder UT/Transkription. Im technischen Bereich – für die Prüfung von neuen Anwendungen – greifen wir auch auf externe Hilfe zurück. Die Barrierefreiheit ist bei uns mit Verweis auf die Sinnhaftigkeit und rechtliche Verbindlichkeit im Styleguide verankert. Der hat Erlass-Charakter und ist grundsätzlich zu berücksichtigen.

Wie lange dauert es, um Mitarbeiter entsprechend anzulernen.

Im redaktionellen Bereich eher weniger lang, ist Teil der Schulung. Das ausreichende technische Wissen erfordert mE. mehrjährige Erfahrung.

Inwieweit ist Barrierefreiheit im Web ein Thema? Inwieweit sind Maßnahmen zur Barrierefreiheit im Web umgesetzt? Wie schaut es mit Usability aus?

Ein Thema für alle die müssen, sonst eher sehr wenig, leider. Die rechtlichen Vorgaben haben aber schon sehr viel zur Sensibilisierung beigetragen. Was die Agenturen/IT Dienstleister betrifft, gilt mE.: entweder sie haben es zum Teil der trademark gemacht oder es wird kaum von selbst berücksichtigt.

Was sind deiner Meinung nach die Gründe, warum so wenig auf Barrierefreiheit geachtet wird? Warum glaubst du, ist es so wenig umgesetzt?

Die Usability oder jetzt auch user experience nimmt an Bedeutung zu bzw. wird sie ihr immer mehr zugestanden. So hat zB die größte Firma in Wien mW mittlerweile fast 100 Leute, vor 10 Jahren warens sicher nicht mal die Hälfte.

Wie kann man den Problemen entgegenwirken? Kann es einen standardisierten Weg geben, in dem Barrierefreiheit kein Zusatzaufwand bedeutet, sondern "ganz normal" ist? Wie könnte so ein Prozess ausschauen? Wie ist euer internes Vorgehen? Denkst du Barrierefreiheit von Anbegin mit? Wie schaut der Prozess aus?

Wir versuchen sehr viel über "geprüfte Lösungen" zu regeln. zB. über entsprechende Vorlagen/templates. Auftragnehmer unbedingt im Detail darauf hinweisen in Beauftragung!

Wer (welche Rolle) sollte innerhalb eines Projektes die Verantwortung für Barrierfreiheit oder auch Usability haben?

Entwickler bzw. Externe – je nach Ressourcen und Kompetenz, aber auch Konzeptionisten/Redaktion,

Wie kann während des Projektmanagements sichergestellt werden, dass alle Accessibility-Vorgaben erfüllt sind?

Wir machen zB bei größeren Projekten mehrere Tests, zu Beginn wird bereits das Konzept, dann das Design, später die Umsetzung getestet.

Ist Barrierefreiheit (auch Usability) tatsächlich ein so großer Mehraufwand/Kostenfaktor?

Schon – insbesondere bei UR/Transkription oder wenn Vorlagen nicht verwendet werden, a11y nicht Bestandteil von Aufträgen ist usw. Wenn alles gut läuft, machts nicht so viel aus, kann bei großen Projekten fast in den Gesamtkosten untergehen. Gilt nicht für UT/Transkript, das kostet einfach immer.

Wie können entsprechende Spezifikationen von Accessibility Vorgaben aussehen, um diese systematisch verarbeiten zu können?

Wichtig ist, dass man ihnen auch sagt, wie sie e,s machen sollen. Mit dem Fehler-Feedback alleine können die wenigsten was anfangen. Gute Anforderungsspezifikationen von Anfang an für alle sind hier immer ein Vorteil.

Wie verständlich sind die WCAG Richtlinien? Glaubst du Entwickler haben Probleme diese zu verstehen? Hast du dich bereits näher mit ihnen ausweinander gesetzt?

Ja, ich finds sie schon tauglich.

Laut wissenschaftlicher Literatur sollten die WCAG Richtlinien überarbeitet werden, da diese teilweise veraltet sind und auch teilweise für die User irrelevante Kriterien auflisten, die es zu erfüllen gibt.

Weiterentwickeln muss man sie sicherlich, insbesondere wegen Apps und anderen neuen Formaten.