Disability and Cyberspace:

The Politics of Inclusion & Exclusion in Web Development Groups

By

Kevin R. Fodness

A Dissertation Submitted to the Graduate
Faculty of Rensselaer Polytechnic Institute
in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: SCIENCE AND TECHNOLOGY STUDIES

Approved by the
Examining Committee:

_____
Ron Eglash, Dissertation Advisor

_____
Dean Nieusma, Dissertation Co-Advisor

_____
Abby Kinchy, Member

_____
Richard Ladner, Member

_____
Langdon Winner, Member

Rensselaer Polytechnic Institute
Troy, New York

April 2016
(For Graduation May 2016)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I have been passionate about computers and teaching from a very young age. I was fortunate enough to have a computer in my home at a time when most of my peers had only ever used a computer at a library or public school. I learned how to “home-row” type at the age of 12, and by 13 I was examining website source code to teach myself how to write HTML—the underlying language of the web. I made up my mind to attend Rensselaer Polytechnic Institute (RPI) for Information Technology, with the goal of becoming a professional web developer.

I received my undergraduate education from Rensselaer Polytechnic Institute between the years of 2001-2005. During my freshman year, I went through several secondary concentrations that I felt would be useful in what I hoped would be a future career as a web developer. I began by taking electronic art classes, but switched to communications when I found that I was insufficiently creative, and eventually to Science and Technology Studies (STS) in my sophomore year. In my junior year, STS became a dual major, and by my senior year, I had relegated Information Technology to a minor. I had made up my mind to get a PhD in STS and become an academic.

However, I was unable to secure a funded spot in an STS PhD program immediately after graduating with my B.S., and I ended up getting a job as a Web and database systems administrator with a federal government contractor called NetStar-1, and I found myself working on-site at the U.S. Department of Labor for three years.

It was during my time at Labor that I first encountered accessibility laws

and standards. At the federal level, Section 508 of the Rehabilitation Act governs accessibility of information systems, including websites. I worked on projects to integrate new technologies in accessible ways for our end-users, such as providing employees with two-factor authentication using cryptographic keys that required an accessible alternative for blind users that could not read the numbers on the key fob. During that time, I became interested in applying accessibility principles to my own web development projects. I also began to wonder why I had not heard of web accessibility during my undergraduate IT education, or why it was not a popular topic among blogs and websites devoted to web development. As someone that had received extensive training in information technology—specifically relating to web programming—at one of the nation's leading technological institutions, I was shocked that I had never before encountered accessibility standards or best practices. Therefore, when I left Washington, D.C. three years later to come back to RPI for graduate school, I arrived with a topic—and a mission. I decided to combine my academic and professional interests to learn more about how web accessibility is treated and emphasized in educational and professional spaces.

This dissertation explores why so much of the web is inaccessible for disabled people, and the level and type of attention paid to accessibility in education, in technical manuals, in professional conferences, and in web developers' conceptions of themselves as good actors doing good work. I attempt to bring STS theory together with practical solutions to making the web more accessible—drawing on my own personal experiences as a professional

and hobbyist web developer of 20 years, and the experiences of colleagues in my professional network. Hopefully, the recommendations that I lay out in the conclusion will enable web developers to make the Internet more accessible for disabled people.

ACKNOWLEDGMENTS

Producing this dissertation has been a very long and difficult journey. It would not have been possible without the support of my family, my committee, my employer, and my fellow PhD students.

During the course of my graduate studies, my wife Kelsey and I bought a house and had three wonderful children—Rayleigh, Dillon, and Ainsley. My family has been a constant source of support and inspiration as I have balanced the demands of being a husband, a father, a web developer, and a PhD student.

My committee has gone to extraordinary lengths to support me as I finished my final phases of graduate study—helping me to refine my ideas, keep my scope in check, and push me outside of my comfort zone when I was getting too close to the data, the informants, and my own experience. It has been a long eight years, and I am extremely fortunate to have a committee that is so dedicated to my success.

When I was three and a half years into my PhD studies, after finishing my coursework and comprehensive exams, financial circumstances forced me to take a full time job to support my family. Since 2012, I have been working full time as a web developer for Fingerpaint, a small marketing agency in Saratoga Springs, NY. Fingerpaint as an organization, and the friends I have made there, have been incredibly supportive of my PhD studies. If it wasn't for their support, I would not be writing this document. Additionally, the web development experience I have gained while working at Fingerpaint has broadened and

deepened my understanding of the topic of this dissertation, and has allowed me to expand my professional network to reach people that I never would have been able to interview otherwise.

My fellow RPI STS PhD students have helped to keep me grounded, especially when I felt like I was suffering from impostor syndrome, or that I was the only one that was having difficulty with some aspect of PhD student life. I have spent significantly more of my time off campus than most of my colleagues, so being able to stay in touch and participate in discussions on social media and over email has been invaluable.

If it wasn't for the support of my family, my committee, my Fingerpaint coworkers, and my fellow PhD students, I would never have made it this far. Thank you, thank you, thank you.

ABSTRACT

This dissertation attempts to answer the question “how can the online experiences of disabled people in the United States be improved?” To that end, it explores why the web is such an inaccessible place, including the extent to which accessibility is taught in higher education, the extent to which accessibility is valued as a part of good development practice, and what—if any—overlaps exist between making websites accessible and other desirable outcomes. I draw on my own 20-year history as a professional and hobbyist full-stack web developer, as well as a survey of 330 web developers, and 20 semi-structured interviews of web developers, designers, strategists, project managers, entrepreneurs, and user experience researchers that are part of my professional network in order to examine this question from multiple angles and in depth. I examine my informants' responses through the lens of ableism and the social model of disability, but posit that the complexities of modern web development are not so easily captured in either of those theories, and require a more nuanced view. I extend and challenge Helen Kennedy's (2012) *Net Work: Ethics and Values in Web Design* by more deeply addressing the responsive design trends of recent years and demonstrating the differences between web development in the U.K. and web development in the U.S. Finally, I posit that *synergistic enablement* is an example of a utilitarian approach to making the web more accessible—rhetorically and technologically tying accessibility to outcomes that may be more financially or politically desirable within capitalist organizations, such as optimizing websites for search visibility.

1. INTRODUCTION

The Americans with Disabilities Act (ADA) and its various amendments represented significant rights gains for disabled people[1] in the late 20th century. However, as computer-mediated information technology (IT) replaces older technologies, laws meant to protect disabled people from exclusion either no longer apply or are not adequately enforced in digital contexts. There now exists a strange gap between the ease with which digital accommodations can be created—far easier than retrofitting architecture for wheelchairs, for example—and the failure of law and corporate self-regulation to achieve a broadly accessible Internet. I will demonstrate here that the problem is much more complicated than is traditionally understood, and that solutions to the problem will need to come from multiple arenas based on a comprehensive understanding of the factors that contribute to web development teams producing sites with varying levels of accessibility.

Most of the literature on web accessibility falls into one of two categories: a) how-to manuals for web developers that provide practical instruction on how to create accessible sites, which may also include a discussion of legal requirements and risk (Lazar, Goldstein, and Taylor 2015) or b) general arguments made about the need for greater accessibility in society, which may include a passing mention of the need for greater web accessibility (Harper and Yesilada 2008). There are a few notable exceptions that focus specifically on the

1 Following the disability studies literature, I use the term “disabled people” to refer to the political class of people with physical or cognitive impairments that are affected by disability laws.

social problems surrounding web accessibility, which are exemplified by conferences such as Web for All (www.w4a.info). However, these perspectives are minority voices in a literature that is saturated with practical instruction and arguments for greater accessibility generally. I will explore some of the social web accessibility literature thoroughly in Chapter 2.

The literature on web accessibility does not delve into the specific reasons for why the web is such an inaccessible place, despite academic programs teaching students to include diverse user groups in testing procedures, as well as provisions of the ADA that require web development groups to make accessible sites (Patrick 1996; U.S. Department of Justice 2012).

This dissertation will analyze the reasons why web developers in the United States produce sites with varying levels of accessibility by presenting the results of a 330-subject web developer survey and a 22-subject set of interviews of a broad cross-section of members of web development groups, including web developers, web designers, project managers, entrepreneurs, and user experience researchers. I define a “web development group” as a group of people who come together to produce a website—who may or may not all be members of the same company, who have different roles to play within this process, where some or all roles may co-exist within the same individual.

1.1 Experiences of Internet Users with Impairments

Most of the people that I speak to about disability and cyberspace mention

visual impairments[2] first—most notably, blindness. In some cases, visual impairments are the only impairments that they think of as impacting a person's ability to use the web. However, there are a wide range of impairments that affect a person's ability to use a web browser, and only a few of them are visual in nature. Predictably, different impairments result in different usage patterns. The difficulties faced by web users with impairments are often not obvious to the non-impaired, so it is worth spending a moment putting the reader “in the shoes” of a variety of users with different impairments.

Certain impairments can require additional software, called “assistive technology” (AT), in order for the user with that particular impairment to successfully use the web. In particular, blind users face significant challenges while browsing due to their reliance on a “screen reader,” which is a software program that either reads the contents of a website aloud using a voice synthesizer or translates the screen into a refreshable braille display. Blindness is perhaps the easiest impairment for the traditionally-abled to understand conceptually in the context of using a digital interface, because digital interfaces are often very visual in nature. Physical feedback is often either not present at all, or is significantly diminished, when compared to analog interfaces. However, the technologies that websites are built with means that making them accessible for blind users is easier than one might think.

2 Throughout this dissertation, I will be using the term “impairment” to describe physical and cognitive differences from species-typical abilities, whereas nearly all of the policy language and most of the disability studies literature refers to such impairment as “disability.” This distinction follows Michael Oliver's separation of “impairment” as the physical or cognitive condition of the individual being separate from the “disability” that is caused by society's failure to accommodate the impairment (Oliver 2009). For more on this topic, please see Chapter 2.

Typically, blind users will navigate through an interface using a keyboard, and switch between sections of a website using keyboard shortcuts such as arrow keys and the tab key. Screen readers also have shortcuts for navigating to the next link, text input box, or heading. When a section of a website is highlighted after the user focuses on it using the keyboard, the screen reader will vocalize the text. Screen readers require descriptive text to be present in the website code to make sense of visual elements like images, animations, and tables—which is often not available. If the visual element, such as an image, is purely for visual interest, this is not much of a problem—but if the image is a button that says “Add to Cart,” then the blind user will not be able to interact with that particular e-commerce site very effectively, if at all. In fact, Target was sued by the National Federation of the Blind for this exact circumstance in 2006 (Danielsen 2008). Likewise, screen readers have features for making content in tables understandable to a user that cannot see the layout of the table, but if the table is not coded properly by incorporating proper metadata and descriptive text, screen readers will not be able to accurately describe the layout of the table and relationships between data points in cells to the user.

Many websites have extensive navigation menus between the top of the page and the content. Every time a blind user switches to a new page, the screen reader starts over at the top, and has to read through the navigation all over again. Although the user can tab through the navigation to get to the content, this manual process of skipping through menus and other front matter

can make the experience of using a website very unpleasant for a blind user utilizing a screen reader. If the website includes a “skip to content” link at the top, this pain point can be successfully mitigated. As Bigham et al. have demonstrated, blind users utilizing screen readers take significantly longer to perform tasks than sighted users, largely because of these types of interface hurdles (Bigham et al. 2007).

Many users have visual impairment, but are not fully blind. If you are reading this dissertation with the aid of glasses or contact lenses, you are on the spectrum of visual impairment. Users with mild to severe partial visual impairment may have problems with websites that have text that is too small, or is a color that has insufficient contrast with the background color of the website.

Although web browsers have the ability to zoom entire websites and apply user-specified styles to override developer-specified defaults, typically only tech-savvy users are aware that these features exist. There are also third party solutions, such as ZoomText, which are often platform-specific (ZoomText is only available for Windows) and can be quite expensive (as of the time of writing, a license for ZoomText Magnifier/Reader for a single user is $599). Some magnifiers do not include a content wrapping option, which means that the user must scroll through the content both vertically and horizontally, which can be confusing and tedious. It is a significantly better user experience if the website is programmed to have sufficiently large fonts with sufficiently large contrast to be readable by everyone, rather than relying on user overrides.

Another common usability problem is when color is used exclusively to indicate a difference between two elements. For example, if the only visual difference between linked text and paragraph text is color, then a colorblind user may find it difficult or impossible to pick out links from a paragraph of copy. If the links were colored differently as well as underlined, then even if a user is not able to distinguish between the color of the link and the color of the paragraph text, they will still be able to discern the link based on its underlined state.

Deaf users have difficulty interacting with audio content—either as bare audio, or, more commonly, as part of a video. If a transcript or captions are not available, then the deaf user will not be able to consume the content of the audio or video. With the rise of user generated content (UGC) on video hosting sites such as YouTube and Vimeo, the inaccessibility of audio and video content for the deaf community is becoming increasingly problematic. Although tools exist to attach transcripts or captions to video content, few users are aware that these tools exist, and they are rarely used by average users.

Although the <blink> tag has all but disappeared from the modern web, there are still blinking or flashing elements on many websites. Perhaps there is content in a banner ad or a video that blinks or flashes rapidly, or elements of the user interface (UI) that move quickly and appear to flash. If the blinking or flashing occurs within a certain frequency, it can trigger seizures for people with epilepsy.

The way the content itself is written and structured on a site can present

problems for users with dyslexia or cognitive impairments. For these users, website content should be written in plain language, with an easy to read font, in high contrast, at a size that is easily visible. Unfortunately, dyslexia and cognitive impairments are often overlooked more than other types of impairment (Clark 2006).

Interacting with a website can be a highly performative act. Depending on how the website is built, it may be required to exercise a high level of manual dexterity in order to use the interface. For websites that have multi-level dropdown menus, for example, users that are not able to use a mouse—or that have imprecise pointing devices, or that have difficulty with fine motor control—will find it extremely difficult or impossible to use the menu system. If keyboard navigation is not supported as an alternate method of interaction, entire sections of the website may become inaccessible to people with motor function impairments. I will dive deeper into the technologies themselves in Chapter 3 to provide a more robust understanding of some of the challenges and opportunities for making accessible sites.

The general term for the systematic exclusion of people with impairments through the design or implementation of a product or process is called “ableism.” Ableism “reflects a preference for species-typical normative abilities leading to the discrimination against them as 'less able' and/or as 'impaired' disabled people” (Wolbring 2008, 253). The exclusion does not need to be self-reflexively conscious or cognitively intentional in order to qualify as ableism. In many cases,

ableism manifests as a subconscious bias against accessible design and implementation, due to the orientation of the designers and implementers toward what are usually their own typically “able” bodies (referred to in the disability studies literature as “normate”[3]), and failing to consider the needs of potential users that do not share their abilities. My research questions examine this orientation and bias in an attempt to discover how it can be reduced or overcome in order to produce more accessible websites.

This project is not an attempt to assign blame to particular groups with respect to intentionality, but rather to discover and nurture possible solutions—cultural, social, technical—to improve Internet accessibility. This project is primarily an examination of web development groups from a social constructivist[4] perspective using the disability studies philosophy of the social model of disability to determine how “disability” is socially constructed from impairment in a web context. Therefore, the primary research question aims to probe the social construction of websites through an ableist lens by querying the people responsible for conceptualizing and building the website, including web designers (responsible for the look and feel of the site), web developers (responsible for

---

3 “The term *normate* usefully designates the social figure through which people can represent themselves as definitive human beings. Normate, then, is the constructed identity of those who, by way of the bodily configurations and cultural capital they assume, can step into a position of authority and wield the power it grants them. If one attempts to define the normate position by peeling away all the marked traits within the social order at this historical moment, what emerges is a very narrowly defined profile that describes only a minority of actual people” (Garland-Thomson 1997, 8).

4 For those unfamiliar with the term, “social construction,” and in particular, “social construction of technology,” maintains that there are many possible paths that a particular technology could take when being developed. These paths are created by “an alternation of variation and selection” (Pinch and Bijker 1984, 411) where decisions about how to vary the technology and which paths are selected are made by social forces, in contrast to a technologically deterministic viewpoint.

writing the code to implement the design), and the managers that oversee the designers and developers. Subsequent research questions attempt to identify ways in which ableist attitudes and mindsets direct the course of web projects toward inaccessible ends, and ways in which websites could be made more accessible by working within established structures and ways of working.

1.2 Research Questions

There are four major research questions that guide this study:

1. **ROUTINE DESIGN & DEVELOPMENT PRACTICES: What factors (conscious, subconscious, and structural) determine whether accessibility features are included in or excluded from the routine design and development practices of web development groups?**

Website accessibility is different from Langdon Winner's classic example of the politics of technological exclusion. Winner argues that urban planner Robert Moses intentionally erected low bridges on one of the main causeways to Jones Beach in New York City so that public transportation—in this case, buses—would not be able to get from the poor districts to the beach, thereby erecting a physical barrier to access for people of little socioeconomic means that could not afford private automobiles (Winner 1988, 23).

It is easy to place the blame on Moses. But although websites that are inaccessible to people with impairments constitute an exclusionist practice, it is harder to assign blame, even though these digital “low bridges” block access in

the same way that Moses' low bridges blocked access to Jones Beach for users of public transportation. Winner acknowledges that exclusionist practices can be, and often are, unintentional—such as sidewalks without curb cuts creating barriers for people in wheelchairs (Winner 1988, 25)—yet that is not a clear match either, since one could not complain about the lack of curb cuts during the period when curb cuts did not exist.

Web development groups are complicated entities. Not all web development groups are composed of people who were formally trained in web development or design, not all web development groups are part of organizations whose primary goal is developing websites, and there are significant pressures on web development groups to produce substantial amounts of content and functionality on limited budgets with limited personnel. Additionally, a website is often constructed using inputs and approvals from various groups or actors, which could include several freelance individuals or companies, and often includes a web development group that is separate from the client that is contracting for a website to be built. Within creative agencies in particular, actors are typically further segregated into specializations—account executives, project managers, designers, copywriters, content strategists, front-end developers, back-end developers, and managers, for example.

This dissertation attempts to identify common factors that shape and influence individuals within web development groups, and how those individuals shape group dynamics and outcomes, during the normal course of website

design and development. While there exists a literature on the organization and structure of development groups, and there exists a literature on web accessibility, there is little linking the two. In fact, much of the literature about development groups focuses on the production of the digital artifact (adherence to budget, adherence to timetable, technological sophistication) and significantly less on social or moral themes such as accessibility, with the notable exception of Helen Kennedy's work (more on this in Chapter 2). This study aims to help identify factors within web development groups that have been able to partially or completely overcome ableism to include higher levels of accessibility in their end products.

To what extent does a collective educational background in web development or design influence whether a group will produce accessible sites? Is there a correlation between accessibility and groups that design websites for other clients versus groups that design websites exclusively for their company? To what extent are budgets a factor in determining whether accessibility components make the cut? What about lack of knowledge of accessibility methods, or legal requirements? Are accessibility components included in the core offering, or only when specifically requested by a client? These are all questions that have not been adequately answered by scholarly inquiry into the means of production of websites, which this dissertation aims to begin to correct by examining the construction of routine practices of web design and development through an ableist lens.

2. **POWER RELATIONS: What role do power relations within web development groups play in "mangling" the routine design and development process with respect to affecting the accessibility of the end product?**

Web development groups, particularly those that design and develop websites for external clients, are under significant time and budget constraints. Additionally, since the customer is often external to the organization, the organizational prerogatives of the web development group will frequently be overridden by the wishes of the client (Fodness 2011; Kennedy 2012, 212). Within the organization, there are tensions between management—which strives to produce websites that meet stated requirements on time and on budget—and the web designers and developers themselves, who often have their own strongly held ideas about what constitutes "doing good work."

Andrew Pickering describes his concept of the "mangle of practice" as "a *dialectic of resistance and accommodation*" (Pickering 1995, 22). In the context of accessible web development, I posit that the decision to include accessibility components, and which components to include, is a process of resistance and accommodation between various actors and their negotiation of technological barriers and affordances—between web developers and managers, between web development groups and external clients, between web developers (the people who write the code) and web designers (the people who create the visual appearance of a website), between web development groups or clients and the

law or threat of legal action. It is important to note that not all participants in a website build are possessed with equal political capital, or the ability or position to express it—thus, decisions about whether to make a website accessible or not, and to what degree, are “mangled” by the difference in political stature of various agents, and differences in knowledge about technological barriers and affordances, in ways that will not be consistent from group to group. How each group determines whether and how to deal with accessibility concerns at a social and technological level creates a “branching history of choices” in which the process could have taken different paths at multiple junctures (Eglash 2011, 998).

As Andrew Feenberg has argued, experts and managers occupy privileged decision-making positions that affect the embedded politics of the products they oversee, which can pose problems when the design and implementation of the technology is insufficiently democratic (Feenberg 1999, ch. 6). I will investigate to what extent my informants characterize the typically meritocratic process of website development as being “mangled” by resistance and accommodation based on individuals' unequal political capital within and between organizations—to what extent managers and experts influence the design of websites in web development groups versus client requirements and user testing—and to what extent web technologies are created in a democratic way with input from users with impairments.

There are four main groups that contribute to power relations in web

development groups: designers, developers, managers, and clients. Designers are interested in the look and usability of the website. Developers are concerned with the programming and implementation of the website. Clients have specific goals in mind for what the website should look like, and what functions it should perform. Managers, including team leads and account executives, are responsible for synthesizing the needs of these groups to produce a product that is satisfactory to all parties. These four groups are engaged in power relations with one another during the design and build process.

I hypothesize that the inequalities introduced in the idealized meritocracy of web development groups play the greatest role in determining the accessibility level of the end product—either because web designers don't want to compromise their designs for accessibility, or because developers don't want to spend the extra time and effort to add in accessibility features, or because managers don't want to risk running over budget or delivering late as a result of perceived “extras,” or because the client introduces requirements that make accessibility best practices extremely difficult or impossible to meet, including limited time and budget for a project. As Kennedy argues, “[Study findings] suggest that convincing clients, equivalent stakeholders and external others of the importance of accessibility is experienced as one of its greatest barriers by web designers” (Kennedy 2012, 212).

I hypothesize that the power relations between these groups tend toward edging out accessibility in favor of other considerations, most likely due to the

non-privileged position of the group or individual that may be making arguments in favor of accessibility. I will argue that prioritizing factors such as time to launch, cost of the website, and preserving the visual integrity of the design at the expense of accessibility are all ableist practices, and lead to disempowerment of accessibility advocates within organizations. Furthermore, I posit that if a web development group has a routine practice of including accessibility components, use of non-democratic political capital will be the deciding factor in dropping these components from the list of requirements for the final website.

The study of power relations as a major contributing factor to the accessibility of websites will contribute to disability studies as an extension of the way that ableism manifests itself in how society comes to produce websites that are inaccessible—in particular, by examining the agency of individuals in the process, and demonstrating the complexity of the social factors that go into a website build. With this research question, I hope to add to the social model literature by delving deeper into the complex causes and manifestations of of ableism within web development groups.

Looking forward, what interventions could be proposed to alter how power relations could work within web development communities of practice to produce more socially just outcomes and create a more accessible web? The best way to answer these questions is to conduct in-depth interviews with various different stakeholders within web development communities of practice to get a sense of the organizational dynamic that exists in those ecosystems—the results of which

are presented in detail in Chapter 5.

3. **DISCIPLINARY NORMS: How are disciplinary norms constructed within web development groups? Do these norms include practices for producing accessible sites?**

Disciplinary norms are a complex social institution, and can be part of the mechanism that facilitates inclusion or exclusion. If accessibility is valued as part of the disciplinary norms of web development groups, does that norm translate into making accessible sites, or are those norms overruled by other factors? Helen Kennedy, in her 2012 study of web development groups and accessibility, concluded that these norms are fostered by a combination of published web standards (Kennedy 2012, 85) and individual designers and developers following thought leaders in the field, typically through blogs and social media, but also at discipline-specific conferences (Kennedy 2012, 104). Kennedy found that ethics and web standards are widely valued by web designers (Kennedy 2012), but it is unclear from her research to what extent these values translate into having an accessible end product.

Given that most web development groups are constituted of non-impaired individuals (Kennedy 2012, 214), and given that many web development groups do not employ user groups at all, much less diverse user groups (Fodness 2011), accessible design that is being done in web development groups is likely based off of standards of practice and web development norms. What are these norms, and how are they constructed?

I posit that if accessibility is not part of disciplinary norms, then it is likely that sites will likely not be accessible unless specifically required by the client. In the research I conducted for this dissertation, I have found signs that point to a fairly widespread belief that producing accessible sites falls within the realm of professional web designer and web developer discipline at an individual level, agreeing with Kennedy's findings—but that value does not necessarily carry over to the practices of the organization, which is where Kennedy and I diverge.

I will examine more closely how web developers and web designers conceptualize disciplinary norms in their particular area of expertise, whether they believe that it includes producing accessible sites as a value, and in particular how disciplinary norms play out under limited budgets and tight timelines. The social model largely focuses on the results of disabling social forces, whereas this study examines the roots of disabling forces in web development, including how education and social experience contribute to form disciplinary norms that are inclusive or exclusionary. This study extends the social model and ableism literature to add an expanded perspective on the ways that disciplinary norms are constructed in web development groups, and the way that larger social projects such as open source software can contribute positively to accessibility.

Published standards exist for the underlying languages that shape the web, including HyperText Markup Language (HTML) and Cascading StyleSheets (CSS), as well as standards for producing accessible websites, such as the

World Wide Web Consortium's Web Content Accessibility Guidelines (WCAG). Is following these standards a widely accepted component of web developer disciplinary norms? My research into this topic points to language standards (HTML and CSS) being incorporated into disciplinary norms, but accessibility standards being largely ignored, or viewed as an “extra” (Fodness 2011)—a finding that is in contrast to Kennedy's work, who maintains that web designers produce accessible sites by employing what she terms “ethical self-regulation” (Kennedy 2012, 100). By extending Kennedy's work on how disciplinary norms are shaped in web development communities, particularly as it relates to modern web development practices in the “mobile-first” age of designing websites that work well on smartphones and tablets, I posit how these disciplinary norms might be reshaped to incorporate web accessibility without the imposition of strict top-down requirements.

4. **SYNERGISTIC ENABLEMENT: To what degree does the accessibility of a site hinge upon tying accessibility to other concerns, such as increased SEO scores, or technological choices, such as frameworks?**

Search Engine Optimization (SEO) is the branch of digital strategy concerned with making a website appear higher in search rankings for specific terms, ideally causing the website to appear on the first search engine results page (SERP). Early forms of SEO, or “black hat” SEO, involved gaming the system to artificially produce high search engine scores by stuffing websites with

keywords (often rendered invisible or listed at the bottom of the page below the footer) and creating so-called “link spam” or “blog spam” on other websites in order to increase the number of inbound links to a site, thereby artificially inflating the perceived importance or impact of a particular website. In academic terms, you can think of “link spam” as submitting articles containing citations to a target publication to journals with low barriers of entry in order to increase the number of citations to the target publication.

Modern SEO, by contrast, is viewed as “white hat” SEO—making the content on a website of genuine interest and usefulness to an audience, and matching up the language in the site's text to the actual terms that users are searching for to make it easier for users to find the information that they want. In fact, so-called “white hat” SEO workers tend to view “black hat” or “grey hat” SEO workers with disdain (Gavrilas 2011).

In the “white hat” SEO world, the more completely search spiders such as Googlebot can understand a site's content, the more of that content will be indexed and ranked by the search engine, and the better the rankings will be. In this context, providing a transcript for a video allows for full-text indexing of that video's content, whereas merely embedding the video does not allow a search engine to be able to understand the video's content. Likewise, providing meaningful and descriptive alternate text for an image allows the search engine bot to understand what the image represents and index it properly.

Using clear and simple language written at roughly an 8th grade reading

level and providing text alternatives for all multimedia content makes it easier for search engines as well as human beings to understand the information being presented on the website. In this context, what is good for Googlebot is generally good for accessibility—transcripts can be used by search engines as well as human visitors that are deaf or hard of hearing, alternate text on images can be used by people with screen readers, and users with cognitive impairments benefit from clear and simple language used in the website copy (Kennedy 2012, 214). The reason for this is largely technological convergence between search engines and assistive technology: the same website rules or structure that make a website comprehensible to one piece of software (an indexing spider) make a website comprehensible to another piece of software (assistive technology, such as a screen reader).

Web design and development has become significantly more complex over the 20 years since the web expanded from government and academic institutions into people's living rooms, offices, and bedrooms. Gone are the days when websites were designed and built by one person with a text editor and a file transfer program. Nowadays, websites need to be designed and tested on a wide range of devices, operating systems, browsers, display resolutions, and other methods of access. Rich multimedia requires the use of different programs to create, edit, and incorporate diverse assets into a website. Distribution of content authorship, advanced feature requirements, and security concerns surrounding code written by inexperienced developers have led to the creation and

widespread use of content management system (CMS) platforms.

To combat the problems inherent with building and testing advanced interfaces, many developers have turned to using standardized CMS platforms and front-end (display) frameworks such as Twitter Bootstrap to solve these problems—many of which are maintained by large, distributed teams that include accessibility components out of the box. Given these modern realities, to what degree are websites made accessible by the (primarily technological) choice of platform or framework versus intentionally building in accessibility features? To what degree is valuing and including accessible components an intentional choice by the project team versus a byproduct of modern web development practices as a whole?

My term for the “ride-along” effect that accessibility components enjoy as it relates to these types of technologies is “synergistic enablement.” Although this phenomenon exists in other contexts, it is particularly important under a neoliberal, or excessively market-driven, frame as it allows for a synergy, or confluence, of factors involved with modern web design and development practice to include accessibility components without a focus on people with impairments being the primary driver for inclusion. In this instance, I view the “ride-along” effect as being positive, but the phenomenon is neutrally valenced. Take, for example, Facebook's data mining as riding along on the “free” service that they provide, where payment is invisibly exacted via corporate use of personal (and often private) data, or Winner's assertion that technologies like

nuclear power plants “require authoritarian management” (Winner 1988, 175) as their form of ride-along. This dissertation explores the ways in which synergistic enablement can exist as a positive force within web development communities of practice, and how it can be leveraged to broaden accessibility of the web within a neoliberal frame.

1.3 Methodology

Web development, as a discipline, is in its infancy. Unlike more established disciplines, such as engineering and architecture, web development does not have a commonly agreed-upon set of standards of practice, or a regulatory body overseeing educational and work product standards. Web developers may be freelancers or full-time employees working for a web development company. The types of companies that web developers might work for are likewise incredibly varied, including both public and private sector businesses as well as government agencies. In some cases, web developers will be responsible for nothing but websites owned and maintained by the company they work for. In other cases, web developers will be employed by a company such as an advertising or marketing agency, or work as a freelancer, building websites for other organizations and individuals. Web workers that are employed directly or indirectly by government agencies will have an explicit mandate to include accessibility features because they are governed by Section 508 legal requirements.

Because the work practices of web developers are so varied, and because

the discipline of web development is so new and unsettled, I felt that the best approach to determining what factors affect the level of accessibility of websites generally would be to speak to a wide range of web developers in different work circumstances, to determine what factors are common among them and what factors might contribute more significantly to producing accessible sites.

To that end, in 2011, I conducted a 26-question survey of 330 web developers to determine to what extent web developers were including accessibility components in their websites, and what correlation there might be between coding for accessibility and factors such as job type, participation in ongoing education in their field through attendance at conferences, and the respondent's level of education. This survey data has been supplemented with the results of 22 semi-structured interviews of web developers, designers, and managers that are part of my professional network to dig deeper into the more complex questions that arose out of analysis of the survey data. Interviewees included freelancers, developers that work for companies whose primary business is making websites for other companies (such as marketing/advertising agencies and web development shops), and developers that are primarily responsible for maintaining websites owned by the company that employs them (such as e-commerce sites, corporate websites, and intranet sites).

In order to determine what factors influence the accessibility level of web properties, it was important to compare and contrast the work experiences of web developers in different types of work roles, as well as different types of

organizations. For instance, are there common educational factors that influence whether a particular web developer is more likely to code for accessibility, irrespective of their type of work environment? Are freelancers more likely to code for accessibility because they have more freedom, or are they more bottom-line oriented and/or client requirements oriented, to the exclusion of accessibility features? In order to answer these questions, a broad survey of web developers in different work environments is essential.

I have personally been building websites as a hobby and a career for 20 years—most recently as a full-time web developer for the past four years at a marketing agency. This experience has allowed me to build out a professional network of web workers, whom I solicited for interviews and background material. This dissertation is a combination of my 20 years of personal experience and knowledge of web development methodologies combined with that gleaned from survey and interview data. Having professional connections with my interviewees —in some cases having worked with them directly—when combined with my technical background allowed me to dig deeper in the interview process. Helen Kennedy, whose work I draw on extensively within this dissertation, also has a background in web design and development. She includes her experiences in her book, following the tradition of a “retrospective ethnography” (Kennedy 2012, 13–14), a tradition which I employ here when discussing my own experiences as a web developer.

The interviews I conducted were semi-structured, with guiding interview

questions, and flexibility to speak more in-depth about particular topics of interest to myself and/or the interviewee. Most interviews took approximately one hour to complete. Interviews covered a range of subjects, including accessibility background and techniques, educational background, work experiences, social and organizational interaction with clients, managers, and end-users, and personal cultivation of disciplinary norms.

All interview transcripts have been coded to the theoretical constructs and research questions I am using throughout the project. In particular, I examined the interview data for points of similarity, particularly within and across work spaces, as well as points of difference, that can be explained by particular elements—such as level and type of education, work environment, disciplinary norms, external social networks, exposure to people with impairments, user group testing, et cetera.

Routine Design & Development Practices: What common elements shape accessible websites? Are company-produced sites more likely to be accessible than freelancer sites? Does a liberal arts education more often than not translate into a focus on accessibility? Are there exceptions to trends, and can they be explained? Interview transcripts were coded for this type of analysis, to draw conclusions across a broad dataset.

Power Relations: What role do managers play in determining accessibility of the end product? What about client requirements and budgets? Does the law or regulation play a role? What about internal team leadership?

Interview transcripts were coded for analysis of power relations within and between organizations that affect web developers in their work practices, as well as the level of collaboration or democratic consensus-building that occurs within web development groups.

Disciplinary Norms: What do individuals conceive of as the “disciplinary norms” of a web developer? What thought leaders do they follow outside of their workplace? Do any of these thought leaders promote accessibility? Is there commonality between web developers across different types of work environments? How do disciplinary norms translate into daily work practices? Is it over-ridden by power relations? Interview transcripts were coded to identify and analyze the construction, maintenance, and negotiation of disciplinary norms in the workplace, and how disciplinary norms translate (or do not translate) into accessible websites.

Synergistic Enablement: How are accessibility components “sold in” to a website build—to managers, creatives, and clients? Are they included automatically, or framed as part of a broader goal (such as SEO), or included as part of technological choice (CMS or framework), billed as an optional extra, or excluded completely? Interview transcripts were coded to identify instances of, and arguments in favor of, what I perceive as examples of synergistic enablement.

Finally, the survey data has been analyzed by loading results into a MySQL database and running intersectional queries to compare the results of

specific data sets, which have been analyzed using statistical methods to demonstrate the statistical significance of differences between specific populations.

1.4 Structure of This Dissertation

This dissertation is laid out in a fairly straightforward manner. First, I provide a literature review to situate this dissertation in the literature on the social model of disability, ableism, transcoding, bridge methodologies, Helen Kennedy's Net Work, and the literatures that I will be building on to advance my concept of synergistic enablement. Next, I provide background and context on the technologies that are typically used to build websites, policy and legal frameworks designed to help people with impairments, and the accessibility standards produced by the World Wide Web Consortium (W3C).

Following these background chapters, I devote a chapter to each of my four research questions—routine design and development practices, power relations, disciplinary norms, and synergistic enablement. In these chapters, I present the results of my survey and interview data that are relevant to the question, and tie those results back to Kennedy's findings and theoretical constructs introduced earlier. Finally, I present my own recommendations for how to improve web accessibility, based on my experience, research, and feedback from my informants.

2. LITERATURE REVIEW

2.1 The Social Model of Disability

> “In our view, it is society which disables physically impaired people. Disability is something imposed on top of our impairments by the way we are unnecessarily isolated and excluded from full participation in society... From this social point of view it follows that the impoverishment of physically impaired people arises out of the fact that, as a group, we are excluded from the mainstream of social activities.” (Oliver 2009, 49)

Disability studies makes a strong argument for “full participation” for people with impairments as a necessary prerequisite for economic and social equality with the non-impaired. Web accessibility is an excellent example of an arena where people with impairments are systematically excluded, and are suffering economic and social consequences as a result of their exclusion—particularly as the Internet becomes more central to work and social life in the developed world. In particular, Julia Scherer argues that the capitalist emphasis on job performance, combined with inadequate assistive technology to help cognitively, behaviorally, or socially impaired people succeed in the information economy, has resulted in employment discrimination for these groups (Scherer 2005, 55).

Michael Oliver's “social model” of disability contends that it is society that turns an impairment into a disability, which is analogous to the STS concept of social construction applied to disability—the fact that society sees impairment primarily through the lens of the medical model, instead of as a “mere-difference” in ability or physical and cognitive makeup, is only one of many possible outcomes. Similarly, what turns an environmental “hazard,” such as an earthquake, into a natural disaster is its effect upon society—the distinction “depends on a population’s exposure to a hazard, the vulnerability of that population, and their ability to respond to the hazard” (Jencik 2010, 8). Oliver describes the intent of the social model as “an attempt to switch the focus away from the functional limitations of individuals with an impairment on to the problems caused by disabling environments, barriers, and cultures” (Oliver 2009, 45). Politically, the move to frame hazards as “natural disasters” instead of taking social/cultural responsibility for environmentally destructive behavior and/or poor planning is a similar, yet antithetical, example.

Likewise, the social model exists in antithesis to the “medical model” of disability, which focuses on impairment as a temporary condition or illness, able to be “cured” (given enough time) with advances in scientific and medical technology. The social model is an attempt to focus instead on the social and material practices of systematic exclusion of particular groups of people based on the physical and mental characteristics of impairment. Similarly, reframing “natural” disasters as the inevitable consequences of environmental pollution and

poor institutional planning shifts the conversation away from the inevitability of "acts of God" into a realm where human society can actively work to prevent the conditions that cause or exacerbate disaster. The social model attempts to reframe the existence of impairment as a given, and impairment as temporary, with the conditions of impairment understood as resolvable through accommodation from society. Such accommodation incorporates both social and technical fixes, as assistive technology is both necessary and inextricable from discussions of accommodation of impairment.

Of central importance to this philosophy is the locus of "harm":

> "For instance, is it the impairment itself that causes the harm? If so, we should focus on reducing or indeed eliminating the impairment, which is a common perspective. Such a view interprets disability as harmful in and of itself. In contrast, there is a view among some disabled people that whilst impairments at times cause inconvenience, tiredness and even pain, the primary source of harm is external to the person, situated in the realm of belief." (Campbell 2009, 16)

The realm of belief that Campbell is describing is the realm of belief in the social arena, as it is concerned with policy and unconscious action that favors the normate body over the impaired.

Oliver and the social model are not without their critics, however. One of the more notable critiques of the social model was written by Tom Shakespeare and Nicholas Watson, who argued that the social model has become its own worst enemy—the orthodoxy of the social model in British disability studies has pushed out all possibility of reconciliation with other methods, approaches, or theories, particularly with the minority and civil rights approaches common in American disability studies. Shakespeare and Watson argue that the social model brackets impairment as if it does not matter in disability politics, even though the social as well as the individual/physical condition contributes to the disabling effect (Shakespeare and Watson 2002). Bill Hughes is less critical than Shakespeare and Watson, given that he largely supports the social model, but he posits that Oliver focuses too extensively on the differences between disabled people and the non-disabled, instead of problematizing the normative notion of the “non-disabled” body (Hughes 2007, 680).

Constitutional and cyberlaw scholar Lawrence Lessig has argued that people who would not be considered “disabled” in physical space can be “enabled” in virtual space by the social and technical norms of virtual space. However, he has warned that advances in technology have begun to reverse this trend. For example, the introduction of video and graphics in websites and interfaces make “the blind become 'blind' again,” video and audio content make “the deaf 'deaf' again,” and the personalization of experience offered by photos and video of the self have removed anonymity for sex, race, and other factors in

online interaction (Lessig 2006, 87–88). However, Lessig warns against what he terms “is-ism”—“the mistake of confusing how something is with how it must be.” In fact, he argues that cyberspace “can be made to reflect any set of values that we think important. The burden should be on the technologists to show us why that demand can’t be met” (Lessig 2006, 32). Thus, the shift away from a text-based medium, which was inclusive because text is understandable and translatable by machines and assistive technology, to a less accessible multimedia environment is not an inevitable byproduct of technological determinism, but rather a choice of the built architectures of technologists and a reflection of social values.

Tom Boelstorff, in analyzing the digital space of Second Life, has argued that the impaired can be "embodied" in Second Life in ways they often can't in physical space, which can serve as an enabling technology for them. However, Boelstorff also argues that changes to the Second Life software have begun to erase this trend as there is more emphasis on physical ability to be able to interact with the software (Boellstorff 2008, 136).

While the field of disability studies has done an excellent job grappling with what STS would term the social construction of disability, and with the problems and promises of assistive technology, relatively little work has been done specifically with regard to impairment and the world wide web—with a few notable exceptions, including Kennedy, Lessig, and Harper and Yesilada. Instead, scholarly work has focused mostly on the limitations of the built

environment, or with interacting with non-web software programs installed on the computers of users with impairments.

2.2 Whose Responsibility? Ableism, Transcoding, and Bridge Methodologies

In “Upon Opening the Black Box and Finding it Empty,” Langdon Winner argued that the then-current approach taken in the Social Construction of Technology concentrated on the social conditions that produced technological artifacts to the exclusion of any significant analysis of the effects of those artifacts (Winner 1993, 368, 375). More recent work has begun to reverse that trend, as STS scholars are addressing moral arguments about how technologies are created, and particularly, the moral role of the individual engineer or technologist (Coeckelbergh 2006), (van de Poel and van Gorp 2006).

The moral role of members of web development teams in producing accessible content is contested. On the one hand, there is the theory of “ableism,” proponents of which argue that creating inaccessible spaces / interfaces / content comes from a position of abled privilege, and should be addressed with the producers. On the other hand, there are less idealistic approaches that take as given the idea that there are many spaces on the Internet that are inaccessible in different ways and to different degrees, and that individuals and groups are empowered to do something about them *post facto*—such as work that has been done in a technological discipline called “transcoding.” Transcoding involves individuals or groups creating middleware to

act as an accessibility filter for inaccessible content, often in a very targeted way, to provide access for people with impairments without recruiting content producers or developers in the process of improving accessibility at the source.

> “Ableism is a set of beliefs, processes and practices that produce—based on abilities one exhibits or values—a particular understanding of oneself, one’s body and one’s relationship with others of humanity, other species and the environment, and includes how one is judged by others.” (Wolbring 2008, 252–3)

The theory of ableism is an inversion of how much of disability studies examines people with impairments—by shifting the focus away from the body of the impaired person, and instead examining the “Other”—the construction of the non-impaired, or normate, body (Campbell 2009, 4). The theory of ableism, as it pertains to web development, contends that web developers exert an unconscious prejudice against people with impairments because web developers are most likely not seriously impaired themselves, and therefore non-reflexively apply the gaze of the Other when considering the users of their product (Kennedy 2012, 214).

Wolbring argues that ableism as prejudice against people with impairments (in the same vein of other -isms as prejudice against other groups of people) is “supported by the medical, deficiency, impairment categorization of disabled people (medical model) … it rejects the 'variation of being,' biodiversity

notion and categorization of disabled people (social model)" (Wolbring 2008, 253). In this regard, ableism mirrors Oliver's contention that there is an inherent social prejudice against people with impairments that stems from the abilities of the majority of the population, and it is the responsibility of society to fix this problem by providing accommodation for impairment.

Where Wolbring breaks from Oliver, and where many scholars of ableism differ from traditional disability studies, is to elevate the discourse above traditional scholarship surrounding impairment and accommodation—and instead focus on ableism as a critique of dominant discourses in which certain social values are placed above others (Wolbring 2008, 252–3). He goes on to argue that ableism is a lens through which we can evaluate and critique differences in ability that are made available to people who can afford to purchase technological enhancements to their bodies, elevating them above the "traditionally-abled" (Wolbring 2008, 257–8).

Campbell's definition of ableism is less ambitious than Wolbring's, but agrees that ableism reflects a medical model view of disability, stating that "... a chief feature of an ableist viewpoint is a belief that impairment or disability (irrespective of 'type') is inherently negative and should the opportunity present itself, be ameliorated, cured or indeed eliminated" (Campbell 2009, 5). Campbell draws on Latour's concepts of *translation* and *purification* to argue that the prospect of dividing people into able/disabled itself is fraught with problems, as "able" bodies are often mediated into such a state through pharmacological

intervention or technological enhancement or correction, and can become temporarily impaired through illness:

> “The processes and practices of translation cannot be separated from the creation of that ordering category termed 'disability'. For many people deemed disabled, in the world of technoscience their relationship with non-human actants has been profoundly cyborgical and hybridisable (e.g., the use of communication and adaptive devices, implants and transplants). As such the networks of association between human and non-human (sentient beings and machines) have always been and increasingly are pushing the boundaries of the practices of purification. The disabled body induces a fear as being a body out of control because of its appearance of uncontainability. The practices of purification insist on this being the case. Ableism's constitutional divide posits two distinct and entirely clear ontological zones: disabled and abled (normate).” (Campbell 2009, 7-8)

Barnes points out that even if ableism-as-prejudice were reduced or eliminated, there would still be “disabled” and “non-disabled” people in society, even if society did not put a negative value on the “disabled” body and a positive

value on the “non-disabled” body (Barnes 2014, 110)—but instead viewed it as “mere-difference” instead of “bad-difference” (Barnes 2014, 89). Therefore, reducing or eliminating ableism-as-prejudice could certainly alleviate prejudicial issues inherent to an ableist society, but does not eliminate the need for accommodation of impairment. This is similar to how there is not a negative value placed on being tall (being tall is a mere-difference, not a bad-difference), but it is still difficult for tall people to find clothing that fits them, because the market for big and tall clothing is much smaller than the market for clothing for the average person. In this example, clothing companies have failed to provide adequate accommodation for the mere-difference of being taller than average.

Transcoding exists as an alternate method of providing accommodation for impairment by remixing content through the use of middleware[5] to transform a non-accessible experience into a more accessible one. Historically, transcoding has focused on two main areas for increasing accessibility: simplification, including indexing, and re-ordering content. However, in recent years, transcoding has become more sophisticated, and is able to tackle more difficult problems (Asakawa and Takagi, in Harper and Yesilada 2008).

An example of a problem that can be solved by transcoding is when blind users need to navigate a site using screen readers, but are forced to skip through a long list of headers and navigation elements prior to getting to page content on every page load (Fernandes et al. 2006, 131). Ferndandes et al. propose a

5“Middleware” is a term that describes a piece of software that is interjected into the “middle” of a connection between two other pieces of software, such as between a Web server and a user's Web browser, which allows the content to be transformed before being delivered.

solution:

> … a transcoding system that automatically provides semantic classification of the Web page's structural blocks. The system applies a set of heuristics, based on statistical analysis, to determine where the actual text content, menus, and other sections of the Web page are. (Fernandes et al. 2006, 124)

The authors also propose an innovative mechanism for dealing with images linked to other pages where no suitable alternate text is found—drawing the alternate text from the page title of the link destination (Fernandes et al. 2006, 127). Notable about these approaches is that they can be applied with no intervention from website creators—in effect, empowering people with impairments through middleware to access previously inaccessible content.

Harper, Bechhofer, and Lunn propose a system that they call SADIe: Structural-Semantics for Accessibility and Device Independence. SADIe builds on the trend in web development to separate content from presentation—in technical terms, structuring the data (content) in a semantic fashion in HTML and storing all of the rules for visual display in Cascading Stylesheets (CSS) (Harper, Bechhofer, and Lunn 2006, 259). SADIe uses a human analysis of a target website's structure, including the role and function of elements and how those elements interact with one another, to create a supplemental stylesheet. This supplemental stylesheet is loaded into a server proxy which allows users of the

website who are utilizing screen readers to reorganize content on the page to provide faster access to specific content areas. The advantage to this approach is that one ontology affects the structure of an entire website, irrespective of how many pages are on that site. The disadvantage is that the ontology has to be created by hand, which means that, in all likelihood, only popular websites will be transcoded (Harper, Bechhofer, and Lunn 2006, 260).

While not strictly transcoding, Google is attempting to solve the problem of inaccessible user generated content (UGC) by using technological means. Google has employed speech recognition software to provide automatic transcriptions for YouTube videos uploaded by the general public (Sydell 2010). As UGC explodes in popularity (YouTube, Facebook, Twitter, Instagram, Vine, etc.) the problems associated with inaccessibility of such content will continue to grow, and the need for a form of assistive technology to translate that content into an accessible form will become increasingly significant. As the web is becoming more complex, transcoding and similar methodologies offer a tractable solution to the increasing problems with making websites and web applications accessible for people with impairments (Asakawa and Takagi, in Harper and Yesilada 2008, 256).

I argue that in the case of web accessibility, since the web is produced in such a distributed fashion by so many different actors utilizing different technologies across a range of educational backgrounds and socioeconomic conditions, that some sort of a bridge methodology is required in order to elevate

the accessibility of the web as a whole. While it is reasonable to legally require that large entities such as Amazon.com create accessible sites, it is unreasonable to apply that same standard to small businesses or individuals, particularly in the era of UGC. The ADA acknowledges and reflects this separation in expectations between large organizations that serve a large number of customers and have more financial resources for accessibility in its distinction between which organizations are responsible for providing what level of accessibility to a “place of public accommodation.” Small businesses are exempt from some of the more costly accessibility provisions of the ADA, because imposing them would put an undue burden on those businesses—which would serve as a form of discrimination against small businesses, due to the relative ease with which a large business can implement accessibility compared to a small one. Therefore, a broader exploration of a combination of social/cultural and technological fixes is required to make more of the web accessible to more people in such a way that accessibility can be practiced by small businesses and organizations as well as large ones with a wealth of resources at their disposal.

2.3 Extending and Updating Kennedy's Net Work

Helen Kennedy's 2012 study of web designers set out to solve the problem that “surprisingly little academic attention has been paid to the people who make websites, despite the pervasiveness of the products of their labour” (Kennedy 2012, 3). Noting that web work is political, and flows from the ethical

foundations of its practitioners, she set out to interview web designers to determine their ethical commitments to standards and accessibility (Kennedy 2012, 3).

In Kennedy's definition, “web designers” are “people who make websites,” irrespective of what roles they play (Kennedy 2012, 5). When referring to Kennedy's work in this document, I use her nomenclature—but my study will differentiate between different types of web workers depending on their role, since I posit that organizational roles have different levels of political capital, which affects power relations and outcomes.

Kennedy problematizes the makeup of the web design workforce, noting that it is overwhelmingly white (85%) and male (82%) (Kennedy 2012, 12), and that the nature of the round-the-clock responsibilities of the work tend to preclude involvement by women, especially women with children (Kennedy 2012, 34). However, she praises the work that is done by this workforce, citing commitments to Tim Berners-Lee's vision of the Internet as a “universal, open, interoperable, and accessible medium” (Kennedy 2012, 42), and the constant desire to become better at their craft (Kennedy 2012, 47).

In chronicling the history of web design, Kennedy demonstrates how much has changed since the first HTML standard was published in 1995, with modern web designers required to know “much more than HTML and CSS” (Kennedy 2012, 71). As web design has become more professionalized, its self-reflexivity with respect to ethics has also grown (Kennedy 2012, 80). Likewise, since web

design is continually in a state of flux, its methodologies and approaches are in a state of “interpretative flexibility,” where definitions and use are still being negotiated (Kennedy 2012, 81).

Kennedy notes that “web standards are of central importance to this community” (Kennedy 2012, 85), and posits that adherence to web standards in absence of regulatory requirements is a form of self-regulation that stems from a focus on producing “good practice and good products,” as an ethically informed decision that is “occasionally, more effective than legislative regulation” (Kennedy 2012, 85). As Andrew Russell argues, standards are created “to guarantee reliability, especially under conditions of duress,” but only have utility when “they can be taken for granted” (Russell 2015). Later in this dissertation, I will probe the differences between community adherence to language standards (such as HTML and CSS) versus accessibility standards.

Kennedy posits three synergistic benefits of web standards:

> “Thus, the benefits of standards are multiple: they help designers work better and more efficiently, help businesses stay on the right side of legislation because they are likely to result in accessible websites, and they make websites more findable because the code is easy for a search engine to read.” (Kennedy 2012, 92)

It is important to note that Kennedy lives in the UK, which operates under

a different set of laws concerning web accessibility than the United States. In the UK, the DDA explicitly requires that websites be made accessible, whereas in the United States there is no such specific requirement for non-governmental websites. Although Kennedy's work examines web designers in both the UK and the United States, when she refers to legislation requiring accessible websites, she is referring specifically to the DDA in the UK. However, she consistently extolls the synergistic benefits of “good business, good work and good ethics” (Kennedy 2012, 93) and problematizes legislative regulation (Kennedy 2012, 102), instead arguing in favor of ethical self-regulation of web designers (Kennedy 2012, 100). Furthermore, she warns against focusing too much on the inaccessibility of the web, arguing that it “overshadows any acknowledgement of the advances that have been made in web accessibility and the commitment of many web designers to the accessibility ethos” (Kennedy 2012, 108).

Despite Kennedy's support for standards, she points out several problems with the WCAG, the prevailing web accessibility standards:

> “In particular, the accessibility needs of people with cognitive, learning and intellectual disabilities have been poorly represented in the WCAG… It is only by going through the WCAG guidance manually and testing the site with people with a range of physical, sensory and cognitive disabilities that a web designer can know whether a website is accessible or not. The

> accessibility of a website needs to be tested by people with disabilities, and this is much more complex than simply inserting a URL into a text field and clicking a button.” (Kennedy 2012, 118–0)

Where designing for accessibility can break down, according to Kennedy, is in this testing phase. It is relatively inexpensive to build an accessible website according to best practices and standards, but usability testing involving people with a range of impairments can be costly, and is “the only real guarantee of accessibility” (Kennedy 2012, 147).

Kennedy also problematizes the rise of user-generated content (UGC) as creating additional barriers to accessibility. If a site owner turns over responsibility for creating content to users, who are not bound by legal requirements to make content accessible, and are typically not aware of the need or means to do so, overall accessibility goes down. Activities meant to enhance accessibility, such as requiring alternate text to be entered for images on image sharing sites, would likely have a detrimental effect on usage and content sharing on those sites (Kennedy 2012, 164).

Of significant importance to the ethical self-regulation of web designers is the concept of community, and the leaders, or “micro-celebrities,” in that community. These “micro-celebrities” have followings on blogs and social media channels, are thought leaders in their communities, and tend to advocate for standards and ethical prerogatives such as accessibility (Kennedy 2012, 193).

However, there is a self-reinforcing nature of exclusion within these circles, as conferences and discussion spaces tend to be white and male dominated, keeping other voices at the periphery (Kennedy 2012, 188, 193).

Kennedy leaves the reader with an overall optimistic impression of the state of web accessibility, but with a caveat that her work was primarily done with respect to web designers' own perceptions of their commitments to accessibility and ethics, without validating whether those commitments translated into actually accessible sites (Kennedy 2012, 215). This dissertation's findings, particularly in light of the survey and interview data that I collected combined with developments in web design since Kennedy's data was collected, suggest reasons to be less optimistic about unreinforced ethical self-regulation as the primary driver of accessibility on the web, even though I rely extensively on self-reported data as well.

2.4 Synergistic Enablement

The inherently technical nature of web development makes for a fertile ground for a hybrid approach between cultural and technical solutions, particularly because web developers are used to balancing social and technical concerns during the normal course of their work—for example, making an intuitive user interface requires a deep understanding of behavioral norms as well as technical ability for implementation. I propose that web development groups and advocacy groups can do a better job of linking social and cultural considerations surrounding impairment with a combination of the social and

technical. For instance, if developers of content management (CMS) platforms worked more closely with accessibility experts, they could build in accessibility features, so that non-professional, non-technical, or “lay” website creators could make accessible sites more easily. To some degree, this is already occurring, such as the Make WordPress Accessible working group for the WordPress CMS (Make WordPress Accessible 2015).

It is difficult for web developers to convince clients to voluntarily invest the money in developing accessible content, such as creating transcripts for videos, if the target audience is the relatively small percentage of the population that is deaf, particularly in the absence of legal regulation forcing such accommodation. However, if video transcripts are reframed as not only good for deaf people, but good for Search Engine Optimization (SEO) and organic search traffic, then clients are much more likely to invest the money in creating accessible content. Harper and Yesilada point out that there are synergies between solving problems for users with impairments and solving problems related to other types of accessibility, such as mobile usage, so time spent on one provides positive effects on the other:

> “Web accessibility is really just a über use case because in the end, we will all be handicapped by the technology or the environment. Work on Web accessibility is helping us address many other domains including those centered around user

> mobility. For instance, work on physical disability and the Web is helping solve problems of the usability of mobile technology." (Harper and Yesilada 2008, xvi)

The concept of synergistic enablement is a more specific example of reframing a problem to create a path to a mutually beneficial solution, as advocated by other STS scholars, including Linda Layne's concept of the "cultural fix":

> "[The Cultural Fix] focuses attention on changed meanings as a vehicle for identifying problems and solving them, which shifts the significance of cultural meanings for people and social action from broad underlying assumptions to specific configurations of meanings that challenge individuals in specific contexts." (Layne 2000, 509)

Layne notes that "one of the criticisms of social fixes by proponents of technological fixes is that they are not practical, that it is notoriously difficult to get people to change their habits and attitudes" (Layne 2000, 510). However, in the case of synergistic enablement, I am proposing the opposite—the social or cultural fix in this case is to attach socially desirable but politically or financially difficult objectives, such as accessibility, to objectives that are politically or financially desirable, which makes it easier for decision-makers to shift their habits and attitudes to encompass the change. As Harper and Yesilada argue,

“*Web accessibility means commercial success*” (Harper and Yesilada 2008, xvi).

As Bijker & Pinch point out, technological problems often aren't “solved” in a traditional sense, but are “closed” or “settled” by partially reframing the problem as the technology is developed. In the area of accessibility, this has happened when moral or social obligations to provide accessible content, or professional commitments to producing well-formed and standards-compliant code, are “solved” by reframing the argument in favor of optimizing the page for search, or by adopting a content management system (CMS) or front-end development framework to improve build efficiency—while simultaneously enhancing the accessibility of the site by choosing a CMS or front-end framework that comes with accessibility components built-in. This shift is an analogue to Pinch and Bijker's example of the air-filled bicycle tire being adopted not for its initial purpose of reducing vibration, but rather for its ability to influence the speed of bicycles to win races (Pinch and Bijker 1984, 426–8).

Synergistic enablement is decidedly utilitarian in its approach. The overall impact on the lives of people with impairments is occurs irrespective of the individual motivations of web designers or developers in this regard.

> It is the business of ethics to tell us what are our duties, or by what test we may know them; but no system of ethics requires that the sole motive of all we do shall be a feeling of duty; on the contrary, ninety-nine hundredths of all our actions are done from other

> motives, and rightly so done, if the rule of duty does not condemn them. It is the more unjust to utilitarianism that this particular misapprehension should be made a ground of objection to it, inasmuch as utilitarian moralists have gone beyond almost all others in affirming that the motive has nothing to do with the morality of the action, though much with the worth of the agent. He who saves a fellow creature from drowning does what is morally right, whether his motive be duty, or the hope of being paid for his trouble… (Mill and Sher 2001, ch. 2)

Another example of this is the recent trend of including captions or transcripts for online video not specifically so they are accessible to the deaf, but so they are accessible to search engine robots and spiders (Snow 2010). This is not the kind of “reframing” that Layne would argue for, but it has produced positive results in the area of accessibility, as people who are deaf or hard of hearing benefit from the transcripts and captions irrespective of why they were created in the first place. Accessible technologies are viable, but as Latour argues, they still need political and social support to be adopted (Latour 1996).

In this way,

Certainly, it seems that the web is socially constructed to be a largely inaccessible place, but very little work has gone into determining why that is the

case and what upstream paths might offer opportunities to change it. Transcoding approaches take as given that the web is inaccessible, and employ a combination of human and technological means to mediate its inaccessibility. Even Kennedy's excellent investigation into web designers and accessibility practices ended with an admission that the study was primarily about web designers' perspectives on their own commitment to ethics, web accessibility, and web standards, without performing checks against the websites produced by her interlocutors to determine how accessible they were in fact (Kennedy 2012, 215).

This dissertation attempts to provide a more robust investigation into the practices and politics of web designers and developers to determine why the web continues to be so inaccessible, and challenges the idea that theories such as ableism are an appropriate distillation of the motivations of web designers and developers. I will extend and challenge Kennedy's findings and conclusions, as well as critically evaluate the potential uses of transcoding methodologies in the evolving web. Finally, I will propose an alternate approach that focuses on accessibility writ large, which helps to sell in the added cost of accessibility by demonstrating additional concrete benefits in other areas.

3. BACKGROUND AND CONTEXT

Impairments in our eyesight can be accommodated by glasses. Impairments in walking can be accommodated by wheelchairs. Impairments, such as not being able to walk, do not have to result in limitations, such as one's ability to move oneself around, unless society fails to provide accommodation for such movement, such as curb cuts for wheelchairs. Where such accommodation does not exist, the impairment becomes a disability, because the lack of accommodation causes a disabling effect on the impaired person. In particular, if society does not provide accommodation for assistive technologies, such as ramps and curb cuts for wheelchairs, then the potential liberatory power of assistive technologies cannot be fully realized.

Michael Oliver (2009) argues that what turns an “impairment” into a “disability” is the design of the built environment, which he calls the “social model” of disability. In cyberspace, the built environment is defined by code (Lessig 2006). Therefore, a “disability” in cyberspace is a combination of a physical or mental impairment of an individual combined with code that does not accommodate that impairment by enabling these individuals to navigate cyberspace. The widespread lack of such accommodation means that individuals with impairments are an under-served community in cyberspace.

The intentional and unintentional ableist practices of many web development groups result in websites that range from being completely inaccessible to highly usable by users with a range of impairments. In order for a

website to be considered “accessible” to a person with a particular impairment, the user needs to be able to execute all basic functions of the website. Accessibility can vary from impairment to impairment, even on the same website. For instance, a website with audio content and no transcripts could be accessible to a blind user but not a deaf user. Additionally, even if a website is “accessible,” it may not be very “usable” for people with certain types of impairment. “Usability” is an abstract measure of how easy it is for the average user in a particular user category to navigate and take desired actions on a website. Even when a website provides accessibility for users in particular categories of impairment, the website may be significantly less usable for the user with an impairment than it is for a non-impaired user.

What constitutes disability in the context of cyberspace is disputed. The dominant standard for making accessible websites is the Web Content Accessibility Guidelines (WCAG) 2.0, which is published by the Web Accessibility Initiative (WAI) division of the World Wide Web Consortium (W3C). The WCAG 2.0 defines cyber-disability as one of seven categories, with very specific subcategories (W3C 2005).

The Americans with Disabilities Act (ADA) (*Americans with Disabilities Act* 1990) has a different definition for disability (U.S. Equal Employment Opportunity Commission 2009), but it was not a law specifically created for the web, and thus includes definitions for disability more broadly. Likewise, the Disability Discrimination Act (DDA-UK) of the UK (*Disability Discrimination Act* 1995), which

does include specific language about web accessibility, the Disability Discrimination Act (DDA-AU) of Australia (*Disability Discrimination Act* 1992), and the Canadian Human Rights Act (*Canadian Human Rights Act* 1985) provide different definitions for disability, as do disability laws in most countries of the world.

There are competing standards for web accessibility that include different definitions of cyber-disability. For the purposes of this dissertation, I will be analyzing cyber-disability as defined by the main standards governing web accessibility in the United States—the WCAG version 2.0. Although the standards are problematic and contested, they are the generally accepted standards for government and private websites in the United States.

The population of users of cyberspace with impairments is ever-changing. Many people with impairments were not born with them. Many people that have a mild impairment do not consider themselves “disabled”—wearers of eyeglasses, for example. As a person ages, they tend to develop impairments—such as diminished eyesight and hearing and reduced fine motor control—which can affect their ability to interact with computers and technology in general. The Census Bureau reported in 2010 that 19% of the United States population has a legally categorized disability of some form. Predictably, the proportion rises with age—only about 10% of people aged 18 to 64 are categorized as “disabled,” but fully 38% of the over-65 population are. As far as impairments that affect web users (all data from US Census Bureau Public Information Office 2010):

- 0.5% (1.8 million people) are “unable to see printed words”
- 0.3% (1 million people) are “unable to hear conversations”
- 5% (16.1 million people) have a cognitive impairment or “a mental or emotional illness that interferes with daily activities”

Additionally, there is a subset of the physically impaired population that finds it difficult or impossible to interact with certain types of input devices, such as a mouse and/or keyboard. Interestingly, despite the fact that, according to the census data above, cognitive impairments include more than five times as many people as are categorized as blind and deaf combined, relatively little attention is paid to cognitive impairments in construction of and adherence to accessibility standards (Kennedy 2012, 214; Clark 2006).

3.1 Technological Context

People with impairments are effectively second class citizens online. Hard-fought gains in the accessibility of physical space, and access to earlier assistive technologies (AT) by people with impairments, are becoming less relevant as the Internet becomes more important to the daily lives of people in both developed and developing nations. The Internet is becoming increasingly central to education and work practice, as well as a means of purchasing goods and consuming media. Without adequate protections for accessibility online, people with impairments are being left behind as part of a different kind of “digital divide”—one based on method of access instead of whether one has access at

all.

Increasingly well-funded industry representatives are fighting hard against any kind of regulations on the Internet, arguing in one instance that groups like the FCC were promoting accessibility “at the impermissible cost of innovation” (Federal Communications Commission 2011). Web developers and content producers are very wary of any kind of perceived or actual restrictions on freedom of expression online, including requiring accessibility features for web content, due to the nature and structure of the web. Unlike creating physical buildings, which includes significant planning, a permitting process that involves an ADA review, and the ability of the government to reject the building plan for not conforming to ADA requirements (ADA.gov 2015), the web has no such approval structure—and I am in agreement with the hypothesis that such an approval structure would be detrimental to the liberatory power and ease of publication access of an open Internet. Therefore, alternative solutions must be sought in lieu of a top-down policy simply requiring all commercial websites to be made accessible—particularly in the neoliberal climate that exists in the United States, and to a lesser degree, in Europe. In order to be able to frame and advocate for such solutions, a deeper understanding of web technologies, and how they are deployed by web development communities of practice, is required.

Websites are essentially built on three technologies—HyperText Markup Language (HTML), Cascading StyleSheets (CSS), and JavaScript (JS). Although CSS and JavaScript are technically optional, virtually no modern websites are

built without them, so I will discuss all three together.

HTML isn't a programming language—it is a markup language. What this means is that it describes, or “marks up,” data. For example, a web developer might use HTML to describe a certain set of text as an unordered, or “bulleted,” list, like this, using the <ul> tag to denote that the items are part of an **U**nordered **L**ist, and the <li> tag to denote each **L**ist **I**tem:

```
<ul>
      <li>List Item 1</li>
      <li>List Item 2</li>
      <li>List Item 3</li>
</ul>
```

If you were to view this HTML code in a browser, without defining additional styles via CSS, this list would display like so:

- List Item 1
- List Item 2
- List Item 3

Since HTML is essentially a way of assigning semantic meaning to text, the same bit of HTML can mean different things to different audiences, depending on how the content is being consumed. For a sighted user, the list appears visually bulleted. For a blind user interacting with the content using a screen reader, the screen reader is able to read the HTML code and convey to the user that the content is in an unordered list (Watson 2005). HTML supports a wide range of different content descriptors, including addresses, emphasized

text, citations, source code, navigation elements, and a variety of form inputs. With the widespread adoption of the HTML5 standard, even more of these descriptive tags have been added. The practice of describing data using HTML by choosing the most appropriate tags is what is called “semantic coding.”

What happens, then, if the content is an image? Or a video? It is not enough to simply use the appropriate HTML tag in that case—something more is needed in order to present the content in a way that is accessible to people with a range of impairments. In the case of an image, there is an attribute for the image tag that allows for a plain text description of what the image contains, which is accessible to a screen reader:

```
<img src=”/assets/img/flower.jpg”
      alt=”A picture of a flower.”>
```

If the “alt,” or “alternate,” attribute is missing, then the screen reader won't be able to tell what is in the image, and won't be able to convey that information to the user. If the image is just for visual interest, that's not much of a problem—but if the image is a button that says “Add to Cart,” for instance, then a major usability problem arises. This was the case with the Target lawsuit in 2006 (National Federation of the Blind 2008), in which the alt text on images used for navigation on the homepage was so unusable that it led a test audience of blind users to abandon the test (Thatcher 2015).

CSS is the technology that transforms the rather bland output of HTML into the visual diversity that you see on the web. CSS is the technology that is responsible for defining fonts, colors, sizes, spacing, borders, and most of the

visual aspects of a website. CSS is not used by screen readers, but is very important to people that interact with a website visually, even if their sight is limited. Properly structured CSS will allow someone with poor vision to zoom in on a website to make the text larger and easier to read. Poorly designed CSS, such as styles that do not include enough contrast between text color and background color, will cause problems for people with limited vision.

CSS, as a technology, has been around for nearly as long as HTML. For nearly as long as CSS has been around, designers have been using it to hide the underlines on links to make the experience of using the website more visually appealing for the sighted majority of users, using the argument that the underlines are visually jarring. For a user that has no problem visually perceiving color, if the difference of the color of the link compared to the color of the rest of the text is stark enough for the user to discern that the colored text is a link, then the user will have no difficulty picking out links from copy. For partially or fully colorblind users, however, merely relying on color to identify a link can make a website nearly unusable (WebAIM 2013). Additionally, certain color combinations are problematic for colorblind users, and can cause disorientation or an inability to see certain pieces of website content.

Previously, I stated that HTML is not a programming language. Likewise, CSS is not a programming language—it is just a series of statements to describe the visual properties of elements on a page. JavaScript, however, *is* a programming language, which web developers can use to add interactive

components to websites. JavaScript has enabled the web to evolve as a platform for running full-scale applications, such as Gmail and Google Docs, in what has been commonly referred to as the “Web 2.0” paradigm. JavaScript can lend a lot of power to web interfaces by allowing elements to be added, moved, and removed from pages, often in response to receiving information from a web server through periodic polling after the initial page is loaded, such as when a web-based email provider automatically refreshes the user's inbox when a new message is received.

However, JavaScript can introduce accessibility problems if not properly programmed and tested (W3C WAI 2014). For instance, screen readers often have difficulty interacting with websites that rely heavily on JavaScript to manipulate on-screen content. The recent rise in the use of JavaScript frameworks to build highly interactive web experiences, such as Facebook, have contributed significantly to this problem (Sutton 2014). JavaScript frameworks allow web developers to more easily build interactive experiences by building on top of a codebase that has been extensively developed and tested by third parties or open source software groups, but the tradeoff is accepting the limitations along with the advantages, such as the limitations of accessibility due to framework maintainers not utilizing accessible methodologies.

3.1.1 WAI-ARIA

The Accessible Rich Internet Applications (ARIA) Suite, developed by the World Wide Web Consortium's Web Accessibility Initiative (WAI), provides an

additional set of attributes for HTML elements specifically for accessibility. “More specifically, WAI-ARIA provides a framework for adding attributes to identify features for user interaction, how they relate to each other, and their current state” (W3C 2014a).

WAI-ARIA attempts to address problems introduced by the use of so-called Web 2.0 sites—sites that are made more interactive by use of advanced CSS and JavaScript techniques, including Asynchronous JavaScript And XML (AJAX), and single page websites and web applications. WAI-ARIA provides a language to describe the relationship and purpose of certain controls to screen readers and other assistive technology. For example, if a button triggers an on-screen pop-up (sometimes called an “overlay” or “modal” window), there are ARIA attributes to describe the relationship between the button and the pop-up that displays when the button is activated. From the WAI-ARIA overview (W3C 2014a):

> WAI-ARIA provides web authors with the following:
>
> - Roles to describe the type of widget presented, such as "menu", "treeitem", "slider", and "progressmeter"
> - Roles to describe the structure of the web page, such as headings, regions, and tables (grids)
> - Properties to describe the state widgets are in,

such as "checked" for a check box, or "haspopup" for a menu.

- Properties to define live regions of a page that are likely to get updates (such as stock quotes), as well as an interruption policy for those updates—for example, critical updates may be presented in an alert dialog box, and incidental updates occur within the page
- Properties for drag-and-drop that describe drag sources and drop targets
- A way to provide keyboard navigation for the web objects and events, such as those mentioned above

In particular, WAI-ARIA provides a solution for enabling users of assistive technology to interact with more complex elements that would typically require visual interaction using a mouse. Functionality that is now common on the web, such as drag-and-drop interfaces, complex dropdown menus, and dynamically updating content can be made accessible by the use of ARIA attributes. In addition to being part of the HTML5 specification, WAI-ARIA is also available for use in the ePub 3.0 specification for making accessible e-books and the Scalable Vector Graphic (SVG) 2.0 specification for making accessible graphics (Cooper 2014).

3.1.2 Schema.org Markup

Schema.org is a website and a standards hub for describing data in a standardized way across various formats on the web, including HTML. It was founded in 2011 as a collaboration between Bing, Google, and Yahoo! “to create and support a standard set of schemas for structured data markup on web pages” (O’Connor 2011). Shortly after, Yandex, the largest search engine in Russia, joined the consortium (Brickley 2011). By providing a standard set of schemas for data across web pages, search engines and other types of applications and technologies are better able to understand information about data (metadata) across wildly differing implementations of data structure. The implications for this at the search engine level are that search engines are better able to read and interpret specific information about certain types of data. Schema.org markup can, for example, be used to identify various pieces of information about a person presented on a website, such as a name, title, photograph, and so on.

Schema.org markup is used to provide a consistent method for third party websites, applications, and tools to interact with foreign data in a predictable way. However, since schema.org markup is generic and extensible, it is not *de facto* accessible or understood by assistive technology—that is to say, there is not a core schema.org accessibility standard which web developers can utilize to write markup in a consistent and predictable way to be understood by assistive technologies. This difference makes schema.org markup in a decidedly different

category than established accessible methodologies such as adding an “alt” attribute on an image tag, which is a core standard in the HTML language specification, and is expected and understood by assistive technologies such as screen readers.

However, there is a project called the Accessibility Metadata Project, which received funding from the Bill and Melinda Gates Foundation, which created a supplemental and optional schema.org markup set specifically for accessibility components. This standard can help to make the accessibility features of a website better understood to assistive technology and search engines, which could theoretically limit or prioritize results that accommodate specific assistive technologies or techniques (Rothberg 2014).

Schema.org markup is in its infancy—at the time of writing, it is less than five years old, and is still somewhat limited in scope. However, the aim of schema.org markup is to abstract certain elements of data from specific implementations, which will allow for a greater ability to translate data into different formats, many of which can be made more accessible in ways specific to an individual's impairment. Building on the transcoding examples in Chapter 2, schema.org markup can be used to assist transcoding methodologies to extract data from websites independent of implementation-specific markup and present it in a tailored way. Google is already doing this in search results by allowing users to use Google to search for information specific to them that Google is aware of through Gmail, such as upcoming reservations, flights, and the like, which

surface the result directly in the Google search window (Google 2015).

3.1.3 Frameworks and Accessibility

The practice of web design and development has become increasingly complex since NCSA Mosaic was released as the first popular web browser in 1993 (National Science Foundation 2015). In the beginning, web designers and developers had to construct designs using only HTML with a rudimentary set of tags and attributes. Styling options were limited, as were options for including multimedia content like images, audio, and video. CSS came about in 1994, partially solving the lack of style support by providing a common standard that was adopted by the various browser makers at the time (Lie and Bos 1999, ch. 20). JavaScript was developed in 1995 by Netscape for its Navigator browser (Rauschmayer 2014, ch. 4) and was later standardized for inclusion into other browsers by the standards organization Ecma International in 1997, when the core of the JavaScript language was renamed to ECMAScript (Rauschmayer 2014, ch. 5).

A number of tools were created to help web designers and developers, especially those with little technical training, make and modify websites easily, including Adobe GoLive (1996), Microsoft FrontPage (1995), Macromedia Dreamweaver (1997), and Netscape Composer (1997). These editors are termed WYSIWYG (pronounced wizz-ee-wig), or “What You See Is What You Get,” because the interface allowed for visual manipulation of content, relegating all code writing to the tool itself. Using these tools, or others like them, web

designers, developers, and content creators could create a website with little more knowledge than how to use a word processor—knowledge of HTML, CSS, and JavaScript were not required. Early versions of these editors were notorious for producing code that was bloated and not standards-compliant (Lennartz 2008), but they were easy to understand and manipulate for an average user, and gained significant traction early on.

By the turn of the century, development of more significant websites went beyond the capabilities of a WYSIWYG editor, and web developers increasingly turned to Content Management System (CMS) platforms to assist with management and manipulation of content. Instead of creating individual HTML pages in a WYSIWYG editor, developers were able to create reusable templates that would wrap around dynamic content read from a database, and the content itself could be managed from a web interface and saved to the database. When a user would request a page through a web browser, the request would be processed by a server-side programming or scripting language such as Microsoft ASP, PHP, or Sun Java, which would make a request to a database, retrieve the appropriate records, dynamically insert them into the template at locations designated by the programmer, and display the rendered temple to the user as HTML code.

At this level, the web developer would need to know HTML, CSS, and JavaScript, since they would need to be written by hand for integration with the CMS' templating system, as well as the server-side language that the CMS uses,

and potentially knowledge of how to write database queries in a separate language called SQL, or Structured Query Language. Additionally, CMS platforms had their own unique template syntax and functions, which the developer would need to learn in addition to the language it was built with. As a consequence, barriers to entry to web development began to increase sharply.

There are many CMS platforms today. The most popular is WordPress, which, at the time of writing, is estimated to power approximately 24% of the entire web (WordPress 2015). The top four CMS platforms, as of writing, are WordPress, Joomla, Drupal, and Magento (W3Techs 2015), all of which are open source, are written in the PHP scripting language, and were designed to work with MySQL as a database engine. Many companies have offered pre-built installs of these platforms as a form of Software as a Service (SaaS), including WordPress.com, which is the most popular. Under this model, users can sign up for an account and create a website in a few clicks, and need not interact with any source code if they do not wish to. SaaS CMS platforms have largely replaced the WYSIWYG content editors of the late 90s as the go-to solution for non-technical website owners. In addition to these open source solutions, there are proprietary platforms that allow non-technical users to create websites, including Squarespace, Wix, Weebly, and even Facebook.

Users and technology platforms are increasingly demanding of the features and performance of websites. Websites became more complex and included more rich media through the early 2000s, and then mobile computing on

cellular networks exploded with the release of the iPhone in 2007. Suddenly, network speeds could no longer be assumed to be at least as fast as a DSL connection for the majority of users, and website creators had to begin to optimize assets such as images and CSS files for fast download on relatively slow mobile networks, and for performance on handheld devices with a fraction of the computing power of a desktop or laptop. When the iPhone was first released, and the transition to mobile-friendly development began, it took the web design and development world back in time by at least five years.[6]

An early approach to mobile was to create a completely separate mobile site that was served to the user based on detecting whether or not they were browsing the site on a mobile device. It took three years until the industry was able to begin to adequately compensate for mobile platforms in a holistic way with the advent of the Responsive Web Design (RWD) pattern in 2010 by Ethan Marcotte (Marcotte 2010). When he introduced RWD, Marcotte built on previous fluid or grid-based methodologies, but explicitly rejected the desirability of a separate mobile site with limited content specifically for mobile users. Instead, Marcotte proposed a mechanism by which the same content would be able to "respond" to changes in viewport size using a relatively obscure technology called CSS media queries. CSS media queries allow certain visual styles to be applied conditionally based on how large the user's browser window is in pixels

6 It is worth noting here that changes in digital technologies have the potential to be significantly more disruptive than changes in physical technologies, due to the relatively fewer limitations that digital technologies face during design, development, and implementation. One reason for this is that the hardware powering digital devices is still undergoing a period of rapid technological innovation and advancement, which allows for radical changes in technologies from one generation to the next.

(known as a “viewport”), which allows the same site with the same content to be viewable and usable across a range of device sizes, from mobile phones all the way up to large desktop monitors and televisions.

By 2010, Search Engine Optimization (SEO) was big business, and many of the previous ways of “gaming the system” for better search rankings had been discovered and eliminated by the major search engines. In order to rank well, websites had to conform to criteria, known as “signals,” laid out by the search engine providers. In 2010, Google began making modifications to its search ranking algorithms to prioritize websites that loaded faster than others (Singhal and Cutts 2010).

Among other things, web developers had to reduce or eliminate the use of many images on their websites, compress images that couldn't be eliminated to make their file size smaller, reduce the number of requests made for assets and resources by combining multiple assets into one file (such as one large CSS file for all of the styles on a site, instead of loading multiple CSS files, each controlling different things), implement server-side compression to deliver requested files more quickly, caching to tell browsers not to re-download files they had already downloaded, and other techniques in order to make their websites outperform the competition in load time and rendering speed.

Such performance tuning required advanced knowledge of not just the technologies and techniques needed to build a website and work with a CMS platform, but also how to configure and tune a web server. Finally, in 2015,

Google began penalizing websites that are not mobile-friendly by ranking them lower than websites that are configured to be mobile-friendly (Makino and Phan 2015), further necessitating the use of responsive web design techniques in order to make content accessible both to users on mobile devices and to rank well on search engine results pages.

As a consequence of this dramatic increase in complexity in the field of web design and development, a number of frameworks have emerged in order to make it easier for web developers to make complex websites without having an in-depth knowledge of all of the technologies required to build a modern website. The tradeoff here is that developers are leveraging technologies that they do not fully understand, and therefore may not fully grasp the implications for concerns such as accessibility, or have the knowledge or access to incorporate accessibility into websites built with these frameworks.

These frameworks fall into one of four categories:

1. **Software as a Service (SaaS) offerings**—such as WordPress.com, Squarespace, Facebook, and others, which allow non-technical users to build a website using tools provided for them by the platform owner.
2. **Content Management Systems (CMS)**—such as WordPress, Joomla, Drupal, and Magento, which, through a robust ecosystem of free and paid themes and plugins, can allow a moderately technical user to build a website with little to no knowledge of how to write code, or can be used by very technologically advanced users to develop fully custom websites.

3. **Back-end frameworks**—such as Laravel, Symfony, Zend, Microsoft .NET and others that allow developers to create custom websites using a collection of pre-built functionality that covers common tasks such as request routing (determining what to do when a user asks for content by entering or clicking on a URL), authentication (providing a mechanism for users to log in with a username and a password), and the like. These frameworks require a working knowledge of the programming language they were created with, as well as front-end web technologies, including HTML, CSS, and JavaScript.

4. **Front-end frameworks**—such as Bootstrap (Twitter), Foundation (Zurb), Bourbon, Angular (Google), React (Facebook), Ember, and others. Front-end frameworks are a collection of code and methodologies for rapidly building the front-end of a website—the part that the user sees when they load the website in a browser. Front-end frameworks can, and often are, used in conjunction with CMS platforms or back-end frameworks. Front-end frameworks make the tasks of creating responsive and interactive designs significantly faster and easier, but can be viewed as bloated, since they include a significant amount of code that will not be used on every website, but which still needs to be downloaded by the end-user. Often, front-end frameworks ship with, or are extended to include, “toolchains” that include other pieces of software to automate repetitive tasks, like optimizing and compressing assets for fast download by the user.

As one might imagine, the level and type of accessibility of the different types of frameworks varies wildly. With SaaS solutions, the responsibility for providing accessibility features almost squarely resides with the platform creator. Users of SaaS platforms have little to no control over the code that is generated by the platform, and cannot influence whether or not it is accessible, even if they want to. Therefore, users of SaaS platforms have no choice but to buy into the level and type of accessibility of the platform if they choose to use it.

CMS platforms, by contrast, are typically follow a shared responsibility model. The CMS primarily provides the administrative portal for content editors, and a framework upon which front-end, or display layer, code can be written in what is typically referred to as a “theme.” The accessibility of the front-end experience is therefore up to the theme developer. It is certainly important that the platform itself be accessible for content editors with impairments, but it is arguably more important that themes are created to be accessible as well, which they often aren't. If the theme is not accessible, then users of the website with impairments will find it difficult or impossible to use, irrespective of whether the CMS itself is built to be accessible.

In the case of purely back-end frameworks, the responsibility for accessibility falls almost entirely on the developer, since back-end frameworks typically bear little responsibility for presentation of front-end content, and any administrative interfaces that may be in use are often custom designed by the developer. In the case of purely front-end frameworks, responsibility is shared—

the framework creators need to ensure that the base of the framework is accessible, but developers that build using that framework need to ensure that they are using the framework in an accessible manner, including adding code as necessary to ensure that the resulting rendered output is accessible.

Certain platforms have earned a reputation for being more accessible than others. WordPress, for example, has a dedicated accessibility team (Make WordPress Accessible 2015) and Twitter Bootstrap has incorporated many accessibility components in recent releases, most notably 3.3.0 (Nixon 2014), which was released on October 29, 2014 (Twitter Bootstrap 2014). However, all frameworks are lacking in accessibility in one area or another, and all could stand to be improved in a variety of ways, especially in communicating the importance of assigning clear metadata to content editors at the time of content creation.

3.1.4 Accessibility of Third Party Software

In addition to the rise of SaaS solutions and frameworks in response to the increasingly complex landscape of web development, third party cloud services have arisen for everything from video hosting (e.g., YouTube, Vimeo) to image hosting (e.g., Imgur, CDN providers such as Akamai and Amazon) to document hosting (e.g., Issuu, Scribd) to authentication (e.g., log in with Twitter or Facebook) and even commenting on blog posts and articles (e.g., Disqus). There are obvious advantages to this paradigm—such as users being able to upload a raw video file to YouTube, and allowing Google to handle conversion and proper content type serving depending on the device, operating system, browser, and

available bandwidth that a user has when viewing the video, rather than requiring the content creator to be responsible for creating multiple video formats and selecting the appropriate one based on the capabilities of the client. However, the disadvantage is that it is easy to create inaccessible content on third party platforms, since those platforms are tailored for easy upload of user generated content, and do not prioritize forcing or encouraging users to add accessible metadata when they upload content.

If a content creator posts content to a platform such as YouTube, which is not fully accessible, it has the potential to reach a broader audience than if the video was hosted locally on the content owner's website. However, the content is not as accessible as it could be if it was hosted on the content owner's website, even if the content creator utilizes all of the available accessibility features of a platform like YouTube, such as captions and transcripts. According to one of my informants, organizations, such as federal agencies, that want to reach a broad audience, but are required to make content accessible, are often put in the situation of having to post content in two places—YouTube, to facilitate reaching a broader audience; and their own website, in order to ensure that the content is available in a location that is fully accessible. If third party platforms were made more accessible, and included direction to content creators on ways to make their content more accessible when uploading it, overall accessibility of user generated content would increase.

3.2 Policy and Legal Background

One of the reasons why people with impairments are an underserved community in cyberspace is due to the ambiguity of disability laws in the United States. The legal and regulatory framework is not explicit about the responsibility of corporations to make their websites accessible. As a consequence, many websites are not created to be accessible for people with impairments. Despite a memo from the U.S. Justice Department to Congress in 1996 declaring that the Americans with Disabilities Act (ADA) applies to websites (Patrick 1996), the Department declined to add specific language in this regard in the 2008 revisions to the law. At the time of writing, the Department is considering adding specific language to the ADA to include websites in the definition of “places of public accommodation” (U.S. Department of Justice 2010; U.S. Department of Justice 2014). The ADA mandates physical accessibility for “places of public accommodation,” which include government buildings, places of business, and certain organizations, but excludes private residences, and has more lenient regulations for small businesses. A change to the ADA to include websites would likely result in a similar distribution of responsibility.

Section 508 of the Rehabilitation Act of 1973 requires all organizations receiving federal funding to meet accessibility standards, and Section 508 compliance is enforced by federal government oversight. However, even Section 508 requirements have failed to ensure that all federally funded websites are compliant, such as the recently published case of the Obama administration's

recovery.gov site failing federally mandated accessibility measures (Grimes 2009).

Numerous lawsuits have been filed against corporations with inaccessible websites, often involving groups advocating for the rights of people with impairments. Precedent has been set for corporate responsibility for accessibility under a variety of laws from multiple countries. Among the prominent defendants were The Project Management Institute in 2007 (Pinsent Masons LLP 2007), Target Corporation in 2006 (Danielsen 2008), Priceline and Ramada in 2004 (New York State Office of the Attorney General 2004), and The Sydney Organizing Committee for the Olympic Games in 1999 (Clark 2002, Appendix A).

The United Nations Convention on the Rights of Persons with Disabilities (UNCRPD)—to which the United States is a signatory but did not ratify, indicating its support but not consent to be bound (United Nations 2015)—sets forth the most detailed and specific international guidelines governing disability and accessibility in a variety of contexts, specifically including “access, on an equal basis with others, to … information and communications, including information and communications technologies and systems … including the Internet” (United Nations 2006). The UNCRPD is explicitly legislatively prescriptive, but includes a provision for not creating an “undue burden” on the creators of such technology in its definition of a “reasonable accommodation,” which squares with similar language in the ADA in the United States intended to cover small businesses and the like:

> “'Reasonable accommodation' means necessary and appropriate modification and adjustments not imposing a disproportionate or undue burden, where needed in a particular case, to ensure to persons with disabilities the enjoyment or exercise on an equal basis with others of all human rights and fundamental freedoms.” (United Nations 2006)

The UNCRPD also specifically requires accessibility to be built into information systems at an early stage, instead of being retrofitted later at greater cost, and requires signatories to urge “private entities that provide services to the general public,” including the “mass media,” to provide such services in accessible way on the Internet (United Nations 2006). It is indicative of the United States' lukewarm commitment to genuine accessibility, particularly as it pertains to the regulation of private business entities, that the US merely signed the treaty and did not ratify it, thereby conferring limited legal status and a minimum of commitment.

3.3 Accessibility Standards and the W3C

The World Wide Web Consortium (W3C) recognized the need for inclusion of people with impairments on the web from the beginning, and formed the Web Accessibility Initiative (WAI) in 1997 (Dardailler 2009). WAI began working on a set of standards that would ultimately become the Web Content Accessibility Guidelines (WCAG) version 1.0, released in 1999. WCAG was intended to serve

as a guide for developers, designers, and content authors to assist with making various components of websites more accessible for people with various impairments (Chisholm, Vanderheiden, and Jacobs 1999). It is important to note that many of the categories of “disability” listed by the W3C often apply to the elderly, who are affected by (often multiple) impairments, but may not be “fully” blind, deaf, or cognitively “disabled” (W3C 2005).

WAI is made up of members of “industry, disability organizations, government, accessibility research organizations, and more” (Henry and Brewer 2005), and is intended to act on behalf of the disabled cyberspace community to advocate for standards and methodologies that will make the web more accessible. WAI encourages participation from stakeholders at all levels of technical ability. However, as I discovered in 2010 when I attempted to phone in to one of their teleconferences advertised on their website, participation is not as simple as just calling in—there is a process to become a member of a working group, which includes a required minimum of participation and service (WCAG Working Group 2015), which can serve as a barrier to inclusion for certain people including busy graduate students and those without corporate sponsorship (Clark 2006).

WCAG 2.0 was released in 2008 as the successor to WCAG 1.0. WCAG 2.0 attempted to improve upon WCAG 1.0 by addressing the challenges of more advanced technologies that had been developed since the late 90s—when WCAG 1.0 was released—as well as putting an emphasis on applicability to

future technologies, and making the standards more readable and amenable to automated conformance testing (Henry 2009). According to the WCAG 2.0, the four guiding principles that an accessible website must embody are "*perceivable, operable, understandable, and robust*" (W3C WAI 2009c). WAI defines these principles thus (W3C WAI 2015a):

> **Perceivable**—Information and user interface components must be presentable to users in ways they can perceive. This means that users must be able to perceive the information being presented (it can't be invisible to all of their senses)
>
> **Operable**—User interface components and navigation must be operable. This means that users must be able to operate the interface (the interface cannot require interaction that a user cannot perform)
>
> **Understandable**—Information and the operation of user interface must be understandable. This means that users must be able to understand the information as well as the operation of the user interface (the content or operation cannot be beyond their understanding)
>
> **Robust**—Content must be robust enough that it can be interpreted reliably by a wide variety of user

> agents, including assistive technologies. This means that users must be able to access the content as technologies advance (as technologies and user agents evolve, the content should remain accessible)

All of the standards in the document are tied to these four accessibility principles, and are given three levels of "success criteria"—A, the lowest, AA, and AAA, the highest. Conformance with level A ensures that users with impairments will have a basic level of access and usability to the system, but their experience will not necessarily approximate that of a non-impaired user, and they will likely have difficulty interacting with the website in certain ways. Level AAA, by contrast, ensures that users with impairments will be able to use the website on a similar level with users that do not have impairments. Additionally, WCAG 2.0 introduced the possibility of providing conforming alternative content (W3C WAI 2015b), which—according to one of my informants—is used by government agencies when they cross-post video material to both YouTube (partly accessible) and their own website (fully accessible).

Although there were important improvements to the standards in the way new and future technologies were treated, the shift to a focus on programmatic testability from 1.0 to 2.0 caused some outrage in the accessibility community, due to downplaying criteria that were not easily programmatically testable, such as criteria that apply to cognitive impairments in particular (Kliehm 2006).

> "If you slog through WCAG 2, you'll notice that even

> something as deceptively simple as that WCAG 1 guideline on clear and simple writing isn't there. Nor is there anything actually stronger than that guideline. In fact, there's nothing at all along those lines to be found in WCAG 2's Principle 3, "Content and controls must be understandable"... Based on my analysis and on presentations by Gian Sampson-Wild, it seems that dyslexics and others with cognitive disabilities have been sacrificed on the altar of testing." (Clark 2006)

The point that these authors are making is that there are limits to what can be tested programmatically. Certain criteria need to be evaluated by actual human beings, at least at the level of content creators with an eye toward creating a positive, accessible experience for the user—but ideally utilizing user groups inclusive of people with impairments. In the words of Karl Groves, an accessibility advocate, "give a damn about what you're doing, and make it usable. *Then* you'll be compliant" (Groves 2015). Even the WCAG 2.0 document itself acknowledges this limitation:

> "Note that even content that conforms at the highest level (AAA) will not be accessible to individuals with all types, degrees, or combinations of disability, particularly in the cognitive language and learning

> areas. Authors are encouraged to consider the full range of techniques, including the advisory techniques, as well as to seek relevant advice about current best practice to ensure that Web content is accessible, as far as possible, to this community." (W3C WAI 2009c)

WCAG 2.0 has been adopted as an ISO standard—ISO/IEC 40500:2012 (International Organization for Standardization 2012) as well as overlapping with the U.S. Section 508 requirements for levels A and AA almost exactly, and in some cases providing more specificity or addressing a deficiency in the 508 requirements (United States Access Board 2015). At this time, despite its flaws, WCAG 2.0 is considered the main international web accessibility standard.

3.4 Conclusion

All of this technical detail is the long way around of demonstrating how far the discipline of web design and development has come since the Mosaic graphical web browser was released in 1993. The knowledge requirements for web designers and developers in 2016 are significantly greater than what was required two decades ago, and this complexity is poised to increase. So how can web designers and developers make the websites they produce accessible given the incredible complexity they are already negotiating just to produce a website that works for non-impaired users?

A further complicating factor is that most web developers do not have a

deep understanding of all of the technologies that they are using. A developer may use a CMS such as WordPress, a front-end framework such as Twitter Bootstrap, and a JavaScript framework such as React without ever knowing the intimate details of how those pieces of software work. Although most of the platforms that most web developers use are open source, the investment of labor time to fully understand the workings of each is beyond what most developers are able to accommodate while continuing to produce websites.

When making a website was a relatively simple matter of writing HTML and using limited multimedia, including accessibility components was relatively simple. Now that web developers are utilizing tools and frameworks in order to produce complex websites quickly and reliably, they are increasingly relying on those frameworks to provide well-written and performant code, and have offloaded some of their own responsibility in that regard. Therefore, if the frameworks and tools that web developers utilize on a routine basis are not accessible, and if the accessibility features that are built into these frameworks and tools are not transparently understandable and usable, it will be difficult or impossible even for a conscientious web developer to make an accessible website.

4. ROUTINE DESIGN & DEVELOPMENT PRACTICES

In practice, design and development practices vary wildly from organization to organization, and even from person to person. There are some commonalities—some of which can be grouped according to the type of industry or business model a particular individual operates under—but there are often more differences than similarities. The practice of writing code for websites is often more of a solitary activity than the steps leading up to it, such as requirements gathering and collaborating on strategy and design elements, and is thus more susceptible to the individual preferences of the person writing the code.

When I speak of routine design and development practices, I am talking about two things:

1. **Routine practices at an individual level**—the typical process an individual (in this case, my informants) will follow when building a website, irrespective of other factors, which is relatively immutable—unless the project they are working on is extraordinary in some way.
2. **Routine practices of a particular organization type**—the typical process followed by organizations of a certain type (e.g., freelancers, dev shops, federal government contractors, etc) will follow—which will vary from organization to organization, but can be loosely grouped together.

I posit that, at an individual level, if the developer has been trained about accessibility and takes it seriously, they will often include components of

accessibility in all of their projects. Likewise, if they are not trained about accessibility or view it as an optional extra, they will not include it in their projects unless explicitly required to do so by the client. Organizations are similar—the only organizations that routinely apply accessibility principles are those that work in industries where accessibility is mandated, such as federal government contractors, or people working in the healthcare space that are dealing with audiences that are likely to have a certain type of impairment. This is what ableism looks like in practice.

All of the people that I spoke with loosely follow some form of the systems development life cycle (SDLC) methodology. The SDLC has distinct phases, which my informants discussed as being represented in some way in their own work practices and the practices of their organization. These phases are:

1. **Planning**—where initial discussions occur around the need to build or modify a system in some way, in response to a business need. For example, “we need a website to provide information about a new product that we are launching.” This phase typically happens with the client, which often occurs before engagement with most of the people that I spoke with.
2. **Analysis**—where information is gathered about what components should be included in the system and how they should function. This information should be solicited from users whenever possible or feasible. However, if the change is primarily related to business needs and goals, then users are often not solicited for feedback at this stage. Modern tools and

technologies like Google Analytics and social listening often serve as a stand-in for direct user engagement.

3. **Design**—where the website or change to a website is laid out visually, but not programmed. Sometimes this is accomplished using a visual design program like Adobe Photoshop, but design can also be accomplished directly in the browser using code. The design phase may include multiple iterations of a design, and may begin with a wireframing stage to explore functionality and general object placement prior to visual design.
4. **Implementation**—where most of the actual code writing takes place. This phase is where the design—in whatever form it takes—is translated into a built website by a developer. This phase also includes any user testing that may be conducted, as well as an audit of the code prior to deployment, and final deployment of the code to a production system.
5. **Maintenance**—where a production system is observed and kept running. This phase often leads back into another round of SDLC when an additional need is identified that would modify the system in some way. This phase also includes application of security patches and system upgrades, monitoring the health of the website, conducting ongoing analytics gathering and analysis, and the like.

There are several modern workflows that grew out of the basis of the SDLC that are adopted by development teams, agencies, and even individual developers. One of the most popular of these new methodologies is called agile

software development, typically referred to as just “agile”:

> **Manifesto for Agile Software Development**
>
> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
>
> **Individuals and interactions** over processes and tools
>
> **Working software** over comprehensive documentation
>
> **Customer collaboration** over contract negotiation
>
> **Responding to change** over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more. (Beck et al. 2001).

Agile places a high value on shipping code and being flexible. Unlike the “waterfall” methodology that had been typically used in software development during the 20th century, agile software development involves rapid iteration over discrete subtasks that can be developed, tested, approved, and pushed live quickly, rather than trying to build, test, and launch an entire system all at once. This iterative approach allows for a more gradual evolution to a system or website, and allows for flexibility to respond to changes in user needs and system performance. However, it can be argued that the emphasis on shipping

code quickly is at odds with ensuring that systems are fully user tested before being put into production, and introduces an ableist emphasizes the role of the developer over that of the usability tester.

The informants that provided the information for this dissertation through interviews and survey data can be loosely categorized into several groups:

- **Freelancers**—individuals working by themselves, hired by companies or other individuals on a per-project basis, who may work loosely or closely with a project team, but only for the duration of the engagement.
- **Dev shops**—typically small groups, primarily composed of developers, but often including at least one designer, that work specifically on web development projects, which are often faster and cheaper than a full-service agency, but often have less depth of expertise than a full-service agency.
- **Full-service agencies**—marketing or advertising agencies that offer a range of services—including design, copywriting, editing, development, project management, digital strategy, keyword research, analytics, and the like—which build websites for clients according to client needs, best practices, research, and a robust internal review process. Typically the most expensive out of all arrangements due to overhead, the number of staff involved in each project, the rigorous review process, and high hourly rates.

- **Direct employees**—in some cases, development of a website is not outsourced to an external group, but is done in-house. Sometimes this occurs with enterprise environments, in which employees are creating custom sites specifically for their employer. In other cases, this includes people that work for companies like Facebook and Google, where they are making websites for their company that will be used by other people.

Routine design and development practices are more homogenous within these groups than between them, which provides a useful lens for analysis. I will now step through the phases of the SDLC and discuss how these phases are typically implemented in the different types of development groups, and to what extent accessibility is included as a routine practice in each.

4.1 Planning

Nearly everyone I spoke to identified the planning stage as one that takes place almost exclusively with the business owner. (In this context, “business owner” is not necessarily the person that literally owns the business, but is rather the person that is responsible for representing the needs of the business in the context of the project.) My informants indicated that the need for a website, or an enhancement or modification to a website, is almost exclusively a response to a particular business need—“we need more traffic to the website,” “we need more people to buy our product,” “we need a website to announce a new product we are launching,” and so on:

> “I think they are mostly business requirements. I think a lot of people wait until that soft launch—actually, it depends. If it’s a new startup, then yeah, usually the people are experts in the field that they are making this thing for. Because they don’t have any existing customers, or anything like that, they’ve had casual conversations with people, but they haven’t formally interviewed them, or anything like that. So they have a good sense of what they think the market needs, and they base the requirements off of those.”
>
> —Informant N

In some cases, the need is identified through engagement with users, and is a response to user feedback. However, those cases almost exclusively existed with informants that worked for companies that provided a product for users, and not with dev shops or agencies that were building websites for other companies. For example, when I asked one of my informants under what circumstances requirements came from end users, she answered:

> “I think it’s more in sites that have that user component, so something where you would have to sign up, or manage an account, or pay for something, or sign up for an event.”
>
> —Informant H

Likewise, companies that operate help desks for customers using their products often get feature and functionality requests through that venue:

> “Well, I don’t know how social it is, but we do get a lot of feedback from customers through our customer service department. One of the main reasons why we are updating our website is because the marketing department has been fighting for it for a long time, and we have started to back it up with complaints of poor navigation and whatnot from customers over the phone. That has really helped us give feedback. That is really where most of our feedback comes from.”
>
> —Informant L

Customer service calls can also be used as a mechanism to push back against developers or other members of the organization that are against making changes to the system for one reason or another:

> “Customer service is another great place, where they would say, “we get a hundred calls a day about this. Can you make it less stupid?” If it’s broken, fix it, or just make it more usable, because no one gets it. I don’t care, engineers. Use your engineering words all you want. Our customers don’t get it. Make it better.”
>
> —Informant Q

The majority of my informants identified the planning phase as being relatively opaque to them, and the results of the planning phase as being mostly unchangeable, even if the request would represent what they perceived as a negative user experience or would be detrimental to the site or users in some way. For example:

> “Our navigation has every single subcategory of every single subcategory of every single category listed in a vertical accordion on the left-hand side, and they aren’t in alphabetical order. They are in order deemed important by superiors, so no one can find anything, essentially.”
>
> —Informant L

The principles of user-centered design were often not in play, with websites being created and modified primarily according to business needs instead of in response to the needs of users as expressed by users themselves. Given that the identified business needs typically did not include an explicit commitment to accessibility, and the decisions about what to include in the project were typically locked after they came from management, this was yet another example of ableism in practice.

4.2 Analysis

The analysis phase is largely differentiated by whether the client and/or the developer has access to user preference data prior to beginning a project,

which usually only happens if there is already a product or service in the market that the company is responding to—which could be an existing website that is being modified or enhanced in response to user feedback and preferences, or it could be a website that is being created to serve a particular need, as identified by research into users' expressed preferences and desires.

All of the informants that I spoke to that build websites for other people—agencies, freelancers, dev shops—indicated that any kind of user feedback or requirements analysis came directly from the company that hired them, and the process by which that information was gathered was relatively opaque. In some cases, information about what users want was available via feedback mechanisms such as email, but more often than not, the requirements came from the business owner to serve a particular business need, such as driving more sales.

The notable exception to this routine was a type of stand-in facilitated by social media called “social listening.” Social listening happens when a website development group is able to leverage the stories that people tell on social media as a stand-in for direct user engagement. There is a rich data set of information, coming from users, that is available online on platforms like Twitter and Facebook, and in reviews on websites like Yelp and Google. By reading what users of these platforms are saying about a particular product or service—or the need for a particular product or service—website developers are able to get information about user needs without having to form a focus group, which can be

costly in both money and time. Social listening is far from perfect, but is a form of "guerrilla" requirements analysis that small shops working under small budgets can leverage in order to obtain information about user preferences and needs:

> "I'm starting to talk to clients about doing a hybrid of social listening for a period of time and surveying, and if they can afford it, focus groups. But, you know, it depends a lot on their budget. Because at the end of the day, with a lot of these clients, they—you know, if their whole budget is twenty grand, and they are doing all this listening, and focus groups, and it's going to eat up fifteen grand, then they don't have anything to spend on executing what we find."
>
> —Informant C

In the cases where an informant worked for a company that was making its own website as a service to users, feedback was often more direct. In most cases, the product had existed for a while, and there were definite mechanisms in place for soliciting and responding to user feedback. For instance, some of my informants were developing products that had an end user base of registered users that were paying a monthly fee for access to the service, and had access to a help desk if they had questions or problems with the website software. Many of the change requests for the website came in through the help desk, which made it easier to attach a business need to the change, since paying users were

complaining about a particular feature not working well, or needing improvement.

Very few of my informants took advantage of a full-fledged alpha or beta testing program, and instead relied on tests and feedback performed immediately after launch as a mechanism for course correction in the build. As a result, updates to such sites would involve an analysis phase that began immediately after the launch of an update to determine what changes needed to be made in response to that update. For example, if the update made another section of the website difficult or impossible to use, or if there were problems with the update itself, users that were already on deck to perform testing and provide feedback as part of an incentivized early feedback program could provide valuable information to the development group as soon as possible after the update was made. Although this technique is not as effective as an alpha or beta test with users prior to launch, it does represent an effort on the part of the development group and the business owner to collect user feedback and respond to it.

One of my informants works as a UX researcher for a medium-size software company, and is responsible for conducting user interviews and participant observation to determine what aspects of an interface are working well, and what needs to be changed. He practices what he calls “guerrilla” or “ad hoc” usability testing, which can be performed inexpensively using tools that are cheap or free, such as screen sharing software. Participants can be solicited either from existing user registrations, or through websites like Craig's List, Twitter, or Facebook. A large amount of information about how users are

interacting with a site can be gleaned even from a short session talking to and observing a user, which can provide valuable insights into how to better structure a site for usability.

> “Ad hoc usability testing is a lot easier than people realize, you know? It’s not that hard to do a WebEx to show your website or your software to somebody. There’s usertesting.com, which allows you to let them do it in an automated fashion, basically. You could set up a test, and they will go find people for you. It’s not that expensive. There are, if people do just a minimum of research, they could find a bunch of guerrilla ways to do research, or test existing designs or software, that don’t cost an arm and a leg, don’t require dedicated staff, don’t require people that are trained.”
>
> —Informant P

Most of the informants I spoke to do not incorporate accessibility into their projects unless it is a specific requirement from the client. There are some exceptions to this, which I will discuss in greater detail in chapter 7, which mostly have to do with overlap between accessibility and other concerns, such as SEO. Additionally, there are some aspects of accessibility that are considered best practices in web development, such as providing “alt” text for images and using

text instead of images whenever possible, which were practiced by a majority of my informants.

Clients that mandated accessibility typically fell into one of two categories:

1. **U.S. federal government agencies.** People who were doing work for federal government agencies were required to make their websites accessible by law. Section 508 of the Rehabilitation Act requires that all federal government websites, and the websites of all organizations receiving federal money, be accessible for people with impairments.
2. **The expected user base can be assumed to have impairments.** Some of my informants worked for clients—especially in the healthcare space—that were making websites targeted at groups of people that could be expected to have particular types of impairments. For example, one of my informants worked on a website targeted at people with macular degeneration, so extra care needed to be taken to make sure that the website was functional for people with impaired vision. However, in these cases, only the specific anticipated impairment of the user base is considered when building in accessibility components—additional work is not performed to make the website accessible for people with other types of impairments.

Based on the responses from my informants, the requirements analysis phase uses feedback from users in a startlingly low number of cases. Typically, user preferences and feedback are incorporated only when an established and

identified user base is already interacting with the software. Social listening is used as a stand-in for direct user engagement when direct user engagement is impossible or not feasible. However, in the majority of cases, the requirements for the website come from the business owner to fulfill specific business needs, instead of practicing the principles of user centered design, introducing ableist bias against users with impairments that were not considered by the business owner when creating the requirements for the system.

4.3 Design

The design phase is a step along the process to the final website build. In most cases, much of what is done during the design phase is an interim step to get to the build, and is generally thrown out after the design phase is done. One of the trends in modern website development involves how to minimize the amount of re-work and wasted effort involved in the design process by adopting new methodologies of design, such as designing in the browser using code, instead of designing in a dedicated visual program like Photoshop. The extent to which this can be accomplished is heavily dependent upon the skill and experience level of the designer in working with code and web native tools, and the appetite of the organization for changing the routine process by which design happens.

4.3.1 Wireframing

On larger projects, wireframing is performed in advance of design.

Wireframing is a somewhat contested practice in the field of web design and development. Widely considered to be a best practice, wireframing requires extra time and effort to be invested up front in a project to lay out what elements will be on a website and how they will work together. Typically, wireframes are done in greyscale with placeholder images and Latin text, so that the wireframe reviewers can focus on the placement and order of elements on the page, instead of being distracted by the specifics of photography, typography, colors, text, or other elements that are part of the full design. If done properly, wireframing allows all stakeholders to ensure that their desires for the website are being addressed structurally at an early phase of the project, which makes the remainder of the design and development process smoother.

> “Because I’m very clear about what wireframes are, and how they should be viewed, I’m always very careful to make sure that they know that this is like the blueprint for a house. They shouldn’t be taking any of the copy or any of the look and feel as being as they see it. It’s all about making sure that all of the elements are there, and that the prioritization of the elements is there, and the functionality and the UI.”
>
> —Informant J

However, the initial time investment required to make the wireframes is sometimes a difficult sell to clients and project managers, and designers can

often feel like their options for design are constrained by the decisions made at the wireframing stage. Additionally, many clients and internal members of web development teams are highly visual people that are unable to fully comprehend what is being communicated in a wireframe due to a desire to see a fully-baked design before providing comments. In such circumstances, the wireframing stage can represent a waste of time and resources if it does not serve its intended purpose as a bellwether of a good design and layout.

> “I’ve tried wireframes before on bigger sites. It is a budgetary thing for smaller sites. But bigger sites, I’ve found that clients just don’t get it. So you send them a wireframe, and they’re like, okay, it just looks like someone drew something on a napkin. So they want to see something visual, and in their face.”
>
> —Informant I

If executed properly, wireframes are able to expose usability problems in an early phase of the project, where they can more easily be identified and resolved before too much time is invested in the final site design and development. This is especially true when wireframes are built using code, instead of designed as flat assets in a layout program like Photoshop or Illustrator. By allowing team members—and potentially a sample group of end users—to interact with a prototype of the website, usability and accessibility issues can be identified and resolved early. Unfortunately, none of my informants

used the wireframing process to communicate design ideas to end users, and very few of them used the wireframing process to evaluate designs with an internal team.

4.3.2 Design Comps

After the wireframing stage is complete, the full design comps phase begins. The design comps phase is where colors, typography, images, and even some copy is laid out for review. Often, design comps are produced in a static layout program like Photoshop or Illustrator. Sometimes, design comps are produced using native technologies in the browser, which more easily translates into final website code. The advantage of static design comps produced in programs like Photoshop or Illustrator is that they are comparatively easier and faster to create and modify, but a developer must write all of the code to translate a static design comp into a website build after the design phase is over, effectively throwing out all of the work that was done on the design afterwards. Designing in the browser takes a longer amount of time up front, but more easily translates into the final website build, since most or all of the code is written by the time the design comp is finished. Which approach a particular team will take depends on the comfort level of the designer with writing code, and the anticipated quantity and scope of revisions from the client after the initial design. If the team anticipates that the client will reject large portions of a design during the first or second reviews, they may opt for static design comps to reduce the amount of time involved with modifying the designs to meet the client's needs. If

the team anticipates that the client will have very few changes to the design, they may opt for coded design comps to ease the transition into build.

> “They definitely use Photoshop for parts of [the design]. We started that way, when I started a year and a half ago, but quickly moved to style guides, just because it’s quicker. They are the ones doing the build anyway, so their code is theirs, and they can own that, and make sure that it works from the start.”
>
> —Informant H

Design comps will be reviewed by the internal team, including the developer, copywriter, account or client liaison, digital strategist, and UX engineer, when these roles exist on a particular project team. The design comps will also be reviewed by the client or business owner, typically after approval by the internal team. In some cases, design comps are shared with potential end users, but more often than not my informants indicated that end users did not see a design until it had transitioned into build.

4.4 Implementation

4.4.1 Development

The development phase is when the design is transitioned into a built website. If the designs were produced using a layout program like Photoshop or Illustrator, the developer will be responsible for examining the design files for

elements like color usage, font choice, and image selection, and translating the designer's vision into a built website. If the design was built using code, the developer will need to incorporate the code from the design phase into the final build, which may involve incorporating static code into a dynamic platform like a content management system (CMS) or a database.

At the outset of a project, the developers will work with other stakeholders on the project to determine what technologies will be used in the build, such as which CMS platform will be used (if any) and what technologies will be available and used on the production server. Typically, development groups have a preferred technology stack and approach, which may change in response to specific client or project needs. For instance, if a particular client requires that a website is hosted on their servers, the development group will need to choose a technology stack that is compatible with the client's servers and infrastructure. In some cases, the specific technology stack is dictated by the client. In others, it is entirely up to the developer to determine which technology stack and toolchain is most appropriate for the project.

Any accessibility components that require programming will need to be incorporated during the development phase. This includes things like “alt” text on images, so that screen readers can understand the content and purpose of images in the design, as well as ARIA roles and schema.org markup, which is particularly important when making interactive components accessible.

“[Accessibility and SEO] is becoming part of the

> standard operating procedure in terms of making those things. It's almost getting to where we don't even think about it now, you just do it. It's becoming part of the standard. I think most agencies, and even most development shops, start to build up their own toolchains. Doing things like that, like having the standard templates and things, which have things in the right order, which is going to help you with that anyway. Like you said, having it run through the linters, and things like that, which is part of the standard build chains that we have as well now, makes it so that you don't mess up stuff like that."
>
> —Informant N

If the developer is using a front-end framework like Twitter Bootstrap, this may be made easier by inclusion of accessibility features in the platform itself. Certain CMS platforms also make inclusion of accessibility features easier, such as WordPress' automatic insertion of "alt" text on embedded images reflective of the descriptive text entered when an image is uploaded to the system.

> "I think that WordPress makes it pretty damn easy to be accessible."
>
> —Informant L

Automatic insertion of "alt" text and other forms of out-of-the-box

accessibility requiring little to no additional work on the part of a developer or content creator are examples of what Jim Johnson terms a “nonhuman delegate” similar to an automated door-closer, utilized when human behavior is unreliable, or when automation is preferable to discipline.

> This is where the age-old choice, so well analyzed by Mumford (1966), is offered to you: either to discipline the people or to *substitute* for the unreliable people another *delegated human character* whose only function is to open and close the door. (Johnson 1988, 300)

However, there are problems with over-reliance on automation. WordPress will automatically insert alternate text to describe images, but by default, it uses the file name of the image minus the filename extension as the alt text. If the name of the image isn't descriptive, and if the content creator does not input descriptive alternate text to replace the default, then the image will not be more accessible than it would have been had the alternate text not been inserted by WordPress at all. There is a danger that relying on automated accessibility techniques provides a false sense of having met an accessibility need, when in fact, the need may not have been met at all.

Most of my informants described a high level of integration between the development process and other tasks required for the website build. Contrary to the waterfall methodology that was popular in the 20th century, agile software

development methodologies and other forms of collaborative development require involvement from developers at every stage of the process. Developers will have an opportunity to review and comment on wireframes as well as design comps, to ensure that designers are not, in effect, promising functionality that is difficult or undesirable to program into the final build. Likewise, designers and other stakeholders typically have several opportunities to review a build as it is being developed, so that problems can be identified and resolved as early as possible.

> “But the general process would be first to start at the ideation phase, where we’re thinking about the idea and how it might work, and then we start doing some design prototyping, and also some web prototyping, depending on what the feature is. More recently, when we refreshed the site in June, prior to that we had the designer work on some new designs for updates to the navigation. Before we dove into development, we decided to prototype a little, so we could spend some time with them both on the desktop and on a mobile device, and see if it was really the direction we wanted to head in.”
>
> —Informant F

Depending on the type of project, type of web development group

performing the work, and the client requirements, there is a range of possible levels of development required and practiced for a particular build. At one extreme, the web development group will not write any code whatsoever, and will simply work within an existing platform such as Facebook pages, Squarespace, or WordPress.com. At the other extreme, the web development group will write all code from scratch. In most cases, however, the web development group will write some custom code and leverage some pre-built code, platforms, or frameworks to assist with the build. Once the build is complete, and functions to the satisfaction of the developer, several stages of tests are typically performed.

4.4.2 Testing

There are several different types of testing that are performed during and after development—and in certain paradigms, even before development begins. Broadly speaking, testing can be broken down into three types:

1. **Functional testing**—in which the functionality of the website is tested against explicitly or implicitly defined functional requirements, which may take the form of a formal functional requirements document, or in agile software development, a user story. An example of a functional test would be: “When the user fills out the form and clicks submit, the record is saved in the database and an email alert is sent to the business owner.” Functional testing may be accomplished in an automated fashion using unit tests, which are programmatic tests designed to evaluate the correct operation of a discrete part, or unit, of an application. If such tests are

written before the corresponding code is created to pass the test, then the developer is said to be practicing “test driven development,” or TDD.

2. **Visual or layout testing**—in which the design of the website is verified to be accurate across different operating systems, browsers, and devices at different screen resolutions. For example, ensuring that a website that was previewed by the developer in Google Chrome also works properly in Internet Explorer, Firefox, and Safari on Windows, Mac OS X, tablets, and smartphones.
3. **Conformance testing**—in which the website is evaluated against one or more standards, which may or may not be programmatically verifiable. Accessibility testing falls under this umbrella, as does W3C standards compliance for HTML and CSS, as well as code standards compliance at an organizational level. Conformance testing also includes testing for any legal or regulatory requirements that may be in effect for a particular project. Conformance testing for adherence to best practices is often accomplished automatically during the build using automated tools known as “linters” or “hinters.”

Testing ultimately involves all members of the team, including the client or business owner. However, all testing begins with the developer. Developers will often use linters or hinters as part of their typical build workflow in order to check their work as they go, to help reduce the possibility of errors making their way into subsequent testing phases. Linters and hinters designed to check for

conformance to HTML language standards will often warn about certain accessibility problems proactively:

> “Obviously, we do run it through linters, and a lot of that will pick up the fact that you’ve missed the alt tag, and the title tag, and stuff like that, because it’s part of the web standards now, to have those things.”
>
> —Informant N

The development team is also typically responsible for performing browser and device testing, at least initially. If there is an internal or external QA group that is responsible for performing more in-depth tests, that group would typically be responsible for performing browser and device testing, but such QA groups typically only exist in larger organizations or on projects with bigger budgets. Likewise, accessibility testing, when it is practiced, is initially performed by the developers, and potentially performed by a QA group. A common tactic for ongoing internal review, where the product being produced is something that could be or will be used by the internal team after launch, is a practice known as “dogfooding.”

> “Dogfooding is the intentional consumption of your own product or toolset.”
>
> —Informant E

Once the developers have completed their testing, the rest of the internal team will typically perform a review. What each team member is looking for in a

review will vary depending on role, and the type of group that is executing the build. Under the agency model, the account executive is responsible for ensuring that the website meets the promises made to the client in the statement of work, the editor is responsible for ensuring that there are no spelling or grammatical errors on the website, the designer is responsible for ensuring that their design was faithfully executed, and the strategist is responsible for ensuring that the website is optimized for search, interfaces with all required external services, and is accurately reporting analytics metrics back to the analytics tool. In cases where websites are built by organizations that are not agencies, these roles are filled by other individuals, but are typically covered in some fashion.

Once the internal team is finished reviewing the site, they will pass it to the client for review. In cases where the development team and the client are part of the same organization, this process really represents an additional round of internal review before releasing the software to the end-users. In cases where the developers are making the website for an external client, the client will be given a staging link to a version of the site that is only accessible to a limited group of people. Once the client has performed their review and provided feedback, there are typically several rounds of revision before the website is approved and pushed live.

Most of my informants did not use user groups for testing, and those that did used whomever was available and willing to serve, instead of taking specific care to craft a user group out of diverse individuals.

> *How were [the user testing] subjects recruited?*
>
> "The student services office took care of it. They just kind of put out feelers for anybody that wants to come get free pizza."
>
> *So it was hunger motivated.*
>
> "Right. It was not a selective process by any means."
>
> —Informant I

Often, user groups are assembled in response to a specific need. For instance, user groups can be assembled by collecting specific users that have been identified by server logs or support requests as having experienced a particular problem with the website, and then contacted directly:

> "Currently, the best way that I have found to do that is to identify only the customers who have engaged with the thing that I'd like to test. So, for example, for a while we were having an abnormally high number of login failures for our mobile apps, and I went through and did what I think of as like a—I did an analog scrape. So I went into all of the support requests that were submitted from the logon failure screen, and then emailed them."
>
> —Informant E

There are three main forms of user testing that are performed on

websites, and none are mutually exclusive:

1. **A/B testing**—in which a subset of users are programmatically and randomly served a version of the website that is slightly different in a specific way that is being targeted for analysis. For example, the primary color of the website might be different, or the position or size of a “Buy Now” button might change. User engagement is tracked and measured against an intended outcome, such as the number of people who signed up for a service, or purchased a product—what is known in the marketing industry as a “conversion.” If a statistically significant number of users that were shown the alternate version of the page convert at a higher rate, then the change will be adopted into the primary view of the website, and the A/B test will be discontinued. A/B testing is often performed on live websites after launch, and is used as a tactic to make minor adjustments to a website to fine-tune conversion performance over time by testing hypotheses about user behavior related to aspects of site design.
2. **Beta testing**—in which a certain subset of end-users are given access to a new version of the software ahead of the general user base. In some cases, users opt in to a beta testing program, which may involve rewards for participation. In other cases, beta testers may be unaware that they are part of the beta testing group, having been selected either at random or because of a particular characteristic, such as belonging to a particular organization or demographic.

3. **Post-launch review**—in which testing occurs after the website has been pushed live for the entire user base and/or general public to view, but a group of users are identified to test the site very shortly after launch and provide early feedback. For groups that have limited or nonexistent user testing budgets, or do not have dedicated environments for beta testing or software set up for A/B testing, post-launch review is a common tactic. In some cases, post-launch review happens as part of a "soft launch," in which a website is launched quietly, without advertising, in order for a small group of users to interact with the site and provide feedback before a large influx of new users discovers the existence of the site.

The amount and extent to which these types of user tests are performed depends heavily on project budget and time constraints. If there is not enough budget for testing, or if the timeline is too tight to allow for a testing cycle, testing steps will be skipped, and left to eventual end users of the live site to report issues through whatever means are available to them. Most of my informants did not leverage user testing prior to deployment, and virtually none conducted explicit accessibility testing prior to launch, except where it was required by the client as part of the website deliverable. As a consequence, most websites are launched without verifying whether or not they are accessible by utilizing user testing including users with impairments, even if the developers put time and energy into meeting published accessibility standards. Therefore, it is not only possible, but exceedingly likely, that these websites include accessibility barriers

that are unknown to the developers, which can only be reported and (potentially) remediated once the website is live to the world.

4.4.3 Deployment

Once all of the stakeholders involved in the project have reviewed and approved the built website, it needs to be pushed live. This process involves copying files, and potentially a database and configuration information, to a production server, and may involve making configuration changes in other systems to point at the new or updated website. Once everything has been pushed live, stakeholders will typically perform an additional round of review on the live site to ensure that everything that they reviewed and approved during development is represented on the final website.

After launch, a process of mutual observation begins. Users observe and make use of the website, and may provide direct feedback on their experiences through a contact form, email, phone, help desk ticket, or social media. Simultaneously, the developers observe the behavior of the users through analyzing traffic data, analytics events, and sometimes more advanced behavior observation techniques such as heat mapping, which tracks mouse movement, or eye tracking, which allows development groups to see what the focus areas are within pages of a website.

> “What we do do is mouse tracking. Hotjar is a really cool mouse tracking platform, where it will track every single user that comes to the site, and you have

> thousands of videos on your dashboard, and we can see them."
>
> —Informant S

When I asked my informants about whether they had received post-launch feedback specifically about accessibility, most of them answered that they had not. This response was fairly typical:

> "I wouldn't say that we've gotten feedback related to accessibility. I feel like most of the feedback has been related to design decisions that we've made on the site, like maybe removing a certain navigational element. The majority of the feedback is related to that, or maybe some bug, something that's not working."
>
> —Informant F

Perhaps the sites made by the people I interviewed do not have major accessibility barriers, and therefore accessibility issues aren't being reported because they don't exist in a glaring way. Or, perhaps accessibility issues are being reported, but are not being flagged by the reporters or understood by the website developers as being accessibility issues, and are instead lumped into the category of general usability.

However, I think it is more likely that users with impairments that encounter accessibility challenges will leave an inaccessible site silently, rather

than taking the time to report the accessibility issue to the website developers. Therefore, development groups that rely on post-launch issue reporting by end users that aren't directly engaged by the development group to perform a review are likely missing out on important feedback that could be gathered by intentionally including users with impairments in the testing process.

Depending on the relationship of the developer to the business owner, the development engagement could end when the website is pushed live, such as when the developer makes websites on a contract basis. In other cases, there may be a long-term maintenance relationship with the live website, which may involve creating and launching new features for the website over time.

4.5 Maintenance

The maintenance phase involves monitoring the website over time, ensuring that it remains available and continues to perform its intended function. Typically, website maintenance involves applying security patches to the system hosting the website, as well as the software running the website, including the content management system and any installed plugins. The maintenance phase also includes responding to bug reports filed by users, and may involve small changes to the website code in order to fix issues that have been identified once the website is live.

One of my informants, who works as an accessibility educator and consultant, indicated that accessibility evolves over time, and needs to be

considered as part of the maintenance phase. In some cases, changes in the way assistive technologies work, particularly with the release of new devices, necessitates periodic re-testing of the accessibility of a website. Additionally, if the content that was on the site at launch was certified as accessible through testing, and new content was added after launch, the new content will need to be reviewed as well. Ideally, new content is reviewed for accessibility as it is developed and before being added to the live site. However, testing after a site goes live tends to be less rigorous or structured than during the initial approval process.

4.6 Conclusion

The majority of the informants I spoke to did not develop websites with accessibility specifically in mind. Most of my informants included at least a basic level of accessibility, where such practices were considered part of best practices in web development in general, such as applying alt text to images and using actual text in HTML instead of text embedded in an image, which would be inaccessible to a screen reader. However, the routine development practices for the people I interviewed did not typically include a robust commitment to designing and developing for accessibility generally, or testing with diverse user groups that included users with impairments, which is in keeping with ableist philosophy about routine practices of production more generally. Most often, the reason why accessibility was not included was because of time or budget constraints combined with the fact that the client didn't value accessibility.

> “So, I don’t think a lot of clients, no matter how much you tell them that accessibility matters, really care about accessibility matters, unless you can present them with how it’s going to affect their bottom line.”
>
> —Informant I

Unfortunately, accessibility is often viewed through an ableist lens as an optional extra—one which can be safely ignored without negatively affecting the project. Despite economic and legal arguments to the contrary, most development groups and clients do not value accessibility as a core component of what is required for a website to be considered complete enough to launch.

Within categories of accessibility, visual accessibility was the category that was given the most amount of attention by my informants, even those that did not focus on accessibility in the build—ensuring that font size was large enough to read, that color contrast was high enough to be easily readable, and so on. Screen readers were also emphasized to some degree, through a focus on presenting website content as actual text instead of embedded in images, and through writing alt text where images were present. However, accessibility addressing other forms of impairment was largely not practiced by my informants, or valued by their clients. The one notable exception to this paradigm was when someone on the project team was what I call an “accessibility advocate”—someone that pushes the rest of the team to consider and include accessibility in a project, even when not explicitly required by the client or

business owner—a subject I will return to in more detail in chapter 5.

5. POWER RELATIONS

When I began this project, I had a hypothesis that a general commitment to accessibility would be present in the people that I interviewed—that web development groups, left to their own devices, would prefer to produce websites with at least a basic level of accessibility. My hypothesis was that three things would interfere with this goal—budget, timeframe, and specific members of project teams exerting disproportionate influence over the project, such as an account manager making design decisions instead of leaving the design decisions to the designer.

The budget argument is straightforward. Projects with limited financial resources must restrict or give up certain things entirely in order to be successful, including some that are considered industry best practices. Focus groups, user testing, and accessibility tend to be some of the first best practices to be cut from a project due to budget constraints.

Exclusion of best practices due to timeframes is less straightforward than exclusion due to budget. Unlike budgetary exclusions, timeframe requirements can affect specific tasks within the design and development process that involve high latency—even if they aren't particularly expensive—such as soliciting user feedback in the form of surveys. When a project needs to be delivered by a certain date—typically driven by an event external to the project, such as ensuring that a website launch is timed with the release announcement for a new product—steps that take a significant amount of time, or which are difficult to

precisely control, end up getting cut. Again, focus groups and user testing tend to be time-consuming, and tend to serve as blockers for other parts of the design and development process, and are not directly under the control of the designers and developers, since they depend on the availability of third parties. Accessibility also tends to get cut when timeframes are a concern, given that accessibility, viewed through an ableist lens, is often considered to be an "optional extra," when not required by the business owner explicitly. Therefore, accessibility is considered to be something that must be "added on" later in the development process, instead of being viewed as an integral component of development, and therefore is ripe for exclusion when trying to cut down on build time in project schedules.

Nearly everyone that I spoke to identified accessibility as being part of web development best practices, and as an ideal to strive for, but indicated that it is often overlooked or cut from projects due to budget and time constraints. This narrative was expected and unsurprising given both my own personal experiences in the field and Kennedy's findings from her survey of web designers and developers. What would be more interesting—and potentially surprising—would be to dig deeper into which team member made the call that accessibility doesn't fit into a particular budget or timeframe, and whether that individual was exerting undue or disproportionate influence over the project due to their relatively higher political capital[7] than the rest of the project team. For instance,

7 "Political capital" is a term that is used to assign a measure to the amount of influence one is capable of exercising in an organization, which is affected by one's title or position within the organization. Political capital also includes individual perception, which is affected by factors such

was it the case that the project team came to consensus on including accessibility components as part of best practices in web development, and the project manager or account executive rejected the plan put forward by the team because they were in a position to do so, feeling it unnecessary, even if the team was confident that they could include support for accessibility while meeting budgetary and timeframe goals?

In my professional experience, inclusion of accessibility has been a consensus decision, and has been rejected by a single member of a project team —sometimes due to timeframe and budgetary considerations, and sometimes because that individual viewed accessibility as an optional extra and could not be convinced by the rest of the team that including accessibility was worth the effort. This experience was backed up by Helen Kennedy's survey and interview data in Net Work, especially when dealing with accommodating cognitive impairments, commonly referred to as intellectual disabilities (ID):

> “A common response to the question of what obstacles participants encountered to implementing ID accessibility was that clients or equivalent decision-makers did not care about ID accessibility, did not want it, or wanted to retain inaccessible elements in their websites, such as confusing labels for functional features. Clients and line managers

as sex, race, and persuasive ability. Political capital decreases when used (e.g., too many unreasonable or extraordinary requests from the same person will be met with increasing hostility, irrespective of title or position) and increases with positive interaction and organizational goodwill.

> were repeatedly cited by INMD participants as the biggest barriers to accessibility." (Kennedy 2012, 146)

Despite my personal experience and Kennedy's results, my investigation largely falsified my hypothesis outside of my own experience. There were some instances—all of which took place at advertising or marketing agencies, interestingly—in which certain members of project teams exerted disproportionate influence over specifics of a project that were outside of their area of expertise. My informants indicated that these instances were largely driven by ego, or a sense of wanting to control a project. However, these situations were in the extreme minority, and are likely not the cause of inaccessible sites in any statistically significant way.

It is possible that the relatively different structure of advertising and marketing agencies, which typically form large teams composed of a number of individuals, each responsible for specific areas of a project, are more vulnerable to abuse by individuals exercising unequal political capital. Such organizations are not hierarchical in a traditional sense, and ostensibly operate according to a meritocratic model, in which team members are primarily responsible for and exert authority over their particular area of expertise. There is typically not one final arbiter of decision-making power, so decisions are negotiated among team members according to merit, but are vulnerable to being overruled by team members with more political power within the organization.

The majority of my informants, representing a range of different

organizational structures, described the project process as proceeding according to a rough consensus or democratic model, with more weight given to the arguments of team members that were speaking within their particular area of expertise. The typical explanation for why accessibility isn't included in projects as part of routine design and development practice is because the team as a whole tends to decide that it is not important enough to include, not because a particular member of the team in a position of power rejects it as unnecessary. However, because teams typically decide against systematic inclusion of accessibility via a consensus model, powerful team members are not put in the position of needing to reject accessibility in order to meet budgetary or time goals, since the team has already come to that decision on its own.

5.1 Democracy as Standard Operating Procedure

The best description for the way that my informants described routine decision making practices within their organizations is a meritocratic consensus model. In their own words, most of my informants described the arrangement as being “democratic,” but in practice, more weight was given to those individuals with subject matter expertise, particularly those with longevity in a particular field. While it is true that my informants indicated that all members of a project team were encouraged to participate in decision-making discussions, not everyone's voice was treated equally, as it would be in a true democracy. Rather, what my informants were describing when they said that their organization was “democratic” was that the people doing the work were able to make a the greater

share of the decisions—that they were not being micromanaged and dictated to by managers or business owners on the specifics of how to do their jobs.

Such a meritocratic arrangement has its drawbacks. Ostensibly, those with the greatest amount of subject matter expertise should have the most weight in making decisions related to that expertise. In practice, however, meritocratic arrangements fail to eliminate bias based on other factors, such as sex and race, and in some cases actually exacerbate them (Castilla and Benard 2010; Knowles and Lowery 2012).

The meritocratic nature of consensus building in web development groups is becoming increasingly quantitative. As new tools and techniques become available to analyze user behaviors—including event tracking, eye tracking, and mouse movement tracking—web development groups are able to draw on empirical data to make arguments for or against specific approaches. The entire point of A/B testing, for example, is to quantitatively prove that one approach is superior to another, such as the color or size of a button, or the location of a login link. There are other sources, such as the Nielsen Norman Group, co-founded by user experience researchers Jakob Nielsen and Donald Norman, that publish articles and analysis on user experience. Being able to wield quantitative data is often a prerequisite for making arguments for or against particular approaches to design and development.

> “I always try to find some kind of piece of factual evidence, otherwise I just don’t feel confident enough.

> Depending on the time that you have to make the decision, sometimes you just don't have that privilege, to go find something, and you just have to know, or think you know, and then make the call. But whenever I can, I like to bring some kind of factual piece to the table, and then it helps everyone grow, really, if it's something that I didn't know before, that we're having a conversation about."
>
> —Informant D

Ideally, every member of the project team is involved with every step of the project. However, in practice, this rarely happens, since project teams typically want to operate with limited overhead, and there are usually multiple demands on the time of senior personnel. As a consequence, there are times when the majority of the members of a project team come to consensus on a particular approach, and get overruled by a senior member of the project team at a later date, since senior members of the project tend to avoid involvement in most projects until very late in the process. My informants indicated that when this happens, it is not out of malice or a show of authority, but rather because that team member possessed a perspective or a piece of information that the rest of the team did not have, which was not introduced until after many decisions had already been made based on limited information due to the lack of participation of the senior team member. According to my informants, when these instances

arose, the project teams were typically accepting of the need for a change in approach, as long as it was not introduced so late in the project that the change necessitated a significant amount of re-work.

> “Oftentimes, it has to do with—I think—the most common instances I’ve seen [consensus being overruled] is when it’s for a strategic purpose, where it’s not immediately apparent to the team, but the person in charge of the strategy pulls things back on track. As far as a specific conversion that isn’t immediately apparent to everyone, but makes a lot of sense when it’s explained.”
>
> —Informant J

There is a danger that too much consensus building too early in the project will limit the creative freedom of team members. If everyone on the project team starts the project with a list of constraints, then it is difficult to create something original, and the project team will typically create the same type of website or experience that they have always created. A number of my informants indicated that their approach was to allow for a phase of creative exploration, followed by a sort of reality check in which the feasibility of what was proposed by subject matter experts was assessed by the rest of the project team. For example, designs proposed by creative would need to be implemented by development, taking into account technological constraints and tradeoffs.

> “What we try and do is, you let the domain experts come up with the ideas, and then you share them with the team. Obviously, people can come up with whatever ideas they want to, but they still need to be feasible and executable. We try and make sure that we don’t restrict each phase too much. You want people pushing boundaries, you want people making new ideas and new innovations. If you tell people, “this is impossible,” right from the beginning, things are going to stay the same forever and ever and ever.”
>
> —Informant N

There are drawbacks to the meritocratic approach. Often, what constitutes “merit,” or even “expertise,” in web development groups is muddled by other factors, such as sex, minority status, and job title. A junior member of a project team with a good idea needs to do significantly more work to defend that idea than a senior member—ostensibly because of the gap in experience, which may not always be justified. In web development, as with many other technical pursuits, the field of knowledge is constantly evolving and changing. Senior members of project teams often have less time to stay current on technologies, techniques, and user experience research. Therefore, junior members of project teams can be better positioned to make recommendations for approach due to

being closer to current information on best practices, but can be overruled by senior members of project teams rather easily.

When the consensus model produces inaccessible websites, because the entire team comes to a consensus on not making the site accessible—either implicitly, via not including the functionality, or explicitly, by expressly rejecting inclusion of accessibility after discussing the possibility of inclusion—something needs to be done to shake up the consensus model. Enter the “accessibility advocate.”

5.2 The Role of the Accessibility Advocate

Designing and programming for accessibility is not the norm within the field of web design and development as a whole, but can be more of a norm within certain web development groups, even when these groups are not working on projects that mandate accessibility. The difference is the presence of a person or persons that play the role of an accessibility advocate.

Accessibility advocates can be individuals that are merely passionate about web accessibility in general. However, based on my interviews and research, I have found that accessibility advocates generally have a personal connection of some sort to the topic of accessibility—either because they themselves have an impairment, or know someone personally that does. Having a connection to someone that has an impairment takes what is otherwise an abstract concept and makes it personal. Instead of designing and programming

for accessibility for the theoretical person with an impairment that may end up using the website, the designer or developer has their personal connection in mind while creating.

> “My fiancée is colorblind. He—there’s been so many instances where I’ve pointed things out to him that he just couldn’t see. So from that—actually, that was a pretty good heads up on colors that just don’t work together for that audience, with his specific type of colorblindness—but it really made me want to know how that affects other groups of colorblind people too, because there’s all sorts of different types.”
>
> —Informant D

My informants described the presence of an accessibility advocate on a project team as raising awareness of the need to design and program for accessibility generally, even if the personal connection was only to one specific type of impairment. Knowing someone with an impairment combined with learning or knowing about how to make the web accessible tends to pay off for accessibility generally.

The presence and effect of one or more accessibility advocates is often correlated to the size of an organization. An accessibility advocate on a project team of only a few people will have a limited reach, but even one accessibility advocate inside of a larger organization can have wide-ranging effects.

Particularly when combined with the consensus model of decision making, an accessibility advocate can make their case for the need or desirability of including accessibility in web projects without needing to be in a position of authority.

> "WordPress also has this really nice—it's a feature of coming out of the open source community—is that there's such an explosion of voices from that community that are really helpful for us as well. But in terms of actual employees, yeah, we have very vocal advocates for accessibility, and for gender and class equality. There are very vocal and opinionated members of the staff, and our flat-ish structure really allows those voices to have a lot of heft, which is really good, I think, morally, and in terms of revenue. I think we're a better company for their being employees."
>
> —Informant E

Public accessibility advocates, operating on social media platforms like Twitter, and posting blogs either on their own sites or on sites devoted to specific communities, can help spur those communities to action and make software more accessible and inclusive where it otherwise would not be. For example, front-end JavaScript frameworks that have emerged in recent years to make the

process of building dynamic web applications like Facebook easier are notoriously inaccessible. Part of this is due to the design of the frameworks themselves, in which front-end performance and ease of development are emphasized over concerns like accessibility and standards compliance. However, a share of the blame also falls to assistive technology manufacturers for lagging advances in web technologies, which makes content inaccessible that need not be.

> “There’s almost no accessibility [in JavaScript frameworks]. It’s a huge problem, and I know that there are some really intelligent people working on it, like Addy Osmani, Sindre Sorhus. Those guys I’ve seen tweeting a lot about getting web accessibility into these JavaScript platforms.”
>
> —Informant I

One of my informants works as an accessibility instructor and consultant. She provides training to a wide range of individuals working in web design and development fields for the public and private sectors. She indicated that the presence of a personal connection is the defining factor for some people about whether they will care about including accessibility in their web design or development. For others, merely becoming aware that the web is not automatically accessible is enough to spur action.

> “I’d say it’s a mixed bag. There are people who just

> don’t care, and won’t care, until it impacts them personally, whether it’s a loved one or an acquaintance, or themselves. But other people, they are shocked to learn that people with disabilities can access things on the web, in some cases. Then there are people who are shocked to learn that it’s poor planning that prohibits people with disabilities from accessing information. They just assume that, sure, everyone can, and there’s nothing special that they need to do. So there’s a good mix.”
>
> —Informant K

Within larger organizations, especially technology companies and open source projects, there are often subgroups devoted to accessibility. One such notable group is called Make WordPress Accessible, which is responsible for ensuring that any changes made to the open source WordPress content management during its semiannual release cycle are accessible. A similar group exists for the Drupal content management system. Large technology companies also have internal accessibility advocacy groups, including Twitter, Facebook, PayPal, Microsoft, Adobe, and Google.

5.3 Mangling Power Relations

In Andrew Pickering's 1995 book The Mangle of Practice, he advances the theory of the “mangle of practice” as “a *dialectic of resistance and*

accommodation" between social, technological, conceptual, and natural factors (Pickering 1995). The mangle is a theory that embraces complexity and the ebb and flow of negotiation between factors. Web development, and particularly developing for accessibility, represents "a dialectic of resistance and accommodation" between various groups and factors:

- **Designers and developers vs. technological limitations:** Designers, representing the creative, and developers, representing the build phase, are often engaged in spirited disagreement about how to implement a design. Designers are concerned about the visual appeal of the website, and often about usability, whereas developers have deep knowledge about what aspects of a proposed design will be difficult or impossible to implement reliably across a range of devices and web browsers. In some cases, designers can push developers to re-examine what they "know" to be true about browser or device support for a proposed design element, and possibly come up with a solution that satisfies both parties. In many cases, developers are in the position of having to either say no to a designer, or to work with them to scale back their design to meet the technological limitations. Even if a team member is in a position of power, no amount of exertion of influence will change the technological barriers or limitations preventing or limiting the request from being granted.
- **Designers and developers vs. accessibility:** Accessibility is a subset of the above argument. If a project team, or just the developers on a project

team, plan on including accessibility in a website, they need to work together from the beginning to ensure success. As a consequence, accessibility standards or guidelines (and occasionally user tests) will necessitate changes to the design and the build. Some of these changes can be challenging to incorporate in a way that is satisfying to all parties, while still remaining accessible. For example, a best practice to accommodate blind users utilizing screen readers is to include a “skip to content” link as the first element on the page, which, when clicked, will bypass the header and primary navigation and jump the focus of the screen reader directly to the main content of the page. Designers often do not want this link to be visible, but it still needs to be selectable by a screen reader. This presents a challenge for developers to keep this functionality while meeting the visual requirements of design in a way that will be compatible across different devices and browsers.

- **The project team vs. the client:** The client, or business owner, typically has final say over the look and functionality of a website, and they also control the budget and set the timeframe. As a consequence, there is a process of negotiation between the project team and the client in order to produce the best possible website within the given budget and timeframe that also meets all of the client's requirements. This dialectic is where negotiations about which best practices—or limited versions of best practices—can be included in the build, with budget and timeframe as

limiting factors.

- **Developers vs. frameworks:** Frameworks, such as CMS platforms and front-end frameworks, provide developers with the ability to produce sites faster and more reliably than building them from scratch. However, the gains in speed and reliability come at a cost. Frameworks are, to some degree, “opinionated,” which is to say that they come with a particular style and way of functioning, which will vary in type and degree from framework to framework. Part of a developer's task is negotiating with this “opinionated” nature of the framework to implement the proposed design. If the developer embraces the opinions of the framework, development time will be relatively short, but the built website may not adhere as closely to the design as the designer would like. If the developer adheres too closely to the design and ignores the opinions or intended use of the framework, then the efficiency gain of the framework is lost, and it can become bloated and slow. Developers need to walk a fine line between implementing the vision of the designer and efficiently managing the build within any frameworks that may be in use.
- **Designers and developers vs. account executives:** In the agency model, the account executive is supposed to represent the will and desires of the client. As such, account executives will often steer the direction of the design and development to meet what they perceive as client requirements or wishes in advance of sharing the design or build with the

client, in an effort to minimize the gap between what is presented and what the client will ultimately approve. However, at times, account executives become too overzealous in the pursuit of client satisfaction, and preemptively require excessive changes to the build that violate best practices or consensus recommendations. There is a process of negotiation between account executives, designers, and developers about where to draw the line between sensible preemptive changes and overreach. From my experience and interview data, I have found that in this interaction, more than anywhere else, the relative organizational power and political capital of the players is often the deciding factor in who gets overruled.

- **Strategy vs. everyone else:** The digital strategist is often in the unfortunate position of having to shut down entire design elements or lines of thinking when the project team strays from the strategic goals of the project. For example, if the website is intended to be optimized for specific search terms, then those search terms need to be included in page titles, headlines, and page copy, which guides and constrains the activities of the copywriter. Likewise, if a design does not properly emphasize the intended action that the user should be taking on a page, then the digital strategist will need to tell the designer to redesign that element to meet the strategic goals. A frustration expressed by several of my informants was that digital strategists are often spread thin, and are unable to be present

for all phases of the project, so this feedback often comes late in the process and necessitates rework.

The dialectic of resistance and accommodation is a daily reality for people working in the web design and development profession, in ways that are often more varied and pronounced than other creative and technological pursuits, due to the depth and breadth of concerns that go into a website project.

5.4 Conclusion

Although my ultimate conclusions for this chapter are markedly different than my hypothesis at the outset of the project, I feel that my informants provided me with a richer explanation for the routine practices of negotiation between project team members and the technologies that they use and support. Although relative power and ego do play a role in the design and development process, which can be significant within particular organizations, it is typically not the deciding factor between including vs. excluding accessibility. Project teams, proceeding according to a meritocratic consensus model, will typically not include accessibility in the final website unless it is an explicit requirement by the client or the law, or unless there is an accessibility advocate on the project team driving everyone toward including accessibility.

My informants indicated that accessibility is typically not actively excluded, but rather simply not included. This is an important distinction—when project team members are aware of the need for accessibility and the methods of

making an accessible site, particularly when there is an accessibility advocate pushing for inclusion, accessibility tends to be included in projects, at least to a certain degree. In most cases, project team members are not aware of the need for designing and developing for accessibility, or how to do so, or the range of different impairments that need to be accommodated for a site to be fully accessible.

When present, an accessibility advocate has the ability to exert wide influence over the output of a variety of teams simply by being able to argue in favor of inclusiveness, and to guide project team members toward methodologies for creating accessible sites, even across disciplines and traditional lines of power. Therefore, including people with impairments in the design and development phase, and encouraging people with impairments to become designers or developers, is of critical importance. Additionally, educating designers, developers, copywriters, and digital strategists generally about the need for accessibility, and the methods for how to do so, would likely improve accessibility overall, even if not all of those individuals become what I would call an accessibility advocate.

6. DISCIPLINARY NORMS

In Net Work, Helen Kennedy advances the hypothesis that web designers and developers are, as a group, “ethically self-regulating,” and committed to web accessibility, at least at a basic level. She argues that the disciplinary norms of web design and development include accessibility as a core moral commitment, which most web designers and developers incorporate into their work as a matter of professionalism, since accessibility is considered to be part of what constitutes “doing good work.”

My findings from my survey of 330 web designers and developers, conducted in 2011, contrasted sharply with Kennedy's survey and interview data, which was conducted in the mid- to late 2000s (Kennedy 2012, 17–18). I hypothesize that this difference is due to three limitations of Kennedy's work:

1. **Geographic bias**—Kennedy conducted her surveys and focus groups with web designers and developers exclusively in the UK (Kennedy 2012, 18), which has a drastically different legal and regulatory framework than the United States in terms of requiring websites to be accessible. In the UK, the Disability Discrimination Act (DDA) requires websites to be accessible, whereas the ADA in the United States does not have language explicitly applying it to websites, and the courts have not ruled that there is an ADA requirement for websites produced in the United States to be accessible. Although Kennedy says that her study is of web designers and developers in the US and the UK, the bulk of her qualitative and

quantitative data collection centers on workers in the UK, and is therefore biased toward the viewpoints and activities of a group of web workers that are operating under a very different system of laws. I ague that the material conditions of web design and development in the US are different enough from the UK that it is inappropriate to use a dataset primarily comprised of UK informants to extrapolate to the intentions, experiences, and realities of web work in the United States.

2. **Interviews with field leaders**—Kennedy conducted interviews with “a small number of web design field leaders” (Kennedy 2012, 18), which she uses to advance the argument that the disciplinary norms of web designers and developers include accessibility because it is valued by thought leaders in the field. Since field leaders advocate particular commitments to standards and the ethics of accessibility, the argument goes, average designers and developers adopt many of the value systems of these thought leaders as their own. I argue that there is a major distinction between what thought leaders in the field are able to hold and advance as core values versus what the average web worker is able to incorporate in their own work, for a variety of reasons, which I will explore in more detail below.

3. **Intentions vs. action**—Kennedy's focus group was comprised of a group of “final year new media students about to embark on careers as web designers” (Kennedy 2012, 18). The focus group data is not a reliable

indicator of work practice because it only measures future intention, not present action, and is not reflective of ideology tempered by the realities of the workplace, given that the focus group participants had not yet entered the workforce. I argue that the realities of the workplace, particularly ideological conflict with clients and managers, cause a rejection or scaling back of accessibility in practice, even if accessibility was emphasized in college or trade school.

I do not mean this chapter to be a rejection of Kennedy's work, or even a challenge to many of its central claims. Rather, I intend to demonstrate two things—first, that Kennedy's work is better read as a report on the ethical commitments of web workers exclusively in the UK, rather than a report on web workers in both the UK and the US; and second, that Kennedy's work is primarily an exploration of the stated ideologies of web workers, rather than a report on their actual work practices—a limitation that she herself acknowledges in the conclusion (Kennedy 2012, 215). I posit that the conditions of web work in the United States differ significantly enough from the UK to deserve their own study, for which this project is a beginning, although it suffers from the same limitations as Kennedy's work of reliance on self-reported data through interviews and surveys.

6.1 Construction of Disciplinary Norms

Disciplinary norms in web design and development are primarily constructed from three sources:

1. **Formal Education**—Most often, post-secondary education at a college or trade school. Increasingly, however, online courses—some from accredited institutions, but many from formalized programs on “learn to code” websites—are supplementing or taking the place of traditional formal education. Web design and development is not a field that typically requires a college education, as evidenced by my survey data. Over a third of my survey respondents did not complete a college education: 7% never attended college, and 31% attended some college but did not receive a degree. 8% received an Associate's, 46% received a Bachelor's, and 8% earned a Master's or higher. As I will demonstrate later in this chapter, formal education plays a relatively minor role in the disciplinary norms of web designers and developers.
2. **Work Experience**—The daily work practices, and the ideologies of coworkers and managers, guide and shape designers and developers. Even when a designer or a developer learns something as part of formal education or ongoing professional development, those ideologies and arguments are shaped by interaction and discussion with coworkers, being presented with alternate opinions and ideologies, and the consensus model of development described in Chapter 5. In particular, designers or developers in mentorship roles—formalized or not—exert significant influence over the disciplinary norms of mentees. The amount of work experience among my survey participants was representative of a

range of career lengths: 7% were junior-level, having less than 2 years of experience; 38% were mid-level, possessing between 2-5 years of experience; 29% were senior-level, having 5-9 years of experience; and 26% were very senior or management level, possessing 10 or more years of experience.

3. **Professional Development**—Attendance at conferences, taking courses from colleges or online “learn to code” websites, reading articles by thought leaders and other designers or developers, visiting news aggregation sites like Reddit or following design and development blogs like AListApart, and following thought leaders on social media channels like Twitter all constitute professional development activities. Nearly everyone I interviewed and 84% of my survey respondents indicated that they participated in some form of professional development. The extent to which my informants were able to participate in professional development activities that cost money, such as attendance at conferences or taking continuing education courses, depended largely on whether their employer was paying for it or not. In particular, my informants that were self-employed often could not justify the cost of paying full price to attend a conference.

Disciplinary norms will differ for designers vs. developers, for the self-employed vs. employees of a larger company, for members of a dev shop vs. a fully-integrated marketing agency, and for small teams vs. large teams. There are

commonalities that run across these axes of difference, but it cannot be said that there is a unified set of disciplinary norms that applies to all designers or all developers. Rather, disciplinary norms are shaped significantly by individual context, and can change when the designer or developer changes jobs.

In keeping with Kennedy's findings, one major commonality emerged from my survey and interview data—the concept of “doing good work.” There are various dimensions to what “doing good work” means, and the definition changes from person to person, but the designers and developers that I interviewed expressed a commitment to producing work that they could be proud of, that reflected their own sense of what making a quality product represented. In many cases, “doing good work” was manifested as a commitment to writing well-structured, standards-compliant code. It also manifested as creating usable, intuitive, and visually appealing interfaces. However, most of my informants indicated that while accommodating accessibility was a best practice and a goal to strive for, it was not part of their disciplinary norms, and was therefore not applied to all of their projects.

6.2 Web Standards and Ethical Self-Regulation

> “[S]elf-regulatory practices like web standards can be conceived as emerging from a set of values which are concerned with good practice and good products…the chapter proposes thinking about self-regulation as both ethically informed and, occasionally, more

> effective than legislative regulation.” (Kennedy 2012, 85)

Kennedy's argument about ethical self-regulation is that the production of disciplinary norms through ongoing professional development, including following the values of thought leaders in the field, includes ethical development on the part of the individual designer or developer. Kennedy maintains that since this ethical development is derived from the self through internalizing ideologies and arguments made by respected others, the individual's commitment to ethical action is more powerful than if ethical action was mandated by law. In this regard, Kennedy takes a philosophically libertarian position against regulation, arguing that regulation will have an effect that is the opposite of its intended action. For Kennedy, the primary vehicle and indicator of a commitment to ethics is adherence to standards, including accessibility standards:

> “The standards-compliance of web designers is a form of individual and collective self-regulation. The individual web designer validates her own code to ensure standards-compliance, and follows the blogs and Twitter updates of leading proponents of standards for their latest views. At the same time, web designers share knowledge and expertise, and engage in ongoing dialogue and debate on blogs and microblogs, reading and commenting on each other's

> sites and code, and so participating in *collective* self-regulation of standards." (Kennedy 2012, 104)

Standards compliance is a type of stand-in for accessibility testing. While accessibility testing is required to ensure that the website is fully accessible, adherence to published standards will most often result in a website that is more accessible than if the standards were not followed, and is often the only feasible option when budgets and timeframes do not allow for a full user test. Even adherence to general W3C standards for HTML and CSS typically results in a more accessible site, since standards-compliant and semantic code is more easily understood by assistive technologies such as screen readers, and accessibility attributes such as an alt attribute on an image tag are required by the general HTML specification.

A second definition of standards emerges here as well—the standards of practice. These standards are different from the codified standards of the W3C, or even codified organizational standards such as an organization-specific coding specification. This second type of standard—what I term "disciplinary norms," to elevate and differentiate it from the formalized, codified standards of the W3C—exists more as learned behavior, or a feeling of doing things "the right way." Disciplinary norms are constructed over time using a variety of sources, so it is difficult to point to one in particular that represents the proper way to approach a particular task. Rather, disciplinary norms exist as constructs within the mind of the individual designer or developer, representing their view of the general

approach taken by their industry toward particular problems or tasks.

Complete accessibility standards compliance at the level of the individual designer or developer is rare. Kennedy argues that the accessibility standards are too complex for most designers and developers to fully understand and implement, since they do not have sufficient time in their professional lives to read and understand them completely (Kennedy 2012, 109). As a consequence, when designers and developers practice accessibility standards compliance, it is often limited to a minimal set of basic standards, such as populating the alt attribute of the image tag with meaningful descriptive text, using text in place of images wherever possible, and ensuring adequate contrast on text to preserve readability. Additionally, adherence to these accessibility best practices are rarely "purely ethical," as they typically have synergistic benefits in other arenas, such as predictable display and performance across operating systems and devices and enhanced search engine optimization (Kennedy 2012, 214).

When Kennedy speaks of ethical self-regulation, she is talking about the commitment of designers and developers to what they perceive as ethical action, irrespective of whether their actions actually produce ethical or just outcomes. I argue that the stated commitment of designers and developers to ethical action is markedly different from the material products of their labor, and the way that their work product is received and used by people with impairments, even if the sites they create are produced with the best of intentions.

6.3 Dimensions of Commitment to Web Accessibility

> “[F]ocusing on the web's inaccessibility overshadows any acknowledgement of the advances that have been made in web accessibility and the commitment of many web designers to the accessibility ethos.” (Kennedy 2012, 108)

Kennedy's point is well taken. It is easy to be pessimistic about web accessibility, given that the state of web accessibility remains rather dismal in 2015. It is true that there have been gains in web accessibility over time, many of which have been aided by advances in web technology that allowed designers and developers to overcome challenges in an accessible way. An excellent example of this phenomenon is the shift from using images containing stylized text—which are not accessible to assistive technologies like screen readers without requiring extra work on the part of the developer—to using custom embedded fonts styling native text in the HTML, which is natively readable by assistive technology without requiring special steps to make the text accessible.

The second half of this statement—“the commitment of many web designers to the accessibility ethos”—requires further scrutinization. Kennedy argues that a stated aim of web designers and developers is to produce accessible sites (Kennedy 2012, 114). My interview data bears this out, in that the designers and developers I interviewed stated that producing accessible sites is a goal to strive for, and is a part of what they conceive of as “doing good work.”

However, most of my interviewees, and most of my survey subjects, indicated that producing accessible sites is not the norm for them, which is typically due to budget and time constraints.

Additionally, there is a knowledge gap that contributes to the production of inaccessible sites. Most of my informants indicated that sites should be made accessible, but that they did not know how to do so, beyond a few basic best practices. Most of these best practices were implemented because they had a primary benefit other than accessibility, such as optimizing the site for search, or speeding up the load time of the site by reducing the number of assets that a web browser would need to load in order to display the site.

Kennedy bases her argument about commitment to accessibility on two factors: first, the designers and developers that she interviewed (not all of whom, it is important to note, had yet entered the workforce) indicated that accessibility was an important moral value for them; second, that designers and developers follow thought leaders in the industry on microblogging services such as Twitter and long-form blog sites such as AListApart, and those thought leaders have a vocal commitment to accessibility.

> “Following the charismatic leadership of [Jeffrey] Zeldman, [Eric] Meyer and others involves trusting in them and the values that they advocate—web standards, accessibility, other forms of doing good.” (Kennedy 2012, 193)

My interviews and survey data agree with this conclusion, with a caveat: the professional development activities of web designers and developers include adopting an understanding of best practices that includes such elements as web standards and accessibility, but in an aspirational way. That is, the designers and developers that I spoke to agreed that standards compliance and accessibility were professional values that they held, but rather than being core components of their work practice that were implemented routinely, they were goals to strive for under ideal circumstances, such as when clients and managers required or allowed for inclusion of accessibility components in a project.

My informants tended to view accessibility and standards compliance as optional extras that would be included if they were given appropriate time and sufficient budget, knew how to implement accessible methodologies, and if inclusion did not compromise other parts of the project that were viewed as more important, especially the visual display of the site.

> “It is clearly the case that, for web design celebrities, the self-promotional and the ethical intersect. The presence of accessibility at web design conferences, in individual presentations, or as ‘a given’ is one example of this.” (Kennedy 2012, 205)

When Kennedy speaks of “web design celebrities,” she is speaking of thought leaders in the field that are typically at or near the pinnacle of their careers, usually working for large creative or technology companies or running

their own. These thought leaders are deeply educated about web technologies, in part because many of them are or were core contributors to the standards or the technologies themselves.

By contrast, non-celebrity web designers and developers interact with web technologies at a much higher level, and have an understanding of the technologies that is far less deep. Average web designers and developers are much less likely to know the extent to which a website's content and markup needs to be structured in order for it to meet the accessibility standards, and are much less likely to have resources available to them to conduct user testing with people with impairments. As a consequence, it is worth exploring the ideologies of average web designers and developers with respect to accessibility and work practice, and what factors shaped these ideologies.

When I conducted a survey of web designers and developers in 2011, I had several hypotheses that I wanted to test regarding factors that influenced designers and developers in producing accessible sites. For instance, I hypothesized that designers and developers with a college education would be more likely to know about and implement accessible designs than those without a college education. I also hypothesized that a specific emphasis on the importance of accessibility in college would translate to an emphasis on accessibility in work practice.

When Helen Kennedy's Net Work was published in 2012 with a focus on the same issues, I was simultaneously elated and depressed—since it appeared

as if a scholar with more experience and deeper insight had published a more robust investigation into my dissertation topic than I would be able to produce. On the one hand, it would be a great source of information; but on the other, I would need to determine how to differentiate myself from Kennedy's work and findings.

My hypotheses seemed to line up with Kennedy's core arguments, but my findings in my survey data did not bear out the same conclusions. Likewise, my interview subjects expressed different sentiments about accessibility than Kennedy's. Although aggregate analysis of my interview data did show some correlation along the lines of my hypotheses, the correlation was not statistically significant using a confidence value of 95%.

Most of my hypotheses involved breaking down the survey respondents by how much they value producing websites that work for users with impairments on a scale from 1-5:

1. Not important
2. Somewhat important
3. Important
4. Very important
5. Central focus

I would expect that the only people answering “5” would be those that either had a close personal connection or personal commitment to accessibility,

or worked in an industry that required accessibility due to law or intended audience. Therefore, the spectrum of typical web design and development projects would be expected to fall on the 1-4 scale, and anyone answering “5” would be an exceptional outlier.

Once my respondents were sorted according to stated commitment to accessibility, I further broke down the results by other dimensions, such as education level and type, professional development activities, et cetera, to see if the difference between those subcategories had a statistically significant effect on accessibility outcomes.

In order to perform my analysis, I loaded my survey responses into a MySQL database, performed normalization on some free-response fields in order to facilitate data reporting in the aggregate, and wrote custom SQL queries to analyze the data utilizing subqueries for each of the accessibility commitment values, which were joined together to produce the raw result. The raw results were then analyzed for statistical significance using the chi-squared method with a 95% confidence rate ($p=0.05$). None of the hypotheses registered as statistically significant.

To provide a simpler view of the data, for all of my hypotheses that involved numeric commitment levels, I also created aggregate charts that include a raw average of the numerical commitment level based on several factors. Using this view, the differences between groups are easier to see, but are still not statistically significant. For more information on my survey questions, the results,

and my methods of normalization, please see Appendix A.

6.3.1 Hypothesis 1: People who attend professional conferences or hold certifications are more likely to value having their websites work for users with impairments

Following Kennedy's arguments about web designers and developers following thought leaders in the field who are committed to accessibility, and attending conferences where accessibility is presented as an element of best practices, I would expect that designers and developers that attend such conferences or who have attained certifications as part of ongoing education in the field would have a higher commitment to accessibility. While the data shows some correlation, the difference was within the margin of error, demonstrating that there is no significant difference between designers and developers that attend conferences and hold certifications versus those that do not. It is also important to note that 82% of people that I interviewed indicated that they do not attend conferences or hold certifications, demonstrating that conference attendance is the exception rather than the rule of routine practices in the industry.

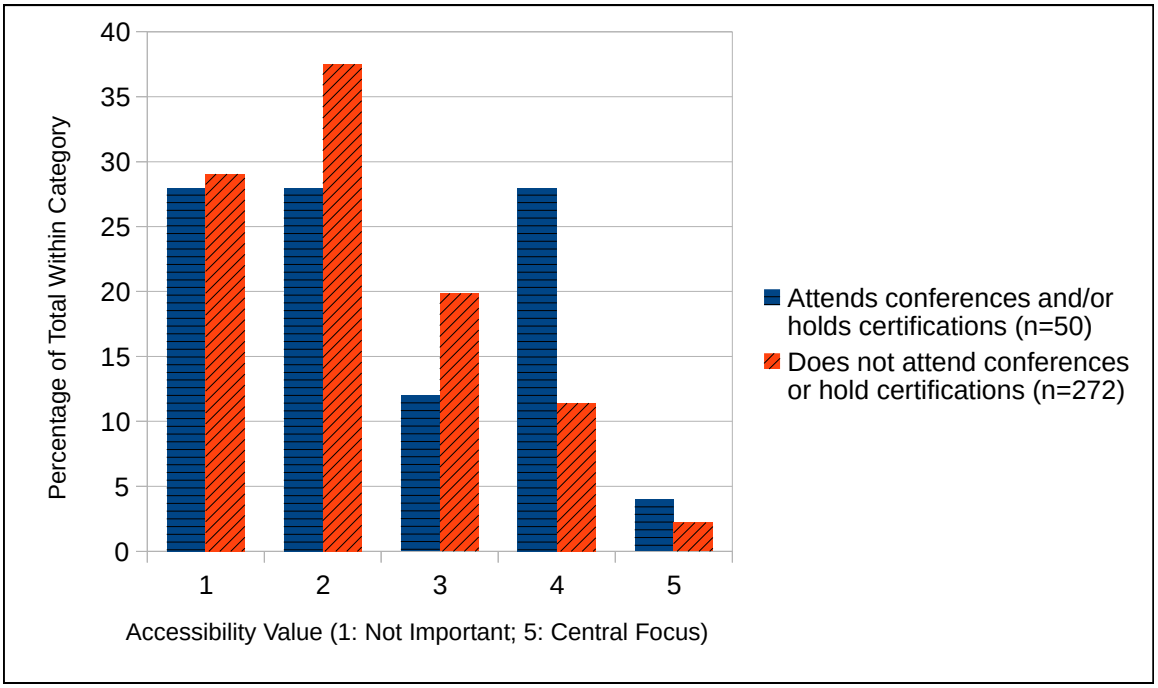

Figure 1. Hypothesis 1

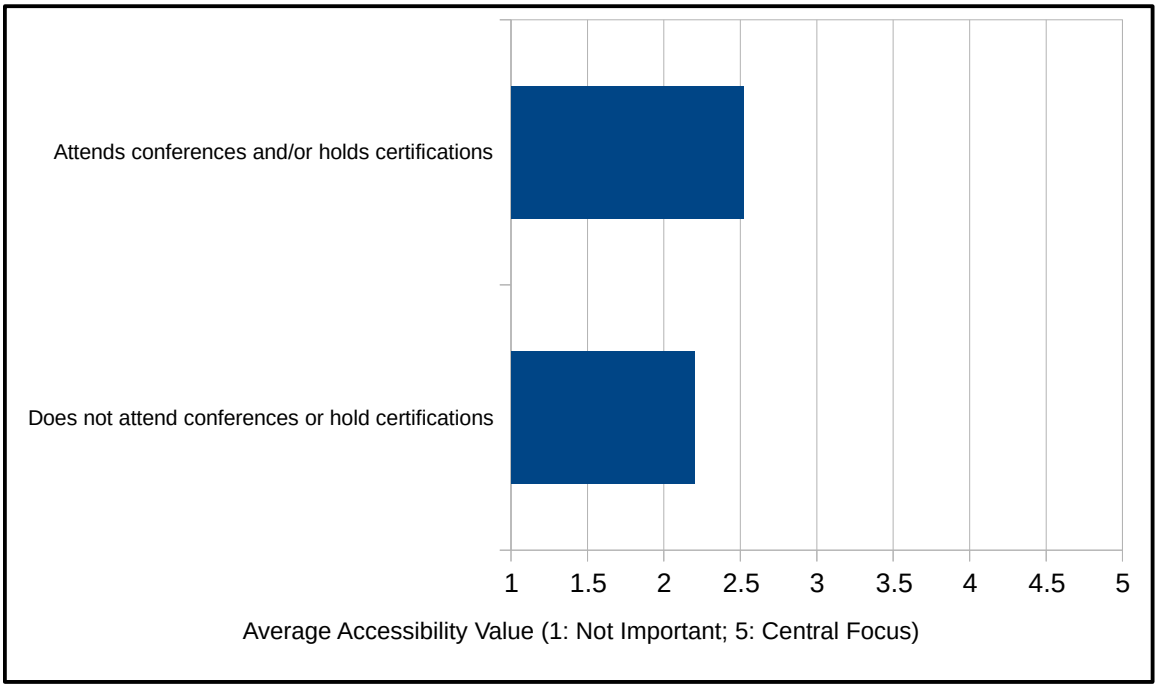

Figure 2. Hypothesis 1 (Aggregate)

6.3.2 Hypothesis 2: People who code for W3C standards compliance are more likely to value having their websites work for users with impairments

Kennedy argues that standards compliance and accessibility are part of professional ethics in web development, particularly because there are published standards for accessibility which are largely testable or verifiable in some way. Therefore, I hypothesized that developers that are committed to one set of standards—the W3C HTML and CSS standards—would be committed to the accessibility standards also. Despite a correlation, statistical evaluation proved the difference to be within the margin of error. It is worth noting that 77% of my respondents indicated that they code to W3C standards whenever possible, indicating that W3C standards compliance, if not accessibility standards compliance, is part of disciplinary norms for web development.

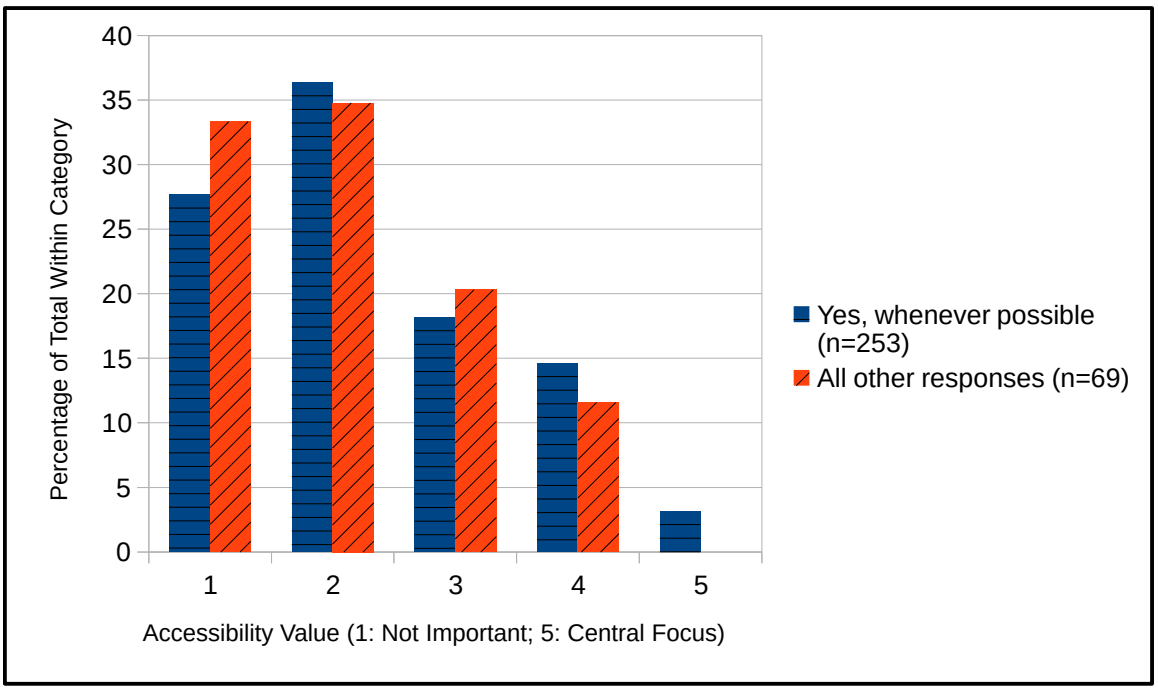

Figure 3. Hypothesis 2

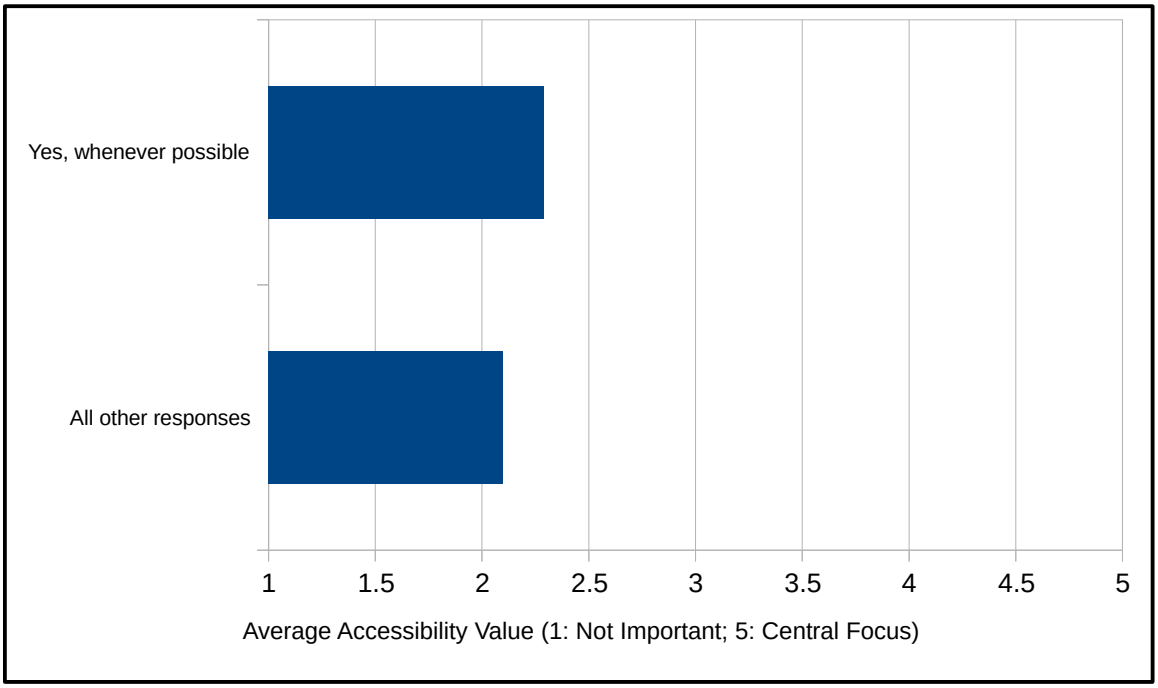

Figure 4. Hypothesis 2 (Aggregate)

6.3.3 Hypothesis 3: Developers that work for a company whose primary line of business is making websites are more likely to value having their websites work for users with impairments

This hypothesis was founded on the idea that a deeper knowledge of all aspects of web design and development, fostered through working in an environment primarily dedicated to it, would produce more awareness and adherence to accessibility techniques. Although my results were not statistically significant, the correlation line points to the opposite of this hypothesis being true. Out of the people that I surveyed, individuals that worked for a company whose primary line of business is making websites were less likely than those working for a company whose primary line of business is something other than making websites, or freelancers, to indicate that they valued accessibility.

My interview data hints at why this might be the case: developers working for companies whose primary line of business is not making websites are more likely to be working on projects directly related to the functioning of the business, which may involve legal regulations or internal policies mandating accessibility in ways that websites built by marketing agencies and their ilk often will not. Freelancers, who demonstrated the highest commitment to accessibility as a group, have more autonomy and control over the designs they produce and the code that they write, and are often better positioned to translate a personal ethical commitment to accessibility into action than those in fully-integrated advertising or marketing agencies. A broader survey would be useful to see if the

difference is statistically significant given a larger sample size.

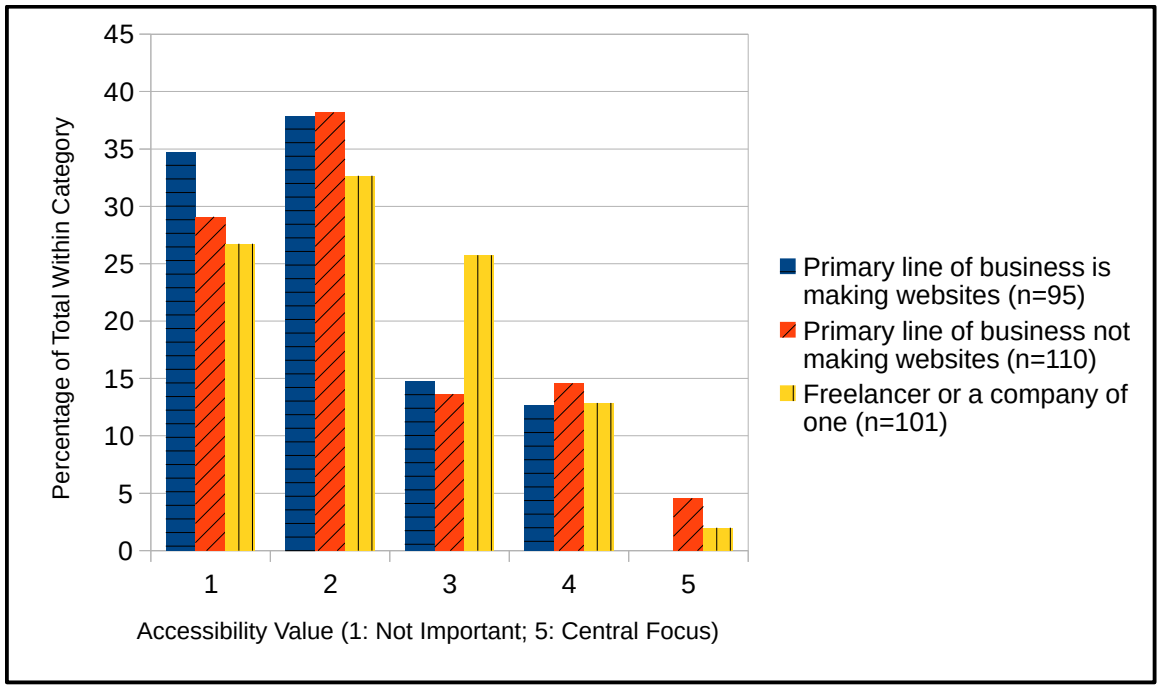

Figure 5. Hypothesis 3

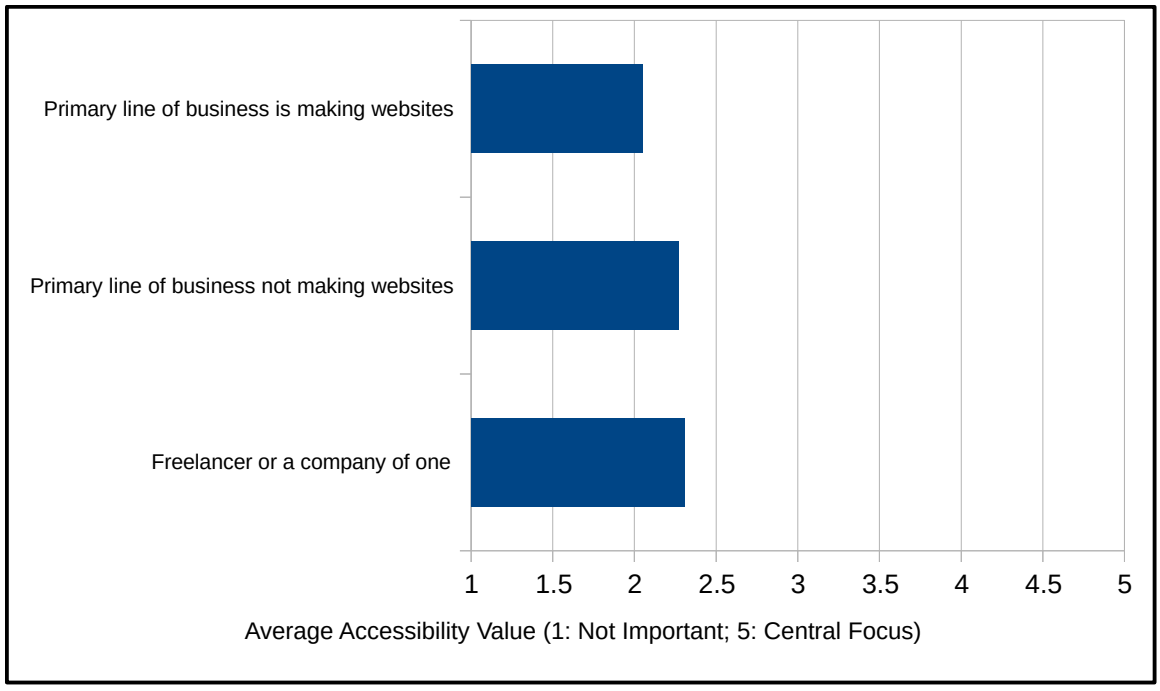

Figure 6. Hypothesis 3 (Aggregate)

6.3.4 Hypothesis 4: Developers that are responsible for all elements of website design are more likely to value having their websites work for users with impairments

Building on the general idea of the previous hypothesis, this hypothesis surmises that so-called “full stack” developers, or those that are responsible for both front-end (HTML and CSS) and back-end (server-side) development would have a deeper knowledge and understanding of accessibility techniques and best practices, and would integrate them at all levels of a website build, instead of being able to place accessibility under the heading of “things that are not my responsibility.” Instead of being able to assume that the other half of the development partnership would integrate accessibility, it would necessarily fall to the full-stack developer, since they are exclusively responsible for writing all aspects of website code.

What I found was that there was virtually no difference whatsoever between front-end, back-end, and full-stack developers in terms of whether they would practice accessibility techniques—on the whole, they uniformly did not.

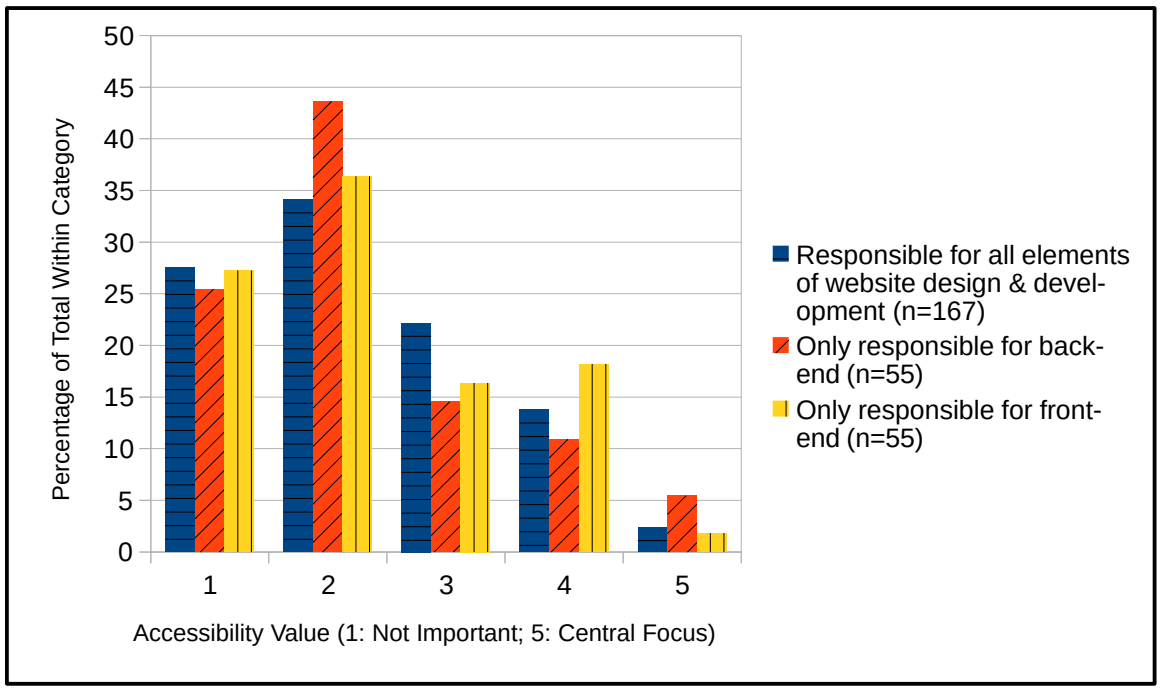

Figure 7. Hypothesis 4

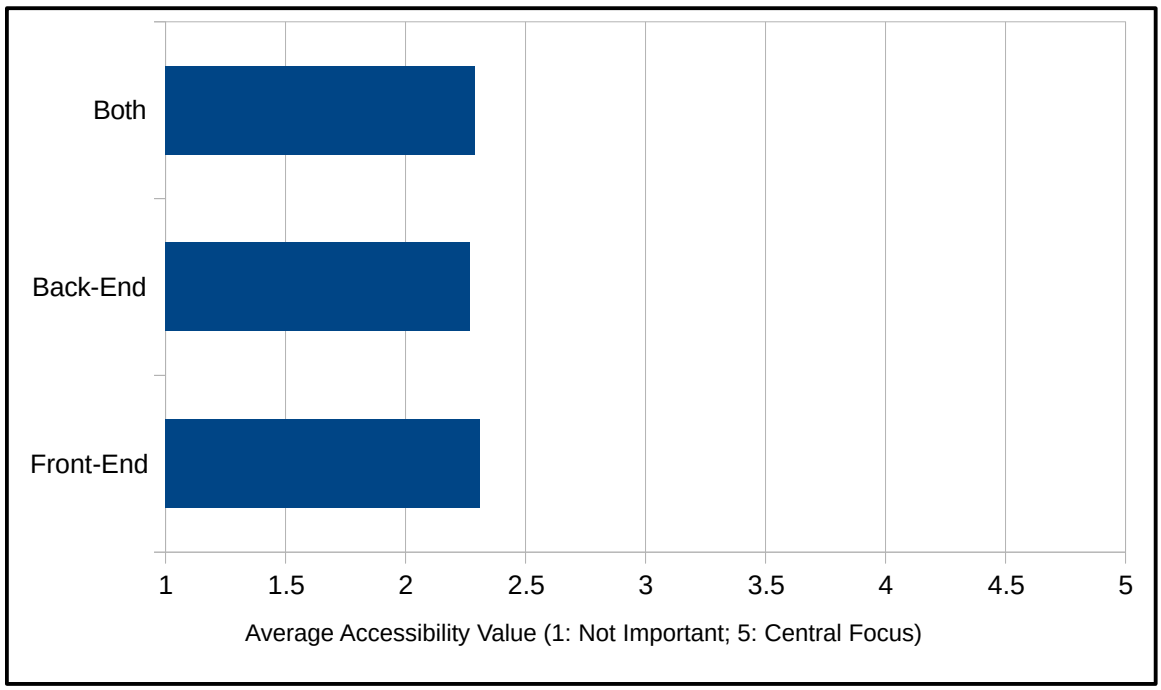

Figure 8. Hypothesis 4 (Aggregate)

6.3.5 Hypothesis 5: Amount of work experience is positively correlated with placing value on having websites work for users with impairments

In keeping with Kennedy's argument that it is very difficult for designers and developers to know and understand all of the aspects of the accessibility standards, I hypothesized that standards and best practices for accessibility would be learned over time, and that more seasoned designers and developers would be more committed to producing accessible sites. When divided by all of my dimensions of longevity of work experience, the results are not statistically significant, but they demonstrate correlation of developers with more than ten years of experience with accommodating accessibility.

I re-ran the statistical significance test by limiting the results just to those with ten or more years of experience versus those with less, and although even this test was not statistically significant, it missed the statistical significance threshold by a very narrow margin. If this survey was conducted using a larger sample size, it is possible that a statistically significant result could emerge.

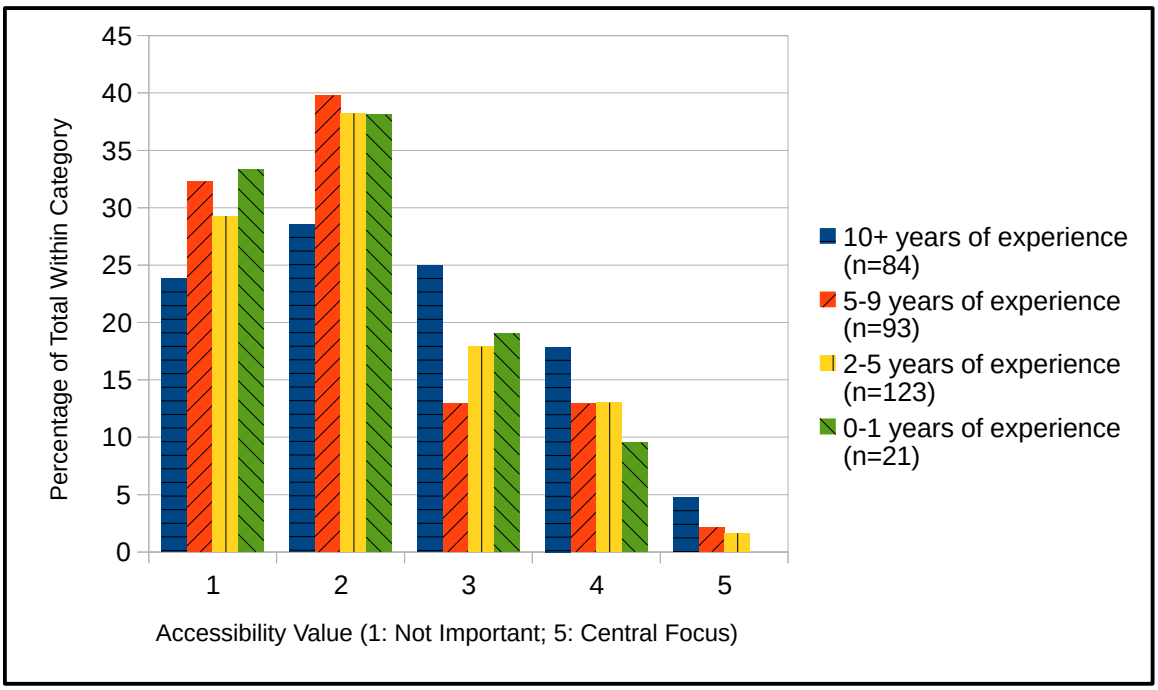

Figure 9. Hypothesis 5

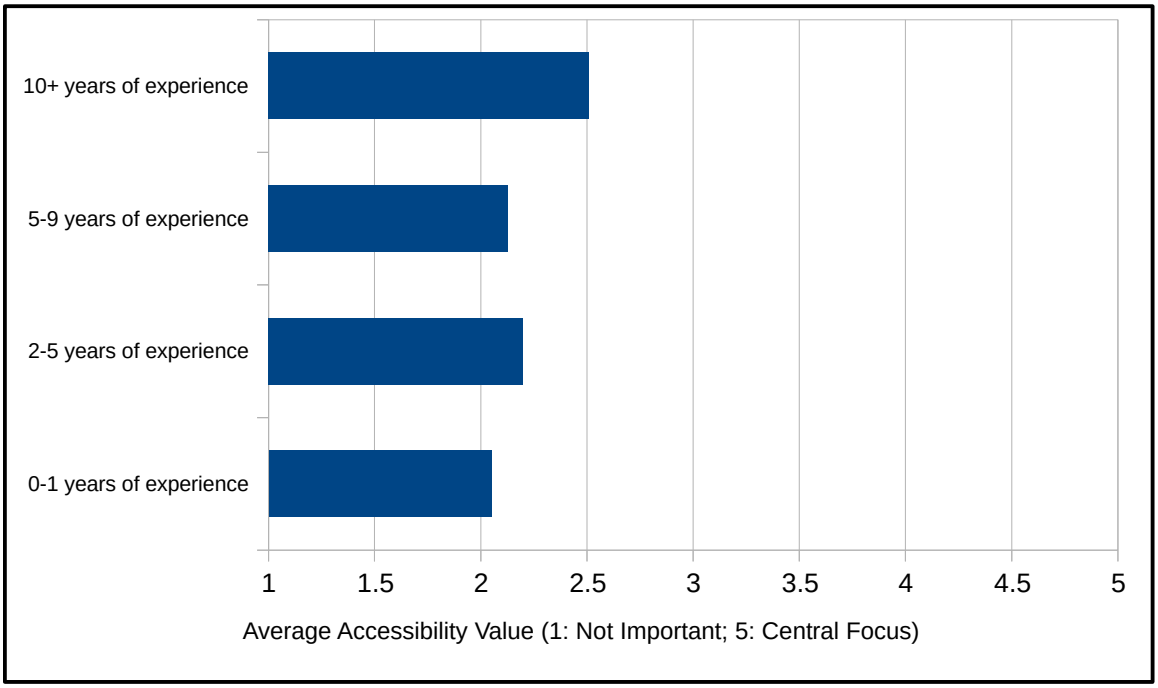

Figure 10. Hypothesis 5 (Aggregate)

6.3.6 Hypothesis 6: Amount of education is positively correlated with placing value on having websites work for users with impairments

In keeping with the general idea of the previous hypothesis, I posited that the more education a designer or developer had, the more likely it was that they were taught about the importance of designing and programming for accessibility. Again, the results were not statistically significant, but the aggregate graph reveals some interesting results. Among degree earners, my general hypothesis seems to be correct.

However, there is a curious bump in commitment to accessibility among individuals that have attended some college, but have not finished, which is a rather large part of my sample size (31% of those that answered this question). It is worth further exploration to determine what it is about designers and developers that started college, but left before attaining a degree, that appears to make them more likely to design and develop for accessibility than their degree-holding college-educated peers.

It may be that designing and developing for accessibility is better learned outside of the context of formal education. Most of my informants and survey respondents indicated that they received little to no accessibility education in college, whereas accessibility appears to be relatively more present within online tutorials and courses, and is advocated among industry thought leaders in blog posts and articles about web design and development.

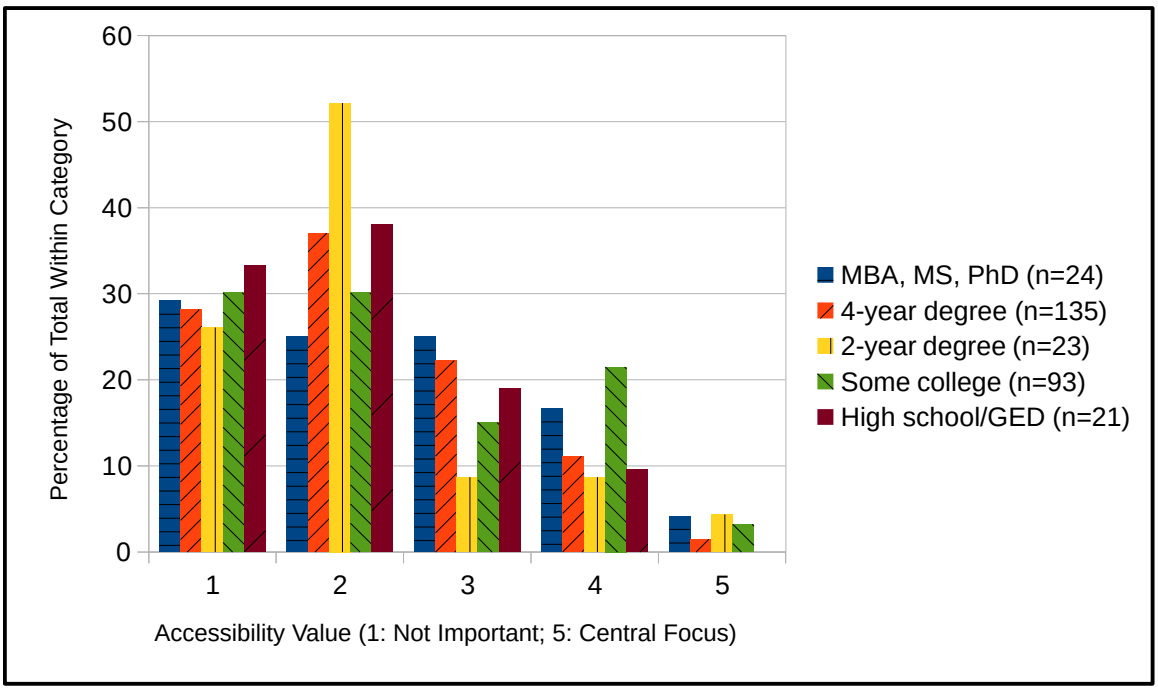

Figure 11. Hypothesis 6

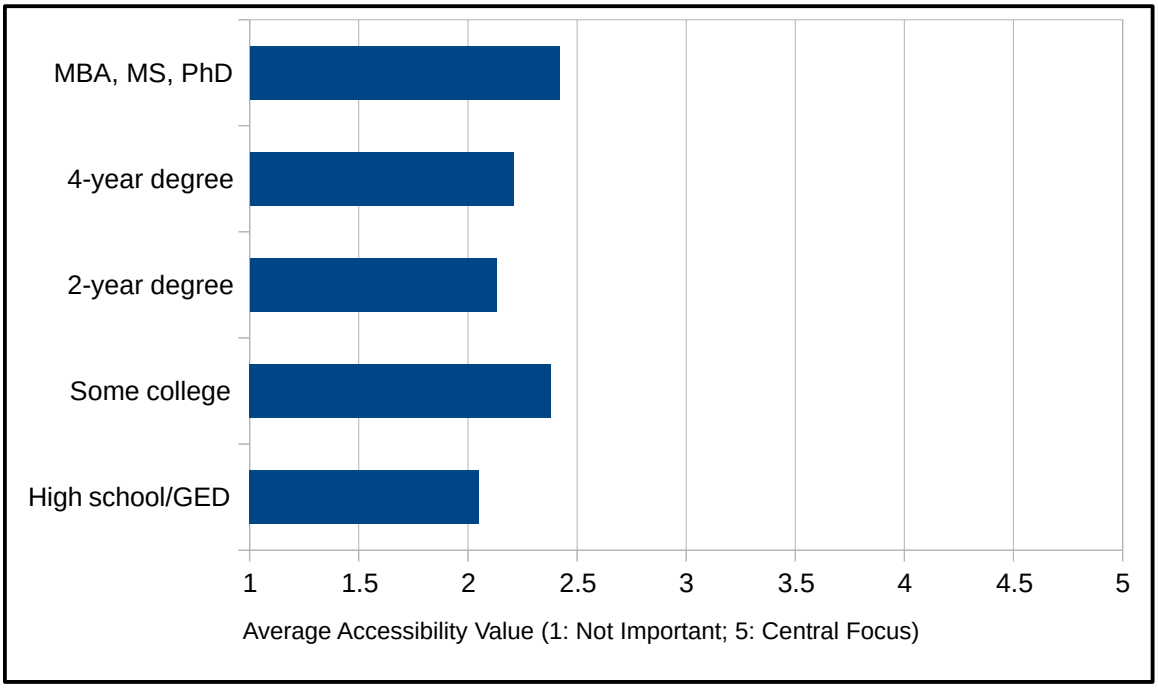

Figure 12. Hypothesis 6 (Aggregate)

6.3.7 Hypothesis 7: Developers that attended a technology-focused college are more likely to value having their websites work for users with impairments

This hypothesis was founded on the idea that technology-focused schools would have a better understanding of, and appreciation for, standards compliance generally, which could include accessibility standards. However, the results do not bear out this hypothesis. The results are not statistically significant, and there is not even a weak correlation between tech schools and accessibility, when compared to liberal arts schools and community colleges.

There is a weak correlation between community colleges and programming for accessibility, which is an element that I explored in greater depth in my interviews, since the sample size was so small in my survey data (3%). Several of my interview subjects, who attended community colleges, indicated that they received a better education on a variety of topics—including accessibility—than their peers who attended a liberal arts or tech school. The reason they gave was that tech and liberal arts schools tend to have courses that are taught by career academics, who are removed from the daily realities of web work, and tend to be unable to stay current on methodologies and approaches to modern web design and development. By contrast, community college courses are often taught by working professionals, or recently retired professionals, who are able to give instruction more closely tied to the daily realities of modern web design and development. The number of people who indicated that they attended

trade schools such as ITT Tech was very small—just 7% of those that answered this question. However, it appears that web designers and developers that attended trade schools received significantly less accessibility education, which warrants further investigation.

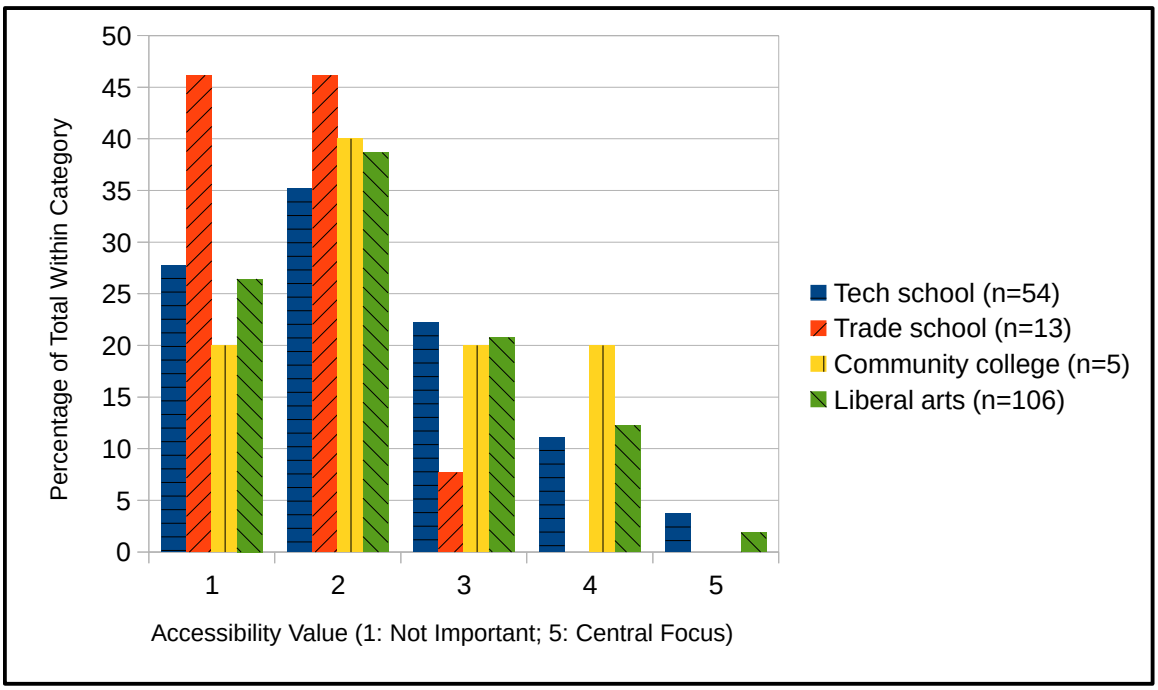

Figure 13. Hypothesis 7

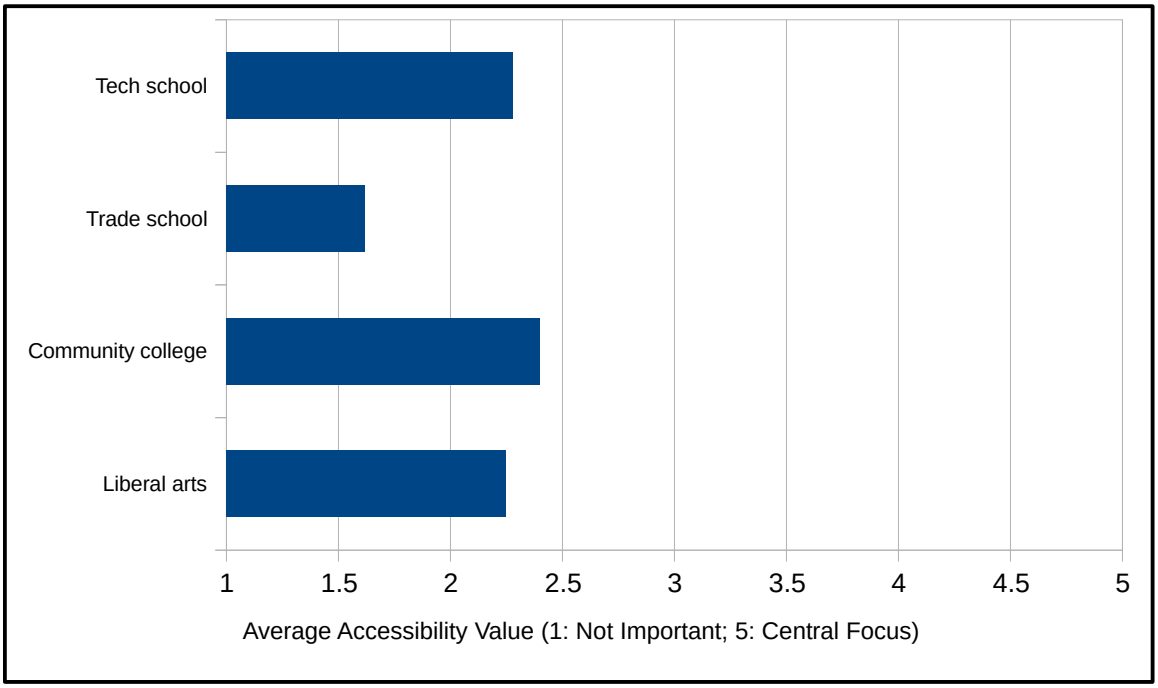

Figure 14. Hypothesis 7 (Aggregate)

6.3.8 Hypothesis 8: An emphasis on user groups in college translates to an emphasis on user groups in work practice

This hypothesis was intended to be rather straightforward—if the practice of raising awareness of social concerns in development during postsecondary education is effective, then I would expect to see a direct correlation between designers and developers that were exposed to an emphasis on user groups in college and a corresponding emphasis on user groups in their work practice.

Instead, I discovered that there is no statistically significant correlation between educational programs that emphasize user testing and the work product of the designers and developers that graduate from those programs. The results for the group that answered that user groups were a central focus in college was too small to be considered statistically significant within the sample (n=4).

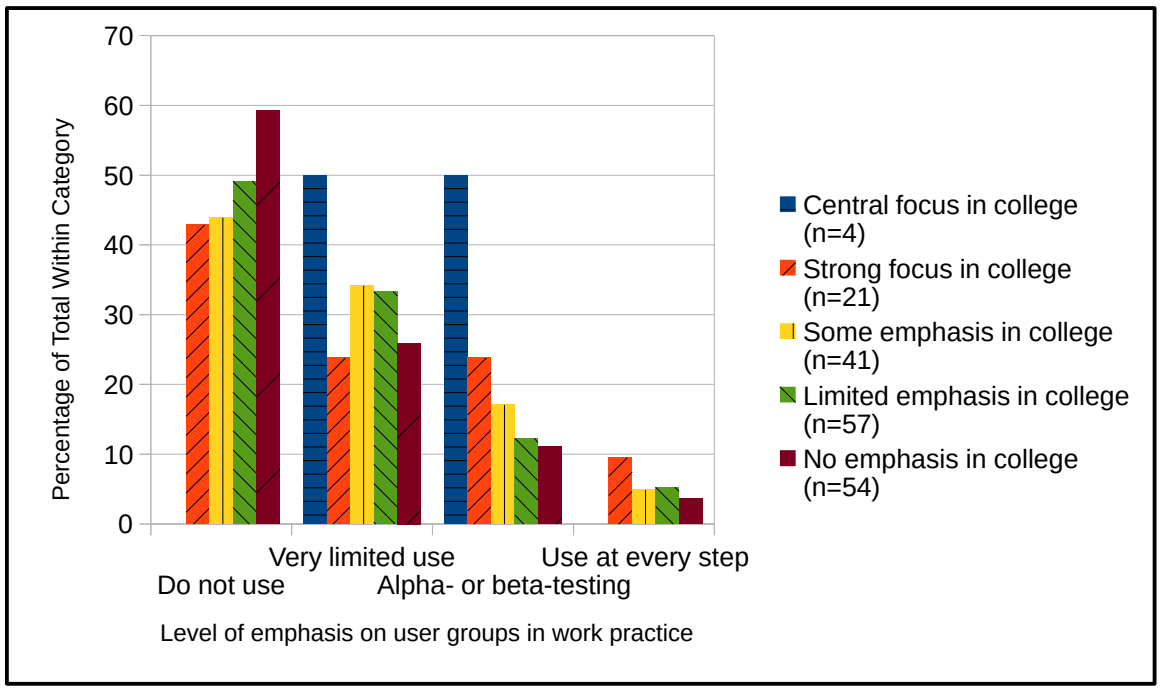

Figure 15. Hypothesis 8

6.3.9 Hypothesis 9: An emphasis on designing for accessibility in college translates to an emphasis on designing for accessibility in work practice

Similar to the previous hypothesis, I posited that if accessibility was taught in college programs, graduates of those programs would be more likely to incorporate accessibility at some level into their work practices. As with the previous hypothesis, I did not find a statistically significant difference between graduates of programs where accessibility was emphasized versus graduates of programs where accessibility was not included, or was only mentioned in passing.

However, when analyzing the aggregate graph, there appears to be a correlation between emphasis in college and emphasis in work practice, with the exception of those indicating a central focus in college, which can likely be explained by the very small sample size (n=3). Conducting the survey with a larger group might yield statistically significant results. It is also possible that there is a bell curve of effectiveness when it comes to accessibility education—too little and it doesn't stick; too much and it breeds resentment. Targeted interviews with graduates of programs that have a strong accessibility focus could shed more light on this subject.

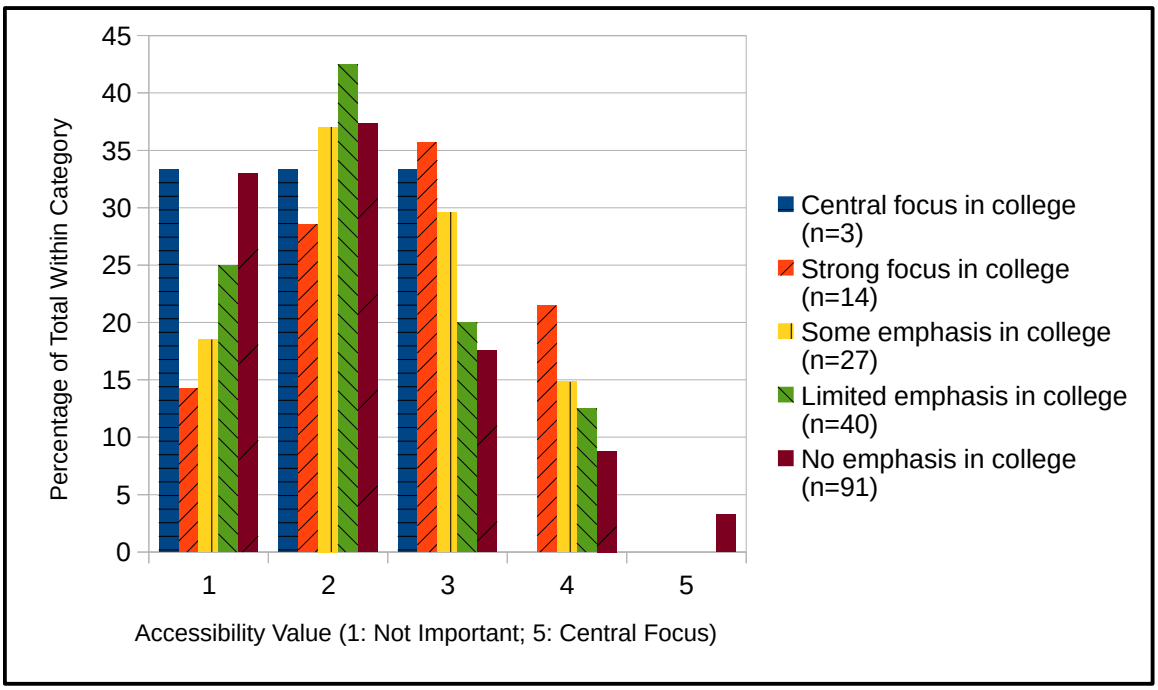

Figure 16. Hypothesis 9

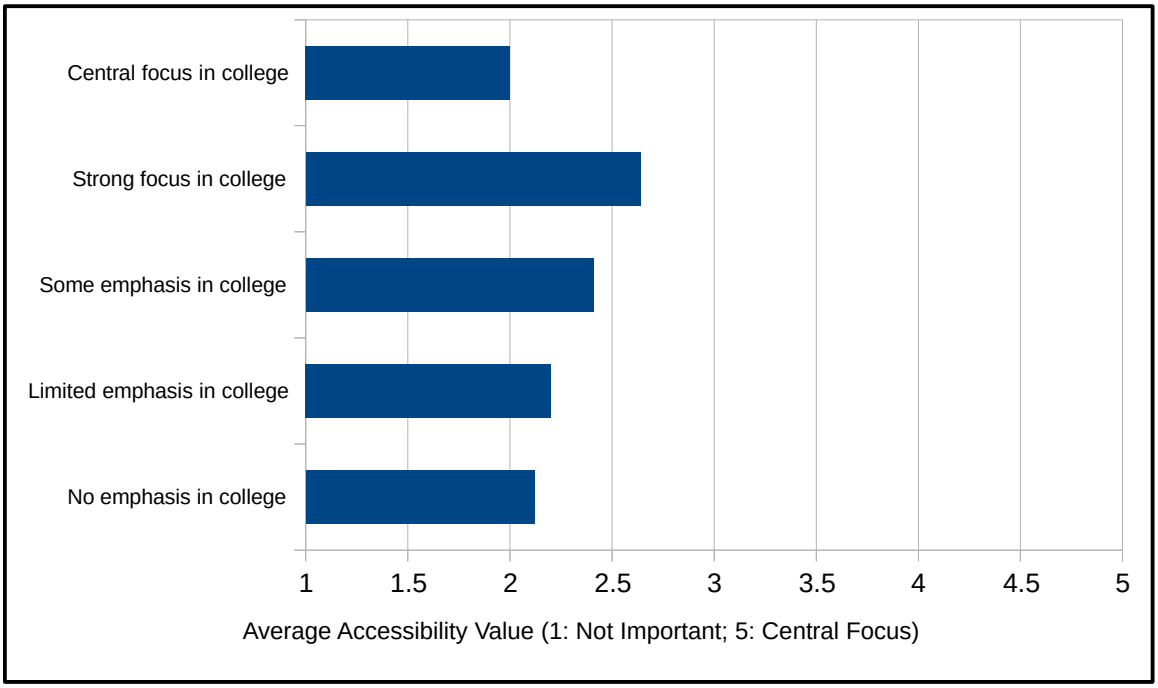

Figure 17. Hypothesis 9 (Aggregate)

6.3.10 Hypothesis 10: There is an inverse relationship between valuing creativity and valuing accessibility

> "Some accessibility sites are downright ugly, but the problem lies with those sites' designers and not with accessibility, which carries no visual penalty."
> (Zeldman 2003, 98) as quoted in (Kennedy 2012, 115)

I hypothesized that there was a perception among designers and full-stack developers that there was a tradeoff between accessibility and visual design—in order to make a site accessible, certain elements of the visual design needed to be abandoned or compromised. Zeldman's quote notwithstanding, accessibility standards do place some constraints or “visual penalties” on the design of a site. For example, design is constrained in order to preserve readability through adequate color contrast, and through necessitating the use of multiple indicators for action, such as using both color and underlined text to indicate hyperlinks. Zeldman's point is that these restrictions are relatively minor, where they do exist, and tend to be part of general usability of the site rather than accommodations made specifically for people with impairments. It is incumbent upon designers to work within the standards to make something beautiful that is also accessible.

When analyzing my responses, I found no statistically significant difference between designers and developers that said that they value creativity and those that value accessibility. In fact, the aggregate data points to a direct correlation instead of an inverse correlation—designers and developers

concerned with creativity are *more* likely to value accessibility, not less.

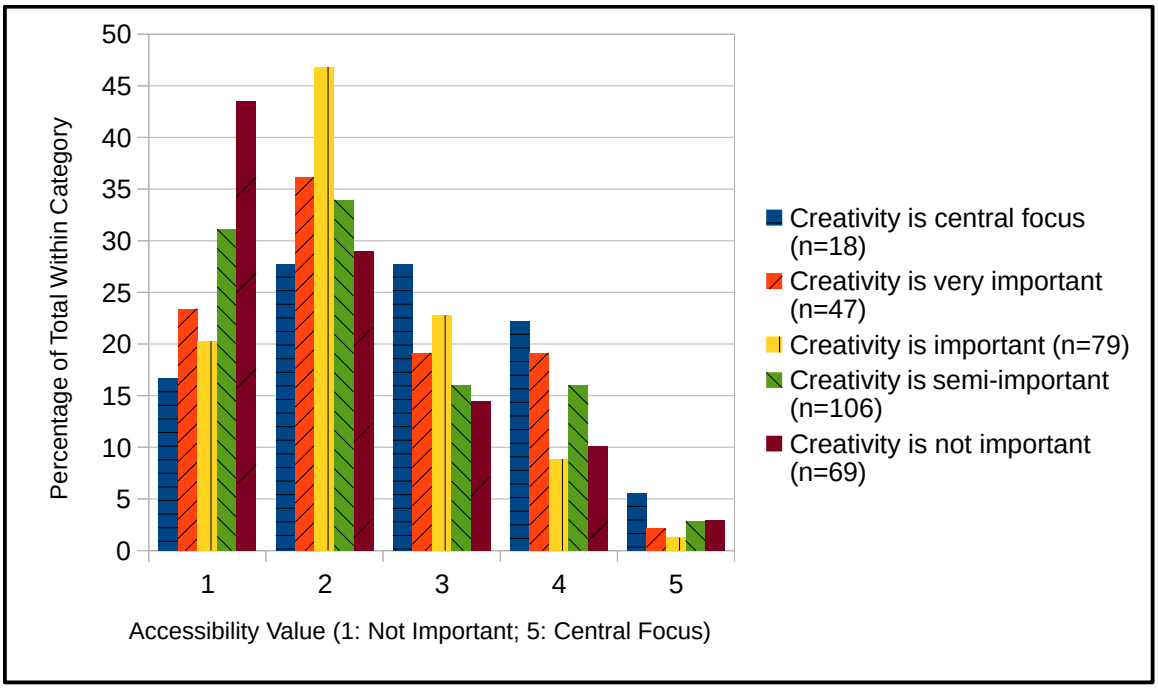

Figure 18. Hypothesis 10

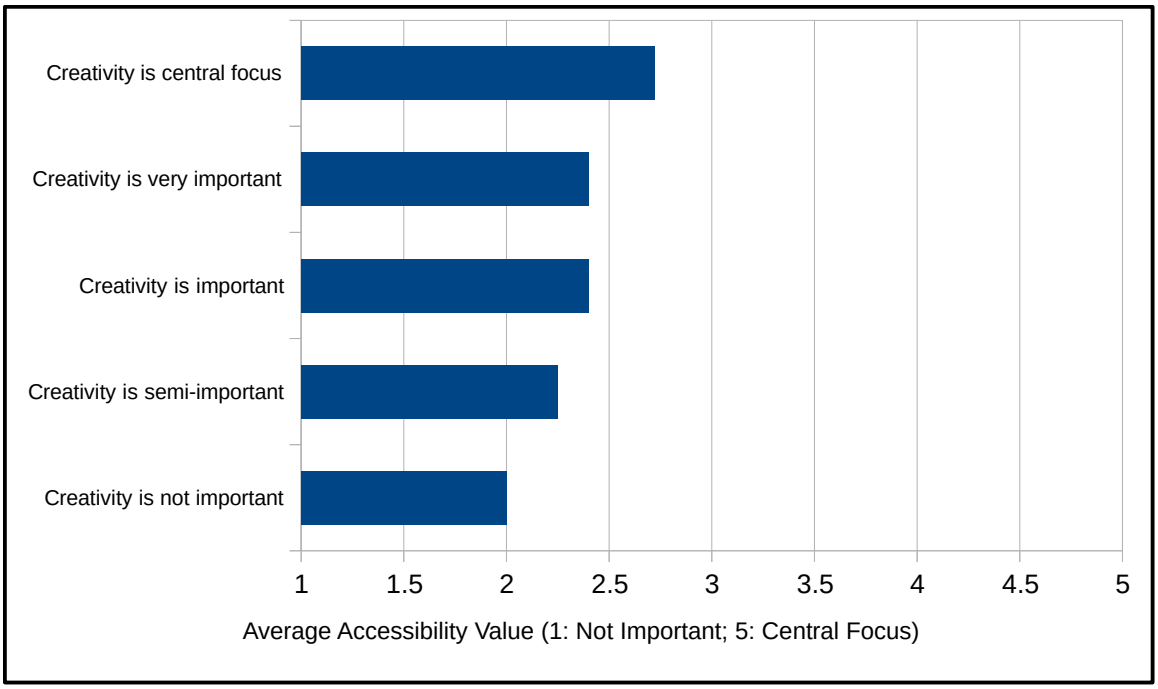

Figure 19. Hypothesis 10 (Aggregate)

6.3.11 Hypothesis 11: Developers in countries with explicit disability laws are more likely to value accessibility than developers in the USA

This hypothesis directly tests my inclination that one of the main differences between my study and Kennedy's study was the focus on US versus UK, and the fact that there is not an explicit general web accessibility law in the US like there is in the UK. If this hypothesis proved true, it would demonstrate that attitudes about accessibility in the UK and other countries with explicit disability laws differ from attitudes about accessibility in the US, which may have something to do with accessibility being a legal requirement. As with the previous hypotheses, the results were not statistically significant. However, the aggregate graph shows the highest average for developers in the United States, which is the opposite of what I expected.

Most of my interviewees indicated that they do not design and code for accessibility most of the time, which is at odds with both the responses given by Kennedy's interviewees and focus groups and with my own survey data. A larger quantitative study of designers and developers in the US and the UK may demonstrate statistically significant differences in opinions and practice between the two countries, and may help to substantiate the differences in qualitative findings between myself and Kennedy.

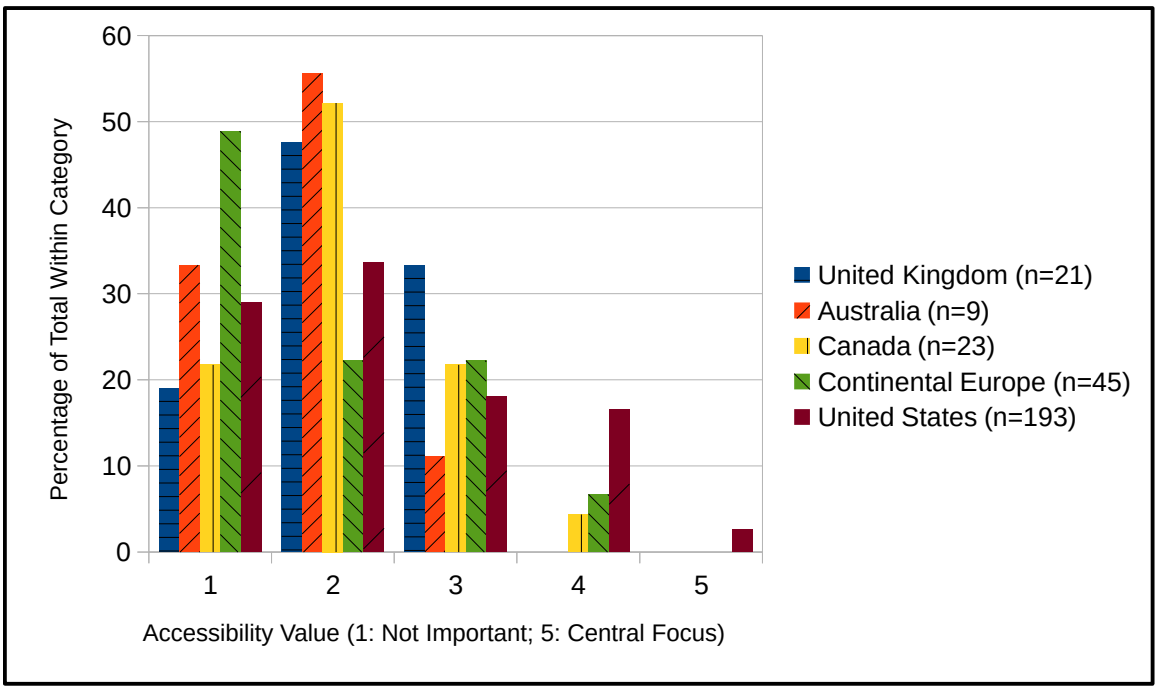

Figure 20. Hypothesis 11

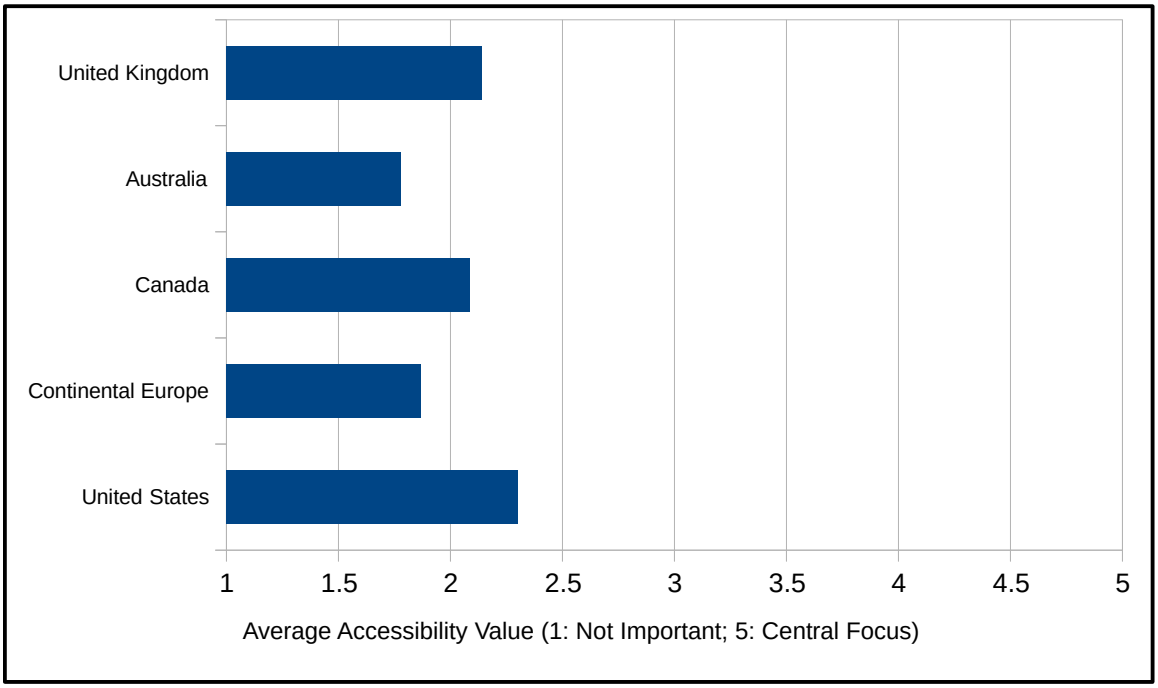

Figure 21. Hypothesis 11 (Aggregate)

A different way to look at these data, particularly in absence of statistical significance in support of my hypotheses outlined above, is to examine the behaviors of the subset of respondents that valued accessibility and look for commonalities. For this task, I selected the respondents that answered that designing and programming for accessibility was “very important,” or a 4 on a scale of 1-5. As stated earlier, I would expect that the only respondents that would answer 5, or “central focus,” would be those that have a connection to accessibility either personally or through the type of business they work for. Therefore, those answering that accessibility is “very important” are the best group to analyze to pinpoint typical behaviors for designers and developers that value accessibility. Members of that group were:

- More than twice as likely to participate in professional development activities than those that indicated that accessibility was not very important (31% vs. 13%)
- Slightly more likely to code for W3C standards compliance than those that indicated that accessibility was not very important (82% vs. 77%)
- Evenly spread over the type of business they work for (company primarily that makes websites, company that does not primarily make websites, freelancer or sole proprietorship)
- Evenly spread across job responsibilities (all aspects, back-end only, front-end only)

- More likely to be experienced—only 2 of 45 had less than 2 years of experience
- As likely as not to have a college degree
- Of those that went to college, as likely as not to have utilized user groups in college, and only 60% indicated that their college courses had emphasized accessibility
- Split on valuing creativity—about half (44%) said it was important, and half (56%) said it was not
- Overwhelmingly from the US—out of the countries where there are explicit disability laws, there were no respondents from the UK or Australia, only 1 from Canada, 3 from continental Europe, and the remaining 32 were from the United States

Although there are some strong signals within this group, particularly as relates to amount of experience and ongoing professional development, it appears that the influence of education is not consistent or direct. Following Kennedy's assessment, the ethical self-regulation of web designers and developers through ongoing professional development and ordinary experience-gaining by following leaders in the field appears to have the greatest influence on individual behavior as it relates to accessibility, which is in line with my interview findings.

6.4 The Road to Inaccessibility is Paved with Good Intentions

One of Kennedy's core arguments is that web designers and developers are ideologically committed to web accessibility. However, there is a gulf between ideological commitment and action, as Kennedy acknowledges in her conclusion, and as my informants indicated in interviews. It is largely agreed that accessibility is a part of web design and development best practices, but accessibility is not well-understood, and is typically one of the first items to be cut from a project in order to save time and money. Additionally, when accessibility is included in a project, it is typically included by virtue of adherence to accessibility standards, rather than doing requirements analysis and beta testing using people with impairments in testing groups.

The primary takeaway from my analysis of the survey data was that most people do not design or develop for accessibility, irrespective of educational background, nation of employment, role within an organization, or ideological commitment to creativity. This finding is at odds with Kennedy's analysis of her survey data, interviews, and focus groups. It is important to note, however, that Kennedy was merely asking whether her informants were ideologically committed to accessibility, not whether they employed accessibility practices in their daily work. Additionally, some of Kennedy's informants had not yet entered the workforce, and thus could not comment on their actual work practices.

> “Another limitation of this book is that it has prioritized practices over products. For example, it has taken

> web designers' verbal commitment to accessibility as evidence of accessible practice, rather than testing the websites that these designers produce for their accessibility." (Kennedy 2012, 215)

Irrespective of a stated commitment to producing accessible sites, designers and developers that do not utilize user testing with users with impairments will not know whether their sites are actually accessible for people with impairments. Accessibility standards are a great first step, but are only one part of a comprehensive strategy for producing websites that are highly usable by web users with impairments.

> "It is only by going through the WCAG guidance manually and testing the site with people with a range of physical, sensory and cognitive disabilities that a web designer can know whether a website is accessible or not. The accessibility of a website needs to be tested by people with disabilities, and this is much more complex than simply inserting a URL into a text field and clicking a button." (Kennedy 2012, 119)

The findings from my survey data indicated that most designers and developers do not include people with impairments in the planning or testing phases of projects. Additionally, my interviewees indicated that accessibility

testing, particularly with users with impairments, is not a priority unless it is a mandated component of the project by the client. Since clients and managers are often the final arbiters of what high-level components are included in website projects, it is essential to convince clients and managers that accessibility is important, and that accessibility testing in particular is critical to ensure accessibility of the end product.

> “Clients or line managers are likely to control budgets, and, whereas accessible web design itself is not costly, accessibility testing can carry financial costs, and this is, as noted in the previous chapter, the only real guarantee of accessibility.” (Kennedy 2012, 147)

In her conclusion, Kennedy discusses the importance of broadening understanding of the need for accessibility to project team members beyond just designers and developers. It is important for designers and developers to know how to make accessible sites, and that making accessible sites is important, but account executives need to be able to communicate that importance to clients as well. If clients do not approve of the effort required to include accessibility components, and any potential sacrifices in visual display for some groups of users to promote accessibility and ease of use in others, then the prospects for routine inclusion of accessibility in web projects are poor.

For societies that have an egalitarian commitment to treating everyone equally, accessibility should be a matter of ethical imperative. However,

conversations about including accessibility often devolve into capitalistic evaluations of return on investment that are at odds with egalitarian values. Inclusion of accessible design practices are pitted against an adverse effect on the project budget and estimated completion date, mediated by the anticipated increase in sales by users with impairments that would otherwise abandon the website if they found it to be unusable. Such calculations are oversimplifications of the process for including accessibility components, and do not include the numerous additional benefits of accessible design methodology that extend beyond traditional understandings of what it means to be a web user with an impairment.

7. SYNERGISTIC ENABLEMENT

In previous chapters, I have alluded to the idea that accessibility, when practiced, is often tied to other factors—search engine optimization, HTML and CSS standards compliance, framework or platform choice, initial load speed, and on-page performance. From my research, I have found that accessibility practiced primarily for the purpose of making a website accessible for people with impairments is rare, unless specifically mandated by law or client requirements. Instead, most often, accessibility is a ride-along with other concerns. My term for this ride-along effect is “synergistic enablement”—the synergy between various technological and social factors in design and development produces a net effect that is enabling to users with impairments in addition to other forms of accessibility not related to physical or cognitive impairment, such as speed and performance on mobile networks and understandability by search engines and software.

The other concerns that I mentioned above, especially search engine optimization and on-page performance, are often held in higher regard by clients and managers because they are more directly translatable to the bottom line than accessibility for a general audience. These other factors become part of the cost-benefit calculation of what goes into a website build based on the anticipated return on investment in both time and money. Anything that isn't perceived as being “worth it” will get cut, which often includes accessibility, when advanced as an end unto itself. This capitalistic argument is a clear example of the most typical manifestation of ableism in web development.

The concept of synergistic enablement is that accessibility for people with impairments can more reliably be included in projects by emphasizing the other benefits of good coding and design practice, including coding and design for accessibility, that appeal most directly to clients and managers. Conversely, if accessibility is required in frameworks or platforms due to direct or indirect legal or regulatory pressure, anyone else that uses those frameworks or platforms will be able to reap the rewards of the work that was done to make them accessible without having to work to build accessibility components in their own projects from scratch.

> The large fonts, simple language and deep, rather than wide, structures required for people with [intellectual disabilities] impact on the usability of a website more than writing code in such a way that it can be read by a screenreader. Conversely, good markup for sensory/physical disability will often result in good search engine optimization, quick downloads, or cross-platform compatibility, and so accessibility for people with sensory or physical disabilities has other pay-offs beyond accessibility. (Kennedy 2012, 150)

Web designers and developers who are committed to accessibility and fulfill the role of the accessibility advocate are a type of activist within their organizations. These activists need to use a range of strategic argumentation

strategies in order to win over decision makers within and without their organizations in order to support normalizing accessible practices. According to Kim Fortun, there are three methods for activists to achieve change—legislative, legal, and “people power” (Fortun 2001, 303). Legislative and legal approaches to web accessibility activism have been tried, and have had some successes, although arguably less than the impact of activism on modification of physical spaces to make them accessible. However, these successes have not had an impact that is anywhere near the scale of the impact of industry shifts toward more accessible practices for other reasons entirely, such as search engine optimization and overall website speed and performance, which are examples of synergistic enablement and are not the result of activist strategies. This is not to say that accessibility should not be pursued as a goal in and of itself, but rather that accessibility is one concern among many—one that, by itself, is unlikely to sway decision makers producing websites for general consumption, but could be used in conjunction with other factors to convince clients and managers to support more usable practices overall.

Viewed through this lens, accessibility practices, combined with other valid concerns from project stakeholders focused on other goals for a website, can be understood as a form of “appropriate technology,” which is “designed to fit into its local setting, synchronized with available material resources, expertise, and labor time” (Fortun 2004, 54). It is easy to argue that large corporations and governments should invest the time in accessible design and user testing, as

many prominent accessibility court cases have done. It is much more difficult for advocates of greater accessibility to convince policymakers and website owners that every website for every business or organization, irrespective of size, should meet the same rigorous standard. As such, an appropriate technology frame is useful for analyzing how best to incorporate accessibility into websites that are built with fewer material resources and less available labor time.

Additionally, the level of expertise of an individual designer or developer is a factor in the inclusion of accessibility components. In previous chapters, I quoted Kennedy's findings that many web designers and developers are not intimately familiar with the accessibility standards, and it would require several additional years of formal education as well as ongoing training in order for them to be, given the complexity of the standards themselves. It is difficult for a designer or developer to have a full command of the standards unless they specialize in making websites accessible, instead of being a web design and development generalist. Inside of large organizations, such specialization is possible—and even desirable—but it would be extremely costly to employ an accessibility expert on every web project, which disproportionately affects the budgets of web projects for small clients with limited resources.

Given these constraints, what can be done to make the web more accessible for people with impairments who want to be able to find out when the local coffee shop is open, or want to purchase a product from the e-commerce store of a small business located on the other side of the country? The local

coffee shop may have a budget of only a few hundred dollars to develop its entire web presence, which is barely sufficient to produce a basic website presenting limited information—much less to employ an accessibility expert, validate the website against accessibility standards, and employ user testing groups consisting of people with impairments.

The appropriate technology movement began in the 1960s as a way to argue for the use of particular technologies in the developing world that would not cause adverse effects in a local setting that was markedly different from the developed world (Winner 1988, 61–3). The types of technological intervention in the developed world are not necessarily appropriate for the context of the developing world. Likewise, the types of technological intervention to ensure accessible websites that are appropriate for large businesses, organizations, and governments are not necessarily appropriate in the local context of small organizations and businesses with limited resources. What is appropriate to expect of large corporations and governments is not appropriate to expect of small businesses, nonprofits, sole proprietorships, and the like. A different approach is needed to address accessibility at that level without causing undue hardship on those organizations.

I argue that synergistic enablement is a way forward for these cases. By focusing activism strategically on particular points, the overall accessibility of much of the web can be improved in significant ways, by playing to the priorities of designers, developers, clients, and managers.

7.1 Synergy with Platforms and Frameworks

Perhaps the most important intersection for synergistic enablement is engaging accessibility at the level of a platform or framework. This includes front-end coding frameworks, such as Twitter Bootstrap and Foundation, JavaScript frameworks such as Angular and React, code inspection utilities that are part of build toolchains, content management systems such as Wordpress and Drupal, and cloud-based publishing platforms such as Squarespace, YouTube, and even Facebook.

> “Another explanation given for the limited support for standards relates to authoring tools. Although there has been some success in persuading browser makers to support standards, progress has been slower with authoring tool manufacturers.” (Kennedy 2012, 103)

As the tools and practices of web development have become more varied and complex, requiring more specialized knowledge and training, barriers to entry have increased. Therefore, the subset of the web publishing population that would have used authoring tools such as Adobe Dreamweaver in the late 20th century have now mostly moved to using cloud-based solutions such as WordPress.com, Squarespace, and Facebook Pages. Kennedy's argument is primarily about traditional authoring tools, such as Adobe Dreamweaver. These tools still have a place, but they are rapidly losing market share to the rise of

easy-to-use hosted cloud solutions.

Perhaps the local coffee shop determines that it doesn't need to pay a web designer and developer to create a unique website at a unique URL—the business owner (or a web-savvy relative) can simply create a Facebook page for the business, upload some photos, get followers, and post periodic updates. There are even fields on Facebook pages for posting hours of operation, contact information, and so on. Under this paradigm, accessibility knowledge of a designer or developer is irrelevant, since the business owner is maintaining their own web presence using a platform, bypassing a direct relationship with a designer and developer completely.

Therefore, the primary intervention should not be at the level of the content creator—in this example, the coffee shop owner—but rather at the level of the company or organization responsible for developing the platform. If Facebook Pages are inaccessible, for example, or they make it difficult or unintuitive to include accessibility components, then web presences created on Facebook will continue to present accessibility challenges to a large number of Internet users with impairments. By contrast, there is an opportunity to improve the accessibility of a large number of small business sites by improving the accessibility of Facebook Pages, and making it intuitive for content creators and site administrators to make their content accessible.

Squarespace is another such publishing platform. Squarespace allows users to create websites using an intuitive point-and-click interface, choosing

from a list of pre-built themes to determine visual display and flow. Non-technical users can easily create a great looking (if slightly formulaic) website that works well across a range of devices, operating systems, and web browsers, and utilizes responsive design methodology to present an optimized version of the page for users on small screens, such as smartphones. Again, non-technical users with limited to no understanding of accessibility techniques, and limited ability to make changes to the Squarespace platform, are at the mercy of the Squarespace developers to make the platform accessible and make the inclusion of accessibility components intuitive.

WordPress.com is an interesting example, in that it is a for-profit cloud-hosted version of the open source WordPress content management system. Tech-savvy users can download and install the WordPress CMS on their own websites, but less tech-savvy users will often opt to create a website using the intuitive point-and-click interface of WordPress.com. Similar to Squarespace, there are a limited set of themes that users can select from in order to determine the visual appearance of their website. What makes WordPress.com interesting is that changes that take place in the open source WordPress project will make their way into the for-profit WordPress.com offering, even though some of the code on WordPress.com is proprietary and closed-source. Unlike Facebook Pages or Squarespace sites, the open source community has an ability to modify the way the WordPress.com site behaves through contributions. I will explore this topic in more detail later in this chapter.

WordPress, as a platform, is regarded as being one of the more accessible CMS offerings available, and includes features that make it somewhat more intuitive to develop accessible sites.

> “[WordPress] will add titles to links, for example, and it will automatically tag things for you using the WYSIWYG editor, so that you really don’t have to think about your document structure, because the weight that you give it when you’re writing it, WordPress handles for you—and, theoretically, is the weight that it should have when you’re done writing.”
>
> —Informant A

WordPress doesn't just include an intuitive way to include accessibility components. It also includes intuitive methods for addressing other concerns, such as search engine optimization and integration with third party services, such as automatically serving images and descriptive text to social networks like Facebook when links are posted.

> “Tools like, for example, the WordPress Content Management System. Tools like Google Webmaster Tools to describe pages—make sure that, at a very high level, your pages have a title and a description, to kind of give it some context. I think those are a couple of the big structural things. And on the

> technical side of things, what it does is it allows us to represent our content, ultimately, once it's been generated, in a way that machines, anywhere from search engines, but also more towards screen readers and devices that have the ability to read off content on a page—it gives them a good, sound base from which to start from."
>
> —Informant B

However, the WordPress ecosystem is complex. There are three main components that are part of a WordPress site—WordPress core, which is the code of the CMS itself; the theme, which is the code that controls what the website looks like to visitors; and plugins, which extend the functionality of WordPress core, and can contribute to the visual display of the site. The only component of these three that is under the control of the WordPress open source community is the core software offering, which governs the use of the administrative interface for publishing and editing content, and the starter themes that ship with the core software. It is the responsibility of individual theme or plugin authors to ensure that their display, content, and interfaces are accessible. Understandably, the accessibility level of individual themes and plugins varies wildly, and can result in transforming what is otherwise a very accessible platform into a website experience that is highly inaccessible.

To this end, developers that work on the "front end" of a website—the part

that end users see, rather than the administrative or “back end” component used to publish and edit content—often employ the use of front-end frameworks to assist with developing what are typically complex interfaces that need to work reliably across a variety of devices, operating systems, and browsers. As I discussed in chapter 3, the ecosystem of web development has become significantly more complex over the past two decades, and front-end frameworks go a long way toward making it easier to write code that renders reliably across a range of use cases.

Front-end frameworks come with tradeoffs. Being able to rapidly write code that is reliable and consistent means that developers have to take the time to learn the framework, and to learn the best way to interact with the framework. Oftentimes, frameworks are “opinionated,” meaning that they have a particular way that they are intended to be used, and a particular way that they are designed to appear and function. As a consequence, designers and developers may have to give up or scale back their own personal opinions about certain elements in order to reap the benefits of the framework.

> “So the constraint really comes from, I want to solve this problem, and how do I do it using their syntax and their workflow versus how I want to do it. And a lot of times—most of the time—once I figure out their opinions and their workflow, it’s way faster for me. And to that end, I am willing to give up some of the

> freedom I have to solve a problem because it just saves me time. And I don't really view the constraints as a hindrance, because I might be giving up, you know, a look, or I may not be using a container that's a certain width that I want, because it's easier for me to use their container width, or their JavaScript."
>
> —Informant A

In addition to time savings and standardization, front-end frameworks open the door to the possibility of greater inclusion of accessibility components and features. For example, the work that has been done by the open source Twitter Bootstrap community over the past few years to make the framework accessible has had far-reaching effects. According to the Twitter Bootstrap website, Bootstrap powers millions of websites around the world (Twitter Bootstrap 2015), all of which are able to benefit from built-in accessibility components.

The accessibility features of Bootstrap and other frameworks are not, and cannot be, fully automatic. However, by including accessibility components where possible as part of the default installation, and by including detailed instructions on how to work with the framework in a way that is inclusive of accessibility, it makes it easier for developers to write accessible code using the framework, even if they did not previously have knowledge of accessibility best practices and standards.

Richards et al. argue that the majority of accessibility gains in the 21st century have arisen “as side effects of changes in Web technology and associated shifts in the way Web pages are designed and coded” (Richards, Montague, and Hanson 2012). In part, this is due to platforms and frameworks for making webpages, but also includes advances in assistive technologies at the browser and operating system level that are better able to present otherwise inaccessible content in ways that do not rely on the developer to code the site in an accessible way.

Fostering and promoting accessibility practices within frameworks and platforms is part of what Kennedy terms the “ethical self-regulation” of web designers and developers, and can be done without legal or regulatory requirements being imposed on framework maintainers or platform owners. However, if legal or regulatory requirements were to be applied, targeting the platform or framework level would have the broadest impact.

7.2 Synergy with Search Engine Optimization (SEO)

> “The argument of having good search engine optimization goes a long way. Everyone wants their search results at the top. If I can say, “well, we need to do this to improve our SEO,” people are more willing to listen to that. It’s really interesting, because the end result is the same. You want to reach a large number of people, as large a number of people as

> possible, so you improve your search engine optimization. However, if you put it in the context that, "well, we need to ensure that people with disabilities can access this," they don't want to hear that as much, which is fascinating. They are alienating those same people that you could reach with that good SEO. It tells you where people's heads and hearts are."
>
> —Informant K

The above quote, given by an informant who trains designers, developers, and content creators about how to make accessible websites, is illustrative of the motivations of many decision-makers in the industry. Arguing in favor of accessibility primarily for the sake of providing access for users with impairments generally does not convince clients and managers to spend the time and money required to make their sites accessible. However, if an argument was made in favor of writing the same or similar code for the purpose of being accessible to a different type of user—namely, search engine spiders—clients and managers would tend to approve the work. In capitalistic terms, the distinction between the two motivations is between trying to reach a large audience by making the website easier to find via organic search—which is easier to translate into monetary terms to justify the investment—versus spending time and money optimizing the site for the relatively small percentage of the population that has

an impairment.

> “I mean, just the basics of accessibility with having semantically tagged things, and having everything available clearly in the HTML, that’s going to help Google see what’s going on, because the Googlebot basically sees what a screen reader sees, because it’s just crawling the code. So you keep it accessible, you get better SEO.”
>
> —Informant I

The reason for this similarity should be unsurprising. Much of the work that is done to make a website accessible involves ensuring that the content is accessible to machines—specifically, assistive technologies. Search engine spiders are another type of machine. The problem with using images to convey information is that one type of assistive technology—a screen reader—is incapable of discerning meaning from the image without a cue from the developer or content creator in the form of “alt” text that accompanies the image. Likewise, a search engine spider is not capable of discerning a different type of meaning from the image unless the same text is present. For example, if you were to perform an image search on Google, there are a number of “signals” that contribute to what images match what search text, including any associated captions, “alt” text, or long description paired with the image. These techniques make the content accessible to both search engine spiders and assistive

technology for the blind.

Likewise, writing meaningful titles and content in easy to understand language makes the site accessible to people with certain forms of cognitive impairment as well as making a better connection between what average users are searching for and what is present on a website. Providing transcripts or captions for video allows deaf or hard of hearing users to understand what is present in the video, but also allows search engines and video hosting sites to index the full content of the audio in the video, instead of limited signals such as title and description. Therefore, some aspects of accessible design and development have become part of routine development practices primarily due to their positive impact on SEO.

> “I guess on the coding side, you know, we try to make sure that things are accurately tagged, and a lot of that I think is mainly for SEO purposes, probably, but also has the added benefit of working well for a screen reader. I don’t find that we spend as much time on that. In the past, when I worked at a government agency, that was much more of a concern, and more of a requirement, at that point in time. But it’s not something that we really have a mandate to do. Something that we’re interested in doing, you know. We want to make sure that the site

> is more readable for people with disabilities as well as machines, like bots that are scraping the page, and I think those two things kind of go hand-in-hand sometimes."
>
> —Informant F

The primary underpinnings of modern SEO are writing site content using the language the user utilizes when searching, and making the content clear and straightforward. By doing so, a content creator will make it easier for a user to utilize a search engine to match keywords to the content on their site, and to be incentivized to stay on the site afterward due to the presence of valuable content. This effect is amplified if the keywords are in the page title, which appears as the primary link in a search engine results page, since it is usually the most prominent element that a user sees when scanning the results page. Alternately, if the keywords are in semantically tagged headings, these headings communicate the relative importance of those keywords to the search engine, which can display those headings as additional links within the results page.

> "I think it all overlaps. In terms of search engines, Google's pretty good about making you do things right. Say what you will about Google, but if you're going to have a good, optimized site, you're going to do everything you can do correctly, if you're going to rank really well."

—Informant L

These techniques also have positive effects on accessibility. Well written hierarchical headings make it easier for blind users with screen readers to skip to the most relevant section of content, and clear language aids people with intellectual impairments that are trying to make sense of the content on a site. In this context, there is a synergistic effect between SEO, best practices design and development, and accessibility.

> “Yes, the logo, visually, is important, but from Google’s perspective, it’s the actual content in the page that’s important, right? We try, to the degree that we can, to reorganize the HTML so that the content is towards the top of the page, and the header and the footer can be moved with CSS to be in the right place visually, helps both accessibility and SEO, because it helps Google find the important stuff faster.”
>
> —Informant N

The fastest way for any type of machine—search engine spider, screen reader, etc—to interpret the content of a webpage is to read the HTML code directly. This approach does not require the use of a web browser at all, since the content is present in plain text. This approach is what the above quote references —ensuring that the order of the content in the HTML makes sense to screen readers and search engine spiders, by putting the most relevant pieces of

content first in the HTML code. Display-layer technologies such as CSS and JavaScript can then be utilized to move the content to a more appropriate place visually on the page within the browser while maintaining the appropriate content order in the raw HTML. Again, this technique has benefits for both search engine optimization and assistive technologies such as screen readers.

However, what is gained in speed is sacrificed in interpretability. Reading the HTML directly is an excellent approach, if the HTML is representative of what is actually on the user's screen. With the rise of JavaScript frameworks that load a minimum of content in the initial HTML that is served to the browser, and then use JavaScript code to fetch additional content and dynamically insert it onto the page, this assumption is increasingly invalid, and is the source of accessibility and SEO challenges with the use of JavaScript frameworks. Facebook, for example, uses a JavaScript framework that it developed, called React, to power the “infinite scroll” functionality on the news feed, among other things. In order for assistive technology, such as a screen reader, to be able to interpret the content of a Facebook news feed, it needs to be able to keep up with all of the dynamic content insertion that is occurring due to the JavaScript framework code.

When using a JavaScript framework, the user is typically served a bare-bones page in HTML that loads the JavaScript framework. Once the framework is loaded, it fetches the relevant information from a server on the Internet and injects the content into the page. Therefore, reading the raw HTML at load-time is virtually useless, because the content is not delivered until after the JavaScript

code is executed and the content is fetched asynchronously from a content server. Thus, asynchronous websites present an accessibility challenge for the current generation of web development methodologies, which will likely need to be solved by using web browsers as mediators between application code (which runs on websites) and interpretive code (search engine spiders and assistive technologies). Search engines and assistive technologies could benefit from greater synergies in this arena by agreeing upon methods for interpreting and surfacing content based on an emerging set of public standards that web developers could implement within their own projects.

7.3 Synergy with Law and Regulation

As Kennedy discovered in her survey of web designers and developers, the law and regulation were cited as the least prominent driver for making accessible sites (Kennedy 2012, 124). However, there is a difference between individual designers and developers citing law and regulation as drivers for individual or small team action versus the effect of law and regulation on larger entities.

> “Although Section 508 does not require private sector technology companies to conform to the standards, it motivates them to do so, given that Federal agencies are required to purchase products from companies which adhere to accessibility standards.” (Kennedy 2012, 124)

This quote from Kennedy illustrates the synergistic effect between accessibility laws and the development of accessible platforms. Although laws such as Section 508 were only targeted at specific public sector segments in the United States, the desire for large software companies like Microsoft to conform to Section 508 to be able to sell to the government has resulted in accessible software that is available to the general public. Additionally, websites and software created to be accessible in one country where a company does business will typically maintain that accessibility when accessed from a country with little to no regulation regarding accessibility. Therefore, law and regulation—even when applied to a subset of organizations that publish content to the web—has far reaching effects on accessibility overall.

> “I know that there are many government agencies that post videos on YouTube—if YouTube leveraged that and actually created a platform that was fully 508 compliant and supported audio descriptions, that would go over like gangbusters. I know that even more organizations would be willing to post videos and use YouTube as their video solution. As it is now, even though technically, according to Section 508, agencies are not supposed to put their content on an inaccessible site, the fact is that they do. But they are still required to put that video somewhere else where

> it is fully accessible and conformant to 508. But YouTube could—I don't know if they could find a way to monetize that, and I hate to say that, but if it were reasonable, I would say that government agencies would be willing to pay to put their content on a site like YouTube if it were fully accessible."
>
> —Informant K

There is a market for technologies and platforms that are Section 508 compliant that has only been tapped to a limited extent thus far. Typically, enterprise-grade software solutions, such as those provided by companies like Microsoft and Oracle, are Section 508 compliant in order to be sold directly to government agencies. Technologies and platforms like YouTube, which are more consumer-focused, are slower to catch up, since government agencies are not their primary audience. The consequence of which is that average web users tend to gravitate toward familiar platforms, such as YouTube and Facebook, which puts government agencies in the unfortunate position of having to post their content twice—once on the popular, but not fully accessible platform and once on their own less popular, but fully accessible platform.

If a company such as Google (which owns YouTube) could make the platform fully 508-compliant, then government agencies would have one place to put their content that is fully accessible and has the benefit of being a desirable destination for their end-users. Beyond the benefit to government agencies trying

to expand the reach of their content, YouTube users unaffiliated with government agencies could take advantage of the Section 508 accessibility enhancements to make their own content more easily accessible.

7.4 Synergy Between and Among Standards

> "I think that semantic and well-formed code is the basis for developers to make something accessible."
>
> —Informant K

Web accessibility is not a concept that was tacked on to general web development practices—it has been present all along. As such, the general web standards produced by the W3C for HTML and CSS include best practices for accessibility, such as ensuring that image tags include the "alt" (alternate text) attribute, and providing a "longdesc" (long description) attribute for providing long-form information about images. There is a "title" attribute on the anchor (hyperlink) tag which allows developers to include additional information about where the link leads, if it is not immediately obvious from the content of the link, such as when the link surrounds an image.

Above even the level of meta-descriptive attributes on tags, the HTML tags themselves are intended to be semantic descriptors of the content. For instance, the tag to make a piece of content appear bold is , which has the added effect of causing a screen reader to pronounce the content within the tag louder than the surrounding content. The advantage to using a tag in this context, instead of simply using CSS to transform the text to

a bold font, is that the tag itself conveys semantic meaning to assistive technology, allowing the assistive technology to treat the content differently when presenting it to the user. Other such examples include the tag, short for “emphasis,” which renders text in italics and signals a screen reader to emphasize the text in playback; the <aside> tag, indicating content that is supplemental, not central, to the main content; and the <header>, <footer>, and <nav> tags, denoting the header, footer, and navigation sections of a website, respectively.

> “Thus, the benefits of standards are multiple: they help designers work better and more efficiently, help businesses stay on the right side of legislation because they are likely to result in accessible websites, and they make websites more findable because the code is easy for a search engine to read.” (Kennedy 2012, 92)

Advances in web standards, particularly with the HTML5 standard that was finalized on October 28, 2014, have added a significant number of new semantic tags and meta-descriptive attributes to the HTML language specification that can be utilized in more accessible ways by a variety of browsers and devices (W3C 2014b). An excellent example of this phenomenon is the way that new form input controls are presented within mobile web browsers.

Mobile web browsers are often harder to use than their desktop

counterparts due to three factors—the lack of a screen large enough to present a significant amount of information at once, the lack of a physical keyboard, and the lack of a separate physical pointing device, such as a mouse. Digital keyboards are often much more difficult to use than physical keyboards due to the size of the virtual keys when compared to their physical counterparts, and the rather confusing interface for handling non-alphabetic characters, which typically involves cycling through a list of alternate keyboards that include additional glyphs. Likewise, touch-based interfaces can be more difficult to use precisely when compared to using a traditional input device such as a mouse, which is capable of being precise down to the pixel.

Mobile browser makers have embraced the new HTML5 form input types, and deliver customized virtual keypads designed for the type of input specified. For example, if the input type is specified as “email” instead of just “text,” then the virtual keyboard will include the @ and . symbols on the first “page” of the virtual keyboard, which means that the user will not have to cycle through a series of alternate keyboards to find the glyphs they are looking for. Likewise, if an input is designated as a “number,” then a numeric-only virtual keyboard will appear.

Some traditional controls have also received a mobile browser makeover, such as the <select> (dropdown) list, which appears on mobile devices as a scrollable list of choices inside of a dedicated area of the screen, instead of a box that drops down within the page as seen on desktop computers. These advances represent gains in accessibility both for people with impairments—since the

process of form input is simplified, and relevant information is surfaced instead of buried—and also for average users, for whom the virtual keyboard often presents difficulties whether one has an impairment or not.

7.5 Synergy with Disciplinary Norms

When the first iPhone was released in 2007, a major shift began to take place among web designers and developers to start taking the mobile Internet seriously. At that time, most websites were optimized for display on a 1024x768 or larger desktop computer screen, and developers made (often largely valid) assumptions that their users had keyboards and mice. When an iPhone user visited a website that was not optimized for mobile viewing, they would typically see the desktop-optimized site in a zoomed out state, which would often render the website unreadable without zooming in and panning around—an action which was unintuitive for most users. As a consequence, many users tended to abandon websites on the mobile web due to usability challenges, not dissimilar to experiences of Internet users with impairments abandoning sites due to accessibility challenges.

The first stage of remedying this problem involved creating separate mobile sites, optimized for iPhone screens that were 320 pixels wide and 480 pixels high, usually with a limited subset of information and served from a different URL. This approach treated mobile users as second-class citizens, as mobile users were unable to access all of the content on the website. In some very salient ways, this treatment mirrors a real socioeconomic divide, as many

people of color in urban neighborhoods are opting for mobile phones connected to cellular data networks as their only means of online computing, eschewing in-home Internet access and desktop or laptop computers completely (Smith 2015; Smith 2014; Smith 2013). Viewed through this lens, separate mobile-only websites with limited content perpetuated racial discrimination in access to information, goods, and services.

When the responsive design paradigm was introduced by Ethan Marcotte in 2010 (Marcotte 2010), it quickly transformed the routine design and development practices for websites as a whole, making smartphones and tablets first-class citizens. Under the responsive design methodology, content priority and hierarchy is decided “mobile first,” which means determining how the website should appear, and what content should appear in what order, on a mobile device (specifically a smartphone) first.

As viewport size increases, for large smartphones and tablets, the content on the page “responds” to the increase in available browser real estate, and moves elements around on the screen to take advantage of the additional space. Understandably, most content on smartphones is displayed in a single column due to space constraints. For users with larger screens, such as tablets, content will typically switch to a column-based view for certain elements, often with main article text spanning the width of the tablet. For users on laptop and desktop computers, layouts typically change again, perhaps bringing supplemental content up from below an article into a sidebar.

Responsive design also accommodates extra large screens. For example, a 1080p full-HD 16:9 screen is 1920 pixels wide by 1080 pixels high, which is 2.5 times as wide as the available screen real estate on a model year 2015 iPad in portrait orientation. As such, there are opportunities for targeting users on large screens, including large desktop computer monitors and Internet-enabled so-called “smart” televisions, which may include approaches such as expanding and moving main navigation to a column on the left side of the screen rather than keeping it collapsed within the header.

There are two important lessons in the shift to the responsive design paradigm—thinking mobile first, to deliberate over content priority in a much more conscious and intentional way; and abandoning assumptions about who the user is, what their operating system or browser or device of choice is, and what means they have at their disposal to interact with elements of a website. For instance, a mobile user will have a small screen, no physical keyboard, and no external pointing device such as a mouse. Websites that rely on “mouse hover” effects will not work on mobile devices, that operate on the basis of touch, instead of manipulating a mouse cursor using a peripheral device. Web accessibility advocates have long argued that assumptions about who “the user” is, and what their physical and cognitive capabilities are, were never valid, irrespective of what device they were using. The wholesale shift to responsive design methodologies has caused the web design and development world as a whole to grapple with assumptions about how the user will interact with a website

in a much more significant way, even if they are not considering the full range of physical and cognitive limitations that users may be faced with, instead focusing on purely technical limitations of the devices that they are using to access web content.

Responsive design also represents an evolution in ideas about “universal design.” Universal design is defined as “the design of products or environments to be used and experienced by people of different ages and abilities **without adaptation**” (Mustaquim 2015, 66, emphasis mine). One of the core tenets of responsive design is that it responds and adapts based on a variety of factors, including the size of a user's screen, the available pixel density of the screen, and technological capabilities of the client, such as an ability to support cutting-edge techniques that may not be available in all browsers. Therefore, responsive design is able to sidestep one of the major criticisms of universal design: that one artifact should be all things to all people without adaptation. The technological promise of responsive design is that it allows designers and programmers to deliver the best experience for different use case scenarios using the same website and code, utilizing conditions to alter the output.

As a dual consequence of the shift to responsive design methodologies and the overall increase in complexity in the field of website design and development, many designers and developers have shifted to using some sort of framework or codebase from which to begin design and development work that incorporates best practices including responsive design and SEO. In some

cases, designers and developers select frameworks that are publicly available, such as Twitter Bootstrap and Zurb Foundation, and in other cases, they build their own, perhaps using a set of helper tools as part of an automated workflow.

> “It’s all stuff that is becoming part of the standard operating procedure in terms of making those things. It’s almost getting to where we don’t even think about it now, you just do it. It’s becoming part of the standard. I think most agencies, and even most development shops, start to build up their own toolchains. Doing things like that, like having the standard templates and things, which have things in the right order, which is going to help you with that anyway.”
>
> —Informant N

In addition to frameworks and toolchains, designers and developers typically utilize industry best practices for visual display and information layout, which are increasingly aligning with best practices for accessibility. For example, thanks in part to the shift to responsive design, base font sizes and contrast ratios between text and background colors have increased. Depending on the size and resolution of a user's screen, under what conditions they are accessing the website, and how bright they have set their display, fonts have to be larger and text needs greater contrast in order to be readable and understandable.

Likewise, designers and developers tend to set up their base templates to include information and presentation methodologies that translate well to standards compliance, SEO, performance, and—in part—to accessibility.

> “Well, I think number one is kind of, you have the idea of, we’re going to have minimum baselines for font sizes, right? There’s this kind of basic approach to design, where we want to make things readable, right? We also go to great lengths to … well, not great lengths … in terms of that basic level, we select tools that describe our pages well to machines, kind of out of the gate. And then, from that point moving forward, once you kind of get away from the technical aspect of a project and you start to focus more on the content, we become more responsible for adding more, right? To best describing that content within the website.”
>
> —Informant B

Kennedy warns us to not be overly pessimistic when it comes to the state of accessibility on the web. Particularly given the advances in web design and development methodologies, techniques, frameworks, and best practices over the past five years, I am inclined to agree with her. According to Kennedy's findings as well as my own, routine design and development practices have

embraced some key accessibility methodologies, which have been advocated by the accessibility community for years, under the umbrella of best practices design for general usability. In my view, the more that accessibility practices can be viewed as part of best practices design for general usability, instead of targeted only at people with impairments, the better uptake there will be within design and development communities, as the investment in making the website more usable will be viewed as affecting all users, instead of just the subset that has an impairment.

7.6 Synergy with Open Source Communities

If I have seen further, it is by standing on the shoulders of giants.

—Sir Isaac Newton

Open source projects are widely used in producing websites. For example, the WordPress open source content management system is estimated to power approximately one in four websites (WordPress 2015). The open source Twitter Bootstrap front-end framework is estimated to power millions of websites (Twitter Bootstrap 2015), and can be used in conjunction with content management systems like WordPress. Therefore, influencing the accessibility of open source software projects that power significant number of websites can go a long way toward making the web accessible, particularly for smaller web presences that lack budgets for user testing and accessibility testing.

“I don’t know if Twitter does user testing, or

> WordPress does user testing. I sort of assume they do, because of how big those projects are, and figure that because there's these huge communities that develop for these platforms, a lot of that stuff is sort of vetted. But I guess it sounds like you're asking, even though we don't do user testing, do I feel comfortable thinking that some of that's—that testing has been offloaded to somebody else? And I would assume yes..."
>
> —Informant A

Many open source projects have groups of accessibility advocates that are dedicated to making the project accessible. WordPress has one such group, called Make WordPress Accessible, which is responsible for evaluating the accessibility of WordPress core releases and periodic official theme releases. Core releases happen approximately every four months, and theme releases happen once per year. Theme releases are named for the year in which they are released—for example, Twenty Fifteen, Twenty Sixteen, and so on.

The Make WordPress Accessible evaluations of core releases are primarily related to the administrative, or content editing, functions of WordPress—the parts of WordPress that make it a content management system. Some components of core affect the view that end users see, such as the login and forgot password screens, and certain built-in sidebar widgets and menus.

However, most of the accessibility features that affect end users are present in the theme, so the team's evaluation of the yearly themes are primarily focused on the experiences of general website visitors instead of content editors. Although WordPress allows site administrators to change the theme away from the default yearly release theme that is evaluated by the Make WordPress Accessible group, the default themes are in use by millions of users, and accessibility-related modifications to these themes have far-reaching effect. Additionally, these default themes are often used as a starting point for custom theme development, so accessibility features included in these themes may make their way into custom themes as well.

The Make WordPress Accessible group, and other such open source accessibility advocacy subgroups, are part of what Chris Kelty calls a “recursive public”—a public that is constantly maintaining the conditions of its own existence (Kelty 2008). Make WordPress Accessible exists as a subgroup within the broader WordPress open source community because the depth of knowledge required to ensure full accessibility of both the core offering and the yearly themes is esoteric enough that average contributors cannot master it. On a project of that scale, it is valuable to have a group that specializes in accessibility review, because they are able to identify accessibility issues that core developers may miss. The members of Make WordPress Accessible report issues back to core contributors, which are often fixed in advance of a major release.

The role of Make WordPress Accessible is disputed within the open

source WordPress community. Since there is a group dedicated to accessibility, some developers view accessibility as the responsibility of the accessibility group instead of as a collective responsibility. This conflict was brought to a head during the release of WordPress 3.8 in 2013, when a major keyboard accessibility flaw was identified late in the release process, without sufficient time to fix the issue prior to the release of the new version of the software. Members of the all-volunteer Make WordPress Accessible team voiced frustration that core developers were not making their updates in such a way as to be accessible from the beginning, and instead relying on the accessibility team to identify accessibility challenges after the fact (Mullenweg 2013).

Irrespective of the specific politics surrounding the role of Make WordPress Accessible and other accessibility advocacy groups within open source projects, the fact that they exist and are able to positively influence the level of accessibility of the end product is of critical importance for making the web more accessible overall. As with any accessibility advocate within an organization, the presence of these groups, and the work that they do, elevates the importance of accessibility within the open source projects that they support, and makes it more likely that accessibility will be included from the beginning instead of being an afterthought.

7.7 Conclusion

The evolution of the web design and development landscape over the past twenty years represents a tension between the social value placed on making

technological advances and the evolving concepts of what constitutes appropriate design for usability. The way that the web design and development community has responded to accommodating changes in technology versus inclusion of accessibility components is telling. Based on my research, web designers and developers are quick to adopt new technologies and cutting-edge methodologies, whereas accessibility components have typically only been included when inclusion of accessibility was mandated by law or regulation, or where there have been overlaps between accessible methodology and other concerns, such as adherence to standards, optimizing websites for search, or where a particular aspect of accessibility has been a part of general web design and development best practices. Therefore, I argue that it is important to focus on the synergies between accessibility and other concerns that are typically more important to clients and managers, such as time to design and develop, adherence to budget, SEO scores, and general usability and user satisfaction, in order to promote better accessibility for people with impairments within organizations, frameworks, platforms, and software generally.

Understood in this way, accessibility for people with impairments can be a component of a larger trend toward overall usability of websites under a variety of circumstances, not all of which need to involve or include impairment. By giving users flexibility to choose how they interact with website content—on which device, with which operating system, with which browser, at which screen size, with which interface elements—the task of making a website accessible for

people with impairments becomes largely assimilated into the task of making a website accessible for a user in control of their own preferred method of access, which cannot be assumed by the designer or developer.

The shift to responsive design methodologies was a part of this overall ideological change, in which designers and developers had to abandon many of their closely-held assumptions about who their users were and how they interacted with the websites they were producing. In particular, the shift to a touchscreen device highlighted why assumptions about the way users would interact with an interface were problematic and needed to be abandoned.

This trend has continued with the rise of social media, and the increasing consumption of social media on mobile platforms, such as the Facebook smartphone app. As social media platforms increasingly accept diverse forms of multimedia, design of multimedia assets has changed to reflect the different circumstances under which multimedia content is consumed. For example, assumptions about consumption of video content—that the user would want to consume the content in the form of a video, that the user would be able to hear the audio in the video and see the picture—are increasingly invalid when the video is displayed on a social media newsfeed such as Facebook.

Although Facebook automatically plays videos present in a user's news feed, the audio track of the video is muted by default—thereby making the default listening experience for the video the same as if the user were deaf. Although hearing users can interact with the video to enable sound, social media is often

browsed under circumstances that are not favorable to playing audio—such as on a train, in line at a coffee shop, in a waiting room, or at the workplace. Therefore, more accessible and usable alternatives are needed to accommodate general use cases that involve watching a video without sound, which includes both users that are deaf and those that prefer to watch the video without sound. As of the time of writing, a number of news organizations that post videos for consumption on social media have begun adding captions to their videos for precisely this reason (Flynn 2015).

Some users prefer to consume the content of a video or audio recording in text format, even if they are able to hear the audio track, for various reasons. One such reason is the amount of time it takes to listen to audio compared to the time it takes to read a transcript. Users wanting to quickly scan or consume the content presented in a video or audio track will prefer a transcript, such as the transcripts that are produced for debates during the U.S. election cycle. Transcripts work well for deaf users, users that prefer to consume content by reading instead of listening, users that have difficulty maintaining constant attention and would prefer to read at their own pace, and users with limited bandwidth for which streaming a video is impossible or impractical. Additionally, transcripts yield significant benefits for SEO, since the entire content of the video can be indexed by a search engine, making the content easier to find for a user performing a search on keywords present in the transcript but not in the video title or description.

A focus on accessibility complements user choice in how they consume the content. Likewise, giving users options of how they would prefer to consume content can result in greater general usability of a website, even for users that are non-impaired, while maintaining benefits for users with impairments. Thus, there is synergy between accessibility and a number of other factors that are components in what is emerging as best practices web design and development, including search engine optimization, usage of open source frameworks and platforms, responsive web design, law and regulation in specific sectors, and adherence to markup language standards.

Synergistic enablement is decidedly utilitarian in its approach. Although it would be ideal for all web designers, web developers, and content creators internalized the importance of making websites accessible, I doubt that that it matters to the deaf user that is able to read a video transcript or the blind user that can interact with all elements of a website whether the motivations of the designer, developer, or content creator were primarily those of accessibility versus some other concern, such as optimizing for search or choosing a platform that was easy to use. Similarly, Pinch and Bijker describe how the transition to air-filled tires on bicycles happened once the public saw the benefits of using air tires for going fast and winning races, even though they were originally developed as a means to reduce vibration (Pinch and Bijker 1984, 427–8).

In this way, synergistic enablement is most powerful as a tool in the accessibility advocate's toolbox in order to strategically advocate for better

accessibility. Instead of making the argument that a business or organization should invest in accessibility for moral or ethical reasons, they can advocate that the business or organization invest in best practices web design and development, which includes accessibility not only for people with impairments, but also for people with slow network connections, small screens, search engines, and users that prefer to consume content differently.

Likewise, I also propose discussing accessibility in education and policy as an integral component of best practices web design and development. By presenting accessibility as something separate and distinct for the primary purpose of improving the experience of users with impairments, it becomes an easy target for elimination due to time or budget constraints, because it is so vulnerable to the cost-benefit argument, particularly for small organizations and businesses. Instead, we should be discussing accessibility in broader terms, which includes accessibility for people with impairments as well as accessibility to software programs like search engine spiders, elements of user choice such as choosing to read a video transcript instead of listening to the audio, and socioeconomic accessibility, such as supporting users that cannot afford a PC and a home Internet connection, but can afford a smartphone.

8. IMPROVING WEB ACCESSIBILITY

Despite improvements in web accessibility due to the shift to a responsive design paradigm, an uptick in utilization of increasingly accessible frameworks and platforms such as Twitter Bootstrap and the WordPress content management system, the state of web accessibility in 2016 still appears to be relatively poor. So what is to be done? Should we pass laws mandating web accessibility? Should we mandate accessibility education, perhaps as early as high school? Should we impose penalties on sites that fail to meet accessibility guidelines, or implement some sort of approval mechanism for web publishing requiring adherence to accessibility standards as a condition for publication?

> “[T]here is not much evidence that self-regulation has been a problem for the standards movement, and some evidence that regulation has proven problematic in other areas of web design.” (Kennedy 2012, 102)

I am inclined to agree with Kennedy, in that I view top-down regulation of web workers as being problematic, in that it does not inspire web designers and developers to actually care about making content accessible, and instead focuses on adhering to a minimum required legal standard to avoid punishment. Likewise, top-down regulation of web workers would likely be counterproductive in the stated aim of making the web more accessible for people with impairments for the same reason. Rather, I would like to encourage and foster current pro-accessibility trends taking place within web design and development circles,

believing that these pro-accessibility trends will ultimately have a larger impact on ableism in web development groups than oppressive top-down regulation. Solutions to improve web accessibility involve interventions at a variety of levels to address the complex causes and manifestations of ableism within web development groups, which involve strategic government and educational intervention combined with voluntary action on the part of framework and platform creators and maintainers.

8.1 Policy Interventions

Although I believe that top-down regulation on web developers will have a detrimental effect if applied too broadly, there are reasons to believe that strategic and limited policy interventions can have a significant positive impact on web accessibility. One example of this that is already in effect is Section 508, which requires federal agencies and organizations receiving federal funding to make accessible websites.

Apart from some notable and highly publicized failures, Section 508 has largely been effective in elevating the accessibility level of federal government websites in the United States, as evidenced by Bigham et al. in their 2006 study of image labeling in government and non-government websites (Bigham et al. 2006, 184). Section 508 requirements have also been useful in increasing the overall accessibility of software by exerting market forces on technology companies producing software for government and non-government use. The companies making the software have to comply with Section 508 in order to sell

to the government, but non-government users of their software benefit from the accessibility features initially coded for compliance to government standards.

The Americans with Disabilities Act is somewhat of a different story. The ADA was enacted before graphical web browsers made the Internet popular outside of academic and government circles, so the law does not contain explicit mention of the Internet or websites. Title III of the ADA covers “places of public accommodation,” which has traditionally been understood to be brick-and-mortar establishments, and not the Internet. The Justice Department issued an Advance Notice of Proposed Rulemaking in 2010 (U.S. Department of Justice 2010) covering its intention to clarify Title III of the ADA to explicitly include websites in its definition of “places of public accommodation,” similar to the way laws like the UK's Disability Discrimination Act are structured.

However, the Notice of Proposed Rulemaking, the next step in the process, has been repeatedly delayed. As of writing, there is no date set for the Notice of Proposed Rulemaking or the comment period that follows. Despite the delay, the Justice Department has been vocal in recent court cases regarding web accessibility with respect to the idea that the ADA has always applied to websites, even if the specific language was not present in the law.

> “The Department is currently developing regulations specifically addressing the accessibility of goods and services offered via the web by entities covered by the ADA. The fact that the regulatory process is not yet

> complete in no way indicates that web services are not already covered by title III." (U.S. Department of Justice 2012, p. 12)

The lack of clear and explicit guidance from the Justice Department makes this statement ring hollow. If the Justice Department was serious about applying the ADA to places of public accommodation—and moreover, if it is maintaining the position that web accessibility was part of the ADA all along—then it should waste no time in issuing the Notice of Proposed Rulemaking, and make the interpretation of the law public and explicit. Instead, the Justice Department is making its guidance known through documents such as Statements of Interest in court proceedings for specific cases, such as the one quoted above in the case of the National Association of the Deaf et al vs. Netflix, Inc. Most lawsuits against private companies for not providing an accessible experience on their website are settled out of court, and the lawsuits that do go to court have resulted in rulings that do not set precedent. However, literature on digital accessibility is embracing legal risk as a motivator for action, and is highlighting organizations that are able to integrate accessibility successfully into their routine development practices (Lazar, Goldstein, and Taylor 2015).

If the Justice Department moves forward with making the ruling on the applicability of Title III of the ADA to websites as places of public accommodation, pursuant to the current definition of places of public accommodation as pertains to brick and mortar establishments, and inclusive of exemptions for small

businesses, I predict that there will be a similar shift toward producing accessible software platforms for website development as there was with Section 508. If companies like Squarespace, Facebook, and YouTube want to sell their services to businesses in the United States above a certain size, they will need to demonstrate that their software meets the legal requirements. In this way, legal intervention will provide market-based financial motivation for increased accessibility at the level of platforms and frameworks, which will provide benefits to all designers, developers, and content editors that wish to use those platforms, not just the businesses that are legally required to provide accessible sites.

> “I think that people in my position want to sell the projects, and do the projects. The user doesn’t really care so much. Imagine if you’re a gas station. The person just cares that the gas gets in the car, the station just cares if they make money. It’s the government’s responsibility to make sure that there’s not gasoline seeping into the ground, that the pumps are calibrated, or whatever. Somebody uninvolved, I guess you could say.”
>
> —Informant R

According to this informant, the government, or another third party, should fill the role of the regulator, ensuring accessibility at a higher level than the individual developer. One such way this can be accomplished is by requiring

accessibility of larger private businesses, which will create a market for accessible frameworks and platforms that can be used by smaller businesses and organizations as well.

Another type of intervention could happen at the marketplace level, where pre-built themes, plugins, and the like are sold, with marketplace owners filling the role of the policymaker or regulator. WordPress.org, the homepage for the open source WordPress content management system, has something like this in place already. Users are able to search for free, open source themes in the WordPress.org themes registry using a variety of criteria, one of which is accessibility. A legal requirement that places of public accommodation online be accessible to people with impairments will necessitate similar filters on other marketplace sites to make it easier to identify which themes and plugins are built with accessibility in mind.

I envision that such a system could be expanded to use a form of gamification, where theme developers could earn “badges” for conformance to certain best practices, including accessibility, standards compliance, performance, responsive design, compatibility with a variety of web browsers, and so forth. This structure would create incentives for developers to create themes that meet certain criteria to better market their theme to potential buyers, who are attempting to evaluate many choices, and need a straightforward method of discriminating between competing options.

No such system would be complete if it only relied on self-reporting in

order to obtain the relevant certifications. There would need to be a customer feedback option that would allow users to report problems with accessibility, performance, display, and so forth, in order to keep theme developers honest. In this way, ongoing evaluation of themes for accessibility could be a community-supported endeavor.

Intervention at the level of frameworks and platforms, spurred by clarifications to existing law, is important, but is only a necessary first step. Without proper training, it is far too easy for content creators to add inaccessible content into an otherwise accessible framework or platform, thus perpetuating the problems of ableism in the production of websites. Since the overall accessibility of a website will suffer if a focus on accessibility is not included at every step, it is important to educate all members of project teams of the importance and need for accessibility, and how to integrate accessibility into every web production task. The most accessible framework in the world, produced by the most conscientious designers and developers, will fail if content creators are not also trained in how to make content accessible within that system.

8.2 Education Interventions

At the core of most web accessibility problems—from design to development to content creation—is a lack of awareness of the challenges that people with impairments face online, which I believe to be the primary factor behind ableism in web design and development. Most web workers are not

aware of what these challenges are, or that they are more complex or varied than simply needing to put alt text on an image, or use text in HTML instead of embedding text in images. This lack of awareness is primarily a failure of education. Based on my interviews, the more aware project team members were of the need to include accessibility features in the websites they made, the more likely they were to include them. This phenomenon was especially true when an accessibility advocate was present on a project team. Therefore, it is critically important that web designers and developers are educated about the need for accessibility as a necessary first step to reducing the effect of ableism on website projects.

Web development project teams are made up of a variety of individuals from a variety of educational backgrounds, working on a range of different tasks to support the website build and content population. As evidenced by my survey data, many web designers and developers did not finish college. Some members of project teams may not have been trained specifically in web production, such as writers that may have a creative writing background, or project managers with an MBA, or designers that were trained primarily in print design.

Therefore, accessibility interventions in education should not be restricted to web development in a post-secondary environment. Rather, accessibility challenges for content in general, including on the web, in software, and in physical media such as print, should be addressed across a range of disciplines, and introduced as early as high school. Planting the seed of awareness will make

later conversations about the specifics of implementing accessibility in particular contexts easier, as the work of demonstrating the need for accessibility will have already been accomplished.

> “[A] lot of the conversations had to do with better understanding the capabilities of people who were being content creators, and trying to figure out how to raise their awareness of how to properly create content, for example, that would be more easily read by a screen reader.”
>
> —Informant B

Although the sample size was rather low, it appears from my survey data that trade schools do worse than liberal arts schools, community colleges, and technology-focused colleges at teaching accessibility. This finding is somewhat counterintuitive, as trade schools should theoretically be teaching the specifics of how to develop for the web, which should include technical matters such as adherence to standards, which would include accessibility standards, particularly given the current legal climate surrounding accessibility. Perhaps trade school curriculum is focused intently only on getting the site to display and function properly for the normate user, and does not delve into standards compliance and accessibility. More study in this area is needed, which can yield more concrete recommendations for intervention at the level of trade schools.

> “I don’t know if trade schools is the right word, but

> really specified schooling will be more prone to address this, because they kind of have to be on the forefront when you're in a specified area. They have to be pushing the envelope and bettering things on all levels to compete with other schools."
>
> —Informant D

A primer on the basics of accessibility for all members of project teams, particularly those dealing directly with clients, is essential to being able to communicate the importance of accessibility to clients. Especially as law changes, and the ADA is clarified to apply to websites as well as brick and mortar establishments, it is critical to explain the risks of not adhering to accessibility best practices to clients to help prevent them from making decisions that would impact site usability and cause them to run afoul of the law. Additionally, as I argued in Chapter 7, accessibility does not stand on its own as a concern unto itself—it is integrated into web design and development best practices for overall usability by all users, not just users with impairments.

> "Educating clients is valuable as well, as site owners there is a certain degree of responsibility to knowing the high level points of accessibility and what it means to people visiting your site, so that responsibility lies on the service provider in sharing that knowledge with their clients."

—Informant T

If web designers and developers are not taught, as part of their professional education, that accessibility is an integral part of website design and development, then it will get relegated to the position of being an optional extra, to be integrated at a later date if time and budget permit. By pushing off accessibility to after the primary design and build is done, it becomes significantly more costly to incorporate accessible practices than if it was considered and included from the start.

Hommels refers to this phenomenon, when it manifests itself in physical space, as “obduracy” (Hommels 2005, 324). For Hommels, obduracy is introduced in the design of the urban environment. Buildings, once built, are difficult to modify, which makes modification during the planning phase a much more cost-effective option. In web design and development, the process of building up bad or inefficient or inaccessible code is called “technical debt,” which will continue to charge “interest” in the form of bad performance or an inaccessible interface until it is repaid. Technical debt is often accrued because timelines and budgets do not allow for the code to be written “the right way” the first time, and thus a form of “loan” is taken out against future development efforts by cutting corners in the moment. Therefore, in order to avoid accruing technical debt—obduracy—accessibility should be taught as part of core usability for web design and development through all levels of the curriculum.

“It’s so much faster just to build [a website] from

> square one with [accessible] code included. So, “the right way” to me is using attributes that we know will support accessibility. Making sure that the design itself won’t produce an inaccessible experience. Making sure that there are buttons there to support people who are not using a mouse.”
>
> —Informant K

It is important to emphasize in education that the goal for all content is to be as accessible as possible, irrespective of the medium, or the anticipated audience, because the anticipated audience is likely broader and more diverse than the content creator believes. Additionally, content created for one potential audience can easily be shared or expanded to others, so there are benefits in making content accessible from the beginning, rather than trying to retrofit it later.

There exists something akin to the Internet version of the American Dream, in which anyone can become famous if their post “goes viral.” Through widespread use of social media, a funny or thoughtful post can be shared and re-shared and reach a large number of people quickly, which can be a powerful (although decidedly self-interested) argument in favor of ensuring that content is made to be accessible, even if the expected audience is thought not to consist of people with impairments. If individual designers, developers, or content creators can establish a personal connection between what they are producing and the diversity of the potential audience, it can make the argument in favor of making

accessible content more compelling, and help to chip away at some of the assumptions that underpin ableism.

> “Say something like social media, okay, where people can take your blog post, and then send it to all of their friends and associates. That’s huge. That actually gets people listening, I think, and it makes them understand that, “well yes, if [making my website accessible] is going to get me more hits, if this is going to get me more exposure, then sure, I want to do this.”
>
> —Informant K

Finally, it is critically important to increase the diversity of teams that are making websites. Web design and development is a field that is overwhelmingly white and male. Teams tend not to include people with impairments. As I argued in this dissertation, the presence of an accessibility advocate can have a profound effect on an organization. By encouraging people with diverse backgrounds, including people with impairments, to pursue careers in web design and development, and by finding ways to keep them in these careers, we can improve web accessibility and usability by designing and developing websites utilizing a broader range of experiences and viewpoints. Unsurprisingly, websites that are designed and developed by upper middle class white men tend to work best for upper middle class white men (Oudshoorn, Rommes, and

Steinstra 2004). I believe that these interventions should begin in the educational process.

8.3 Community Interventions

It is the responsibility of all members of web development teams, including content creators, to ensure that the web is an accessible place for everyone. The web design and development community has a role to play in advocating for accessibility as part of professional practice, as part of doing good work. To a large degree, accessibility is included in what it means to do good work, but not as much as it could be, or should be. Partly, this is due to ignorance of what it means to make accessible sites. It is also often due to an inability to convince clients or managers that accessibility is important, and is worth the investment. I also believe that it is due to framing accessibility as being something unto itself, instead of inextricably intertwined in all of the practices that make up web production. The accessibility advocate can play an integral role in bringing teams around to this concept, particularly given the lack of general accessibility education and awareness among web development groups.

> “It’s really easy to take for granted your ability to identify colors in a conventional way. And it’s, I think, really excellent that we have people who are able to be that kind of advocate, or are able to say light blue and light gray look really nice, but it’s going to be a real challenge for some people. And the best thing to

> come out of that is, in fact, these considerations being added to things like a commit checklist, right? And that, I think, is progress. Because it is really easy to forget, and it is really easy to lose sight of the fact that the way that you operate and interact with things isn't the only way that people interact with things."
>
> —Informant E

Accessibility advocates can take many forms. It might be an individual on a project team, or a member of Make WordPress Accessible, or a contributor to another open source project that reminds people to include accessibility, or even an online tutorial creator that just takes the extra time to include accessible code in their examples. The effect of an accessibility advocate is far-reaching—a little work goes a long way. Education can help with this, as can better inclusion of accessibility in professional development activities such as conferences. Ultimately, accessibility needs to be reframed as an integral component of general usability, and needs to be incorporated into routine work practices in much the same way as responsive design.

> "So hopefully, like responsive design, as we think about [accessibility] more and more, and as it becomes more and more prominent in what we're reading, or what we're thinking about, even like performance issues on mobile networks, and things

like that, I think that it will become more of a thing."

—Informant H

One of the ways that accessibility can become codified into general design and development practice, similar to responsive design, is by incorporating accessibility best practices into frameworks and platforms that designers and developers are already using. Many frameworks have already embraced accessibility. By continuing this trend, there are amplified effects for developers that adopt these frameworks.

> "I use the Genesis framework, which is some of the most accessible code I've ever seen, and they're continuing to improve on that. The next release, coming out in a couple of weeks, is focused exclusively on accessibility, where they are just making sure that the entire codebase is accessible, soup to nuts."
>
> —Informant I

Ensuring that platforms and frameworks are accessible is great for developers, but doesn't address the content creation layer. Most of the content that will be published to the web is produced by non-experts. It is unreasonable to expect that these non-experts will be able to figure out how to make their content fully accessible. It is possible to modify authoring tools and platform interfaces to help guide content creators toward adding metadata for

accessibility, and trying to convince them to take the steps to appropriately tag and describe content, but there will be some portion of user generated content that just isn't accessible. Building on the idea of transcoding introduced in chapter 2, community resources can be tapped either for free or for pay to make content more accessible.

> “I think there are a few ways that we can do that. One of them would be to tap into the crowdsourcing sites for providing captions to video. That could be a big wave to ride. There are some people who will sit there and caption a five minute video for someone else, because it’s going to have a big impact. So why not, if people are willing to do that?”
>
> —Informant K

There is a market that currently exists for accessibility testing, transcribing, captioning, and the like utilizing dedicated services such as Knowbility and general purpose services such as Amazon Mechanical Turk. An additional benefit of these services is that they can provide opportunities for people with impairments who otherwise may have difficulty finding work to be able to participate in accessibility testing, often from the comfort of their own homes. Accessibility and usability testing with people with impairments is the only true measure of accessibility, even for sites that have followed the accessibility standards to the letter. By performing this testing in an ad hoc manner, smaller

organizations can take advantage of targeted testing only when they need it, and only at a level that they can afford.

8.4 Future Technological Possibilities

There are some developments that are in their infancy at the time of writing that could have a profound impact on transcoding methodologies and user choice to empower users to consume content in ways that work for their abilities and preferences. These new techniques point to the possibility of greater inclusion of accessible methodologies in future web design and development. There are some worrying trends, such as the rise of JavaScript frameworks that present challenges to accessibility, particularly with assistive technologies such as screen readers. There will always be challenges with user generated content, but better authoring tools and better general education for content creators can help to mitigate some of these challenges.

Ultimately, I foresee many of the accessibility challenges present today being mitigated by a shift away from the traditional conception of the website. We are moving into an era of computing more defined by apps than by websites, more defined by content than by presentation. Apps (which is a term that has come to mean applications for smartphones as well as the traditional conception of the PC application) are tightly integrated with their operating systems, and are able to communicate information to the operating system about the content in ways that are mediated by the app code. Platforms like JAWS for Windows could be replaced with built-in solutions similar to VoiceOver for iOS and OS X, which

is produced by Apple and bundled with the operating system, instead of being a separate (and expensive) piece of software.

By integrating assistive technology at the level of the operating system, and by providing mechanisms for desktop and mobile applications to communicate information to the operating system to be utilized by assistive technologies in different ways, a true separation of concerns can be realized between the application layer (responsible for fetching and preparing content) and the operating system layer (responsible for communicating the information back to the user utilizing their preferred method of delivery, including assistive technologies). To some extent, this is happening already, and I foresee the trend continuing.

The mobile web has undergone several major shifts in implementation. As of the time of writing, we are witnessing the beginning of what I believe to be yet another reimagining of what the mobile web is and how it functions. As the mobile web has become slow due to reliance on showing advertisements to generate revenue, and over-reliance on bloated frameworks and third party utilities, the experience of browsing the web on mobile networks has become unpleasant. In response, Facebook has launched the proprietary Instant Articles feature, and Google has launched the open source Accelerated Mobile Pages (AMP) initiative. Both technologies are an effort to speed up the mobile web by providing an alternate version of website content that drastically reduces the amount of time it takes to download and render the page content by utilizing

simple layouts utilizing a minimum of JavaScript.

Both Facebook Instant Articles and AMP are primarily intended for shared article or blog post content, so that when a user loads a shared article in an app such as Facebook or Google Hangouts, the content loads quickly and smoothly. I believe this represents a first step toward decoupling the information in an article from the design of a website, since both Facebook Instant Articles and AMP render with minimal stylistic inheritance from the parent website. Elements such as the global header and footer bar are typically removed entirely, or at least drastically scaled back.

Following this general ideology, if websites are able to serve up information in a way that is at least somewhat disconnected from their primary design, it represents somewhat of a return to first principles in website design and development. Facebook Instant Articles and AMP more closely resemble content from the very early web than most modern websites—where content was fluid, contained a limited number of images, and was presented simply. By stripping out most of the JavaScript and complicated styles, the receiving application is able to present a simple rendition of an article in a way that is theoretically more accessible than the version that is fully baked into the website shell.

Another example of this pattern, although implemented differently, is an app like Apple News, which aggregates and presents content from a variety of sites in one location, through an app that is integrated with assistive technology

such as VoiceOver. These technological changes provide greater potential for integration with methodologies such as transcoding, which allow for user agents on the receiving end to present semantically tagged content in different ways, depending on the needs of the user. Instead of being packaged with a website that has strict opinions about display and function, many of which need to be modified or overcome in order to function properly with assistive technology, an intelligent user agent can use trimmed down semantic content as an input source and tailor the output based on the user's particular needs. In the end, the accessibility challenges of the modern web may be more easily solved by hearkening back to a fundamental principle of computer science—separating content from presentation.

8.5 Conclusion

Solutions to improve web accessibility are not simple or straightforward. They are multifaceted, and involve interventions at a variety of levels. If interventions at any particular level encroach too significantly on designer and developer autonomy, they risk backfiring, since web designers and developers place a high value on their individual autonomy to create the kind of website that they envision. If, by contrast, web designers and developers feel that accessibility is part of general usability, and promoting general usability constitutes doing good work and making good websites, they will be more likely to embrace and advance accessibility than if it was forced on them by law or regulation.

In this dissertation, I have attempted to demonstrate that the process of

designing and developing websites for the modern web is very complex. As a consequence, designers and developers have become increasingly reliant on code that they did not write. I do not believe it will be possible for most of the web to be accessible without addressing accessibility challenges at the platform and framework level, which will require a combination of policy, educational, and community interventions.

It is my hope that by expanding conceptions of what it means to do good work to include accessible practices alongside responsive web design, search engine optimization, and providing an engaging experience for a website's users, that the web can become a more accessible place for everyone.

9. REFERENCES

ADA.gov. 2015. “DOJ ADA Responsibilities: ADA Certification of State and Local Accessibility Requirements.” Accessed October 4. http://www.ada.gov/certcode.htm.

Americans with Disabilities Act of 1990, 42 U.S.C. § 12101 (1990). http://www.ada.gov/pubs/adastatute08.pdf.

Barnes, Elizabeth. 2014. “Valuing Disability, Causing Disability.” *Ethics* 125(1): 88–113. doi:10.1086/677021.

Beck, Kent, et al. 2001. “Manifesto for Agile Software Development.” Accessed October 26, 2015. http://www.agilemanifesto.org/.

Bigham, Jeffrey P., Anna C. Cavender, Jeremy T. Brudvik, Jacob O. Wobbrock, and Richard E. Ladner. 2007. “WebinSitu: A Comparative Analysis of Blind and Sighted Browsing Behavior.” In *ASSETS ’07: Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, 51–58. New York, NY: ACM.

Bigham, Jeffrey P., Ryan S. Kaminsky, Richard E. Ladner, Oscar M. Danielsson, and Gordon L. Hempton. 2006. “WebInSight: Making Web Images Accessible.” In *ASSETS ’06: Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, 181–88. New York, NY: ACM.

Boellstorff, Tom. 2008. *Coming of Age in Second Life: An Anthropologist Explores the Virtually Human*. Princeton, NJ: Princeton University Press.

Brickley, Dan. 2011. “Yandex Now Supports Schema.org Markup.” *Schema Blog*. November 4. Accessed October 4, 2015. http://blog.schema.org/2011/11/yandex-now-supports-schemaorg-markup.html.

Campbell, Fiona Kumari. 2009. *Contours of Ableism: The Production of Disability and Abledness*. New York, NY: Palgrave Macmillan.

Canadian Human Rights Act, R.S.C., ch. H-6 (1985).

Castilla, Emilio J., and Stephen Benard. 2010. “The Paradox of Meritocracy in Organizations.” *Administrative Science Quarterly* 55(4): 543–76. doi:10.2189/asqu.2010.55.4.543.

Chisholm, Wendy, Gregg Vanderheiden, and Ian Jacobs. 1999. “Web Content Accessibility Guidelines 1.0.” *World Wide Web Consortium*. May 5. Accessed October 8, 2015. http://www.w3.org/TR/WCAG10/.

Clark, Joe. 2002. *Building Accessible Websites*. Indianapolis, IN: New Riders Press.

———. 2006. “To Hell with WCAG 2.” *A List Apart*. May 23. Accessed September 16, 2015. http://alistapart.com/article/tohellwithwcag2.

Coeckelbergh, Mark. 2006. “Regulation or Responsibility? Autonomy, Moral Imagination, and Engineering.” *Science, Technology, and Human Values* 31(3): 237–60.

Cooper, Michael. 2014. “WAI-ARIA Expands Web Accessibility.” *W3C Blog*. March 20. Accessed October 4, 2015.

http://www.w3.org/blog/2014/03/wai-aria-expands-web-accessibility/.
Danielsen, Chris. 2008. “National Federation of the Blind and Target Agree to Class Action Settlement.” National Federation of the Blind. Accessed April 14, 2011. http://www.dralegal.org/downloads/cases/target/Final_Exhibit_F_press release.txt.
Dardailler, Daniel. 2009. “WAI History.” *Web Accessibility Initiative*. June. Accessed October 8, 2015. http://www.w3.org/WAI/history.
Disability Discrimination Act, 1992, no. 169 (Aus.).
———, 1995, c. 50 (Eng.).
Eglash, Ron. 2011. “Multiple Objectivity: An Anti−relativist Approach to Situated Knowledge.” Edited by Ranulph Glanville. *Kybernetes* 40(7/8): 995–1003. doi:10.1108/03684921111160223.
Federal Communications Commission. 2011. “Comments of the Consumer Electronics Association: In the Matter of Implementation of Sections 716 and 717 of the Communications Act of 1934, as Enacted by the Twenty-First Century Communications and Video Accessibility Act of 2010.” Federal Communications Commission. Accessed October 27, 2011. http://content.aristotle.com/CEA/CEAACSNPRMComments.pdf.
Feenberg, Andrew. 1999. *Questioning Technology*. New York, NY: Routledge.
Fernandes, António R., Alexandre Carvalho, J. João Almeida, and Alberto Simões. 2006. “Transcoding for Web Accessibility for the Blind: Semantics from Structure.” In *Proceedings ELPUB2006 Conference on Electronic Publishing*, 123–33. Bansko, Bulgaria: IMI-BAS.
Flynn, Nicole. 2015. “Why You Need Closed Captions and Subtitles for Facebook Video.” *cielo24*. September 2. Accessed November 29. https://cielo24.com/2015/09/closed-captions-and-subtitles-for-facebook-video/.
Fodness, Kevin. 2011. “Web Developer Survey Results.” Accessed October 27, 2011. http://kevinfodness.com/web-developer-survey-results/.
Fortun, Kim. 2001. *Advocacy After Bhopal: Environmentalism, Disaster, New Global Orders.* Chicago, IL: University Of Chicago Press.
———. 2004. “Environmental Information Systems as Appropriate Technology.” *Design Issues* 20(3): 54–65.
Garland-Thomson, Rosemarie. 1997. *Extraordinary Bodies: Figuring Physical Disability in American Culture and Literature*. New York, NY: Columbia University Press.
Gavrilas, Razvan. 2011. “Black Hat vs White Hat SEO - Infographic.” *SEO Blog | cognitiveSEO Blog on SEO Tactics & Strategies*. August 8. Accessed March 15, 2015. http://cognitiveseo.com/blog/229/black-hat-vs-white-hat-seo-infographic/.
Google. 2015. “Google Schemas.” *Google Developers*. Accessed October 4. https://developers.google.com/schemas/.
Grimes, Seth. 2009. “Relaunched Recovery.gov Fails Accessibility Standards.”

Information Week. September 30. Accessed April 14, 2011. http://www.informationweek.com/news/software/bi/228901107.

Groves, Karl. 2015. “To Hell With Compliance.” *Karl Groves*. January 6. Accessed October 8. http://www.karlgroves.com/2015/01/06/to-hell-with-compliance/.

Harper, Simon, Sean Bechhofer, and Darren Lunn. 2006. “SADIe: Transcoding Based on CSS.” In *ASSETS ’06 Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, 259–60. New York, NY: ACM.

Harper, Simon, and Yeliz Yesilada, eds. 2008. *Web Accessibility: A Foundation for Research*. Human-Computer Interaction Series. London, UK: Springer.

Henry, Shawn Lawton. 2009. “How WCAG 2.0 Differs from WCAG 1.0.” *Web Accessibility Initiative*. January 15. Accessed October 8, 2015. http://www.w3.org/WAI/WCAG20/from10/diff.php.

Henry, Shawn Lawton, and Judy Brewer. 2005. “WAI Mission and Organization.” *Web Accessibility Initiative*. July. Accessed October 8, 2015. http://www.w3.org/WAI/about.html.

Hommels, A. 2005. “Studying Obduracy in the City: Toward a Productive Fusion between Technology Studies and Urban Studies.” *Science, Technology & Human Values* 30(3): 323–51. doi:10.1177/0162243904271759.

Hughes, Bill. 2007. “Being Disabled: Towards a Critical Social Ontology for Disability Studies.” *Disability & Society* 22(7): 673–84. doi:10.1080/09687590701659527.

International Organization for Standardization. 2012. “ISO/IEC 40500:2012 - Information Technology -- W3C Web Content Accessibility Guidelines (WCAG) 2.0.” *ISO - International Organization for Standardization*. Accessed October 8, 2015. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=58625.

Jencik, Alicia. “Deconstructing Gender in New Orleans: The Impact of Patriarchy and Social Vulnerability Before and After a Natural Disaster.” PhD diss., University of New Orleans, 2010. Accessed February 26, 2013. http://scholarworks.uno.edu/cgi/viewcontent.cgi?article=2119&context=td.

Johnson, Jim. 1988. “Mixing Humans and Nonhumans Together: The Sociology of a Door-Closer.” *Social Problems* 35(3): 298–310. doi:10.2307/800624.

Kelty, Christopher M. 2008. *Two Bits: The Cultural Significance of Free Software*. Durham, NC: Duke University Press.

Kennedy, Helen. 2012. *Net Work: Ethics and Values in Web Design*. New York, NY: Palgrave Macmillan.

Kliehm, Martin. 2006. “To Hell with Joe Clark.” Text. *Learning the World*. August 31. Accessed October 8, 2015. http://learningtheworld.eu/2006/to-hell-with-joe-clark/.

Knowles, Eric D., and Brian S. Lowery. 2012. “Meritocracy, Self-Concerns, and Whites’ Denial of Racial Inequity.” *Self and Identity* 11(2): 202–22.

doi:10.1080/15298868.2010.542015.
Latour, Bruno. 1996. *Aramis, or the Love of Technology*. Translated by Catherine Porter. Cambridge, MA: Harvard University Press.
Layne, Linda. 2000. “The Cultural Fix: An Anthropological Contribution to Science and Technology Studies.” *Science, Technology, and Human Values* 25(4): 492–519.
Lazar, Jonathan, Daniel F. Goldstein, and Anne Taylor. 2015. *Ensuring Digital Accessibility through Process and Policy*. Amsterdam: Elsevier, MK Morgan Kaufmann.
Lennartz, Sven. 2008. “25 Useful WYSIWYG Editors Reviewed.” *Smashing Magazine*. May 6. Accessed October 4, 2015. http://www.smashingmagazine.com/2008/05/25-wysiwyg-editors-reviewed/.
Lessig, Lawrence. 2006. *Code Version 2.0*. 2nd ed. New York, NY: Basic Books.
Lie, Håkon Wium, and Bert Bos. 1999. *Cascading Style Sheets: Designing for the Web*. 2. ed. Boston, MA: Addison-Wesley.
Make WordPress Accessible. 2015. “Make WordPress Accessible.” *Make WordPress Accessible*. Accessed March 16. https://make.wordpress.org/accessibility/.
Makino, Takaki, and Doantam Phan. 2015. “Rolling out the Mobile-Friendly Update.” *Official Google Webmaster Central Blog*. April 21. Accessed October 14. http://googlewebmastercentral.blogspot.com/2015/04/rolling-out-mobile-friendly-update.html.
Marcotte, Ethan. 2010. “Responsive Web Design.” *A List Apart*. May 25. Accessed October 4, 2015. http://alistapart.com/article/responsive-web-design.
Mill, John Stuart, and George Sher. 2001. *Utilitarianism*. 2nd ed. Indianapolis, IN: Hackett Pub.
Mullenweg, Matt. 2013. “Tl Dr I Think the Team Can Prioritize….” *Make WordPress Accessible*. December 12. Accessed November 29, 2015. https://make.wordpress.org/accessibility/2013/12/12/tl-dr-i-think-the-team-can-prioritize/.
Mustaquim, Moyen M. 2015. “A Study of Universal Design in Everyday Life of Elderly Adults.” *Procedia Computer Science* 67: 57–66. doi:10.1016/j.procs.2015.09.249.
National Federation of the Blind. 2008. “National Federation of the Blind and Target Agree to Class Action Settlement: Target Will Ensure That Web Site Remains Accessible to Guests Who Use Assistive Technologies.” *DRA Legal*. August 27. Accessed April 14, 2011. http://www.dralegal.org/pressroom/press-releases/national-federation-of-the-blind-and-target-agree-to-class-action.
National Science Foundation. 2015. “The Internet - Mosaic: The Original Browser.” *America’s Investment in the Future*. Accessed October 5. https://www.nsf.gov/about/history/nsf0050/internet/mosaic.htm.

New York State Office of the Attorney General. 2004. “Spitzer Agreement to Make Web Sites Accessible to the Blind and Visually Impaired.” *NYS OAG Media Center*. August 19. Accessed April 14, 2011. http://www.ag.ny.gov/media_center/2004/aug/aug19a_04.html.

Nixon, Daniel. 2014. “Improving Bootstrap’s Woeful Accessibility.” *Daniel Nixon*. May 29. Accessed October 4, 2015. https://danielnixon.org/improving-bootstraps-woeful-accessibility/.

O’Connor, Michael. 2011. “Schema Blog.” July 20. Accessed October 4, 2015. http://blog.schema.org/2011/07/on-june-2-nd-we-announced-collaboration.html.

Oliver, Michael. 2009. *Understanding Disability: From Theory to Practice*. Second. New York, NY: Palgrave Macmillan.

Oudshoorn, Nelly, Els Rommes, and Marcelle Steinstra. 2004. “Configuring the User as Everybody: Gender and Design Cultures in Information and Communication Technologies.” *Science, Technology, and Human Values* 29(1): 30–63. doi:10.1177/0162243903259190.

Patrick, Deval L. 1996. “Memo to Senator Tom Harkin.” Government. *United States Department of Justice*. September 9. Accessed December 6, 2009. http://www.justice.gov/crt/foia/tal712.txt.

Pickering, Andrew. 1995. *The Mangle of Practice: Time, Agency, and Science*. 1st ed. Chicago, IL: University Of Chicago Press.

Pinch, Trevor J., and Wiebe E. Bijker. 1984. “The Social Construction of Facts and Artefacts: Or How the Sociology of Science and the Sociology of Technology Might Benefit Each Other.” *Social Studies of Science* 14(3): 399–441.

Pinsent Masons LLP. 2007. “Computer-Based Exam Discriminated against Blind Candidate.” *Out-Law.com: IT & E-Commerce Legal Help from International Law Firm Pinsent Masons*. January 24. Accessed April 14, 2011. http://www.out-law.com/page-7692.

Rauschmayer, Axel. 2014. *Speaking JavaScript: An In-Depth Guide for Programmers*. First edition. Sebastopol, CA: O’Reilly Media, Inc.

Richards, John T., Kyle Montague, and Vicki L. Hanson. 2012. “Web Accessibility as a Side Effect.” In *ASSETS ’12: Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, 79–86. New York, NY: ACM.

Rothberg, Madeleine. 2014. “Accessibility Metadata Project: Final Report.” *Accessibility Metadata Project*. October 20. Accessed October 4, 2015. http://www.a11ymetadata.org/accessibility-metadata-project-final-report/.

Russell, Andrew. 2015. “Nothing Special: Standards, Infrastructure, and Maintenance in the Great Age of American Innovation.” *The Committee on the Anthropology of Science, Technology, and Computing (CASTAC) Blog*. April 28. Accessed June 15. http://blog.castac.org/2015/04/nothing-special/.

Scherer, Marcia J. 2005. *Living in the State of Stuck: How Assistive Technology*

Impacts the Lives of People with Disabilities. 4th ed. Brookline, MA: Brookline Books.

Shakespeare, Tom, and Nicholas Watson. 2002. “The Social Model of Disability: An Outdated Ideology?” *Research in Social Science and Disability* 2: 9–28.

Singhal, Amit, and Matt Cutts. 2010. “Using Site Speed in Web Search Ranking.” *Official Google Webmaster Central Blog*. April 9. Accessed October 4, 2015. http://googlewebmastercentral.blogspot.com/2010/04/using-site-speed-in-web-search-ranking.html.

Smith, Aaron. 2013. “Smartphone Ownership 2013.” *Pew Research Center: Internet, Science & Tech*. June 5. Accessed November 29, 2015. http://www.pewinternet.org/2013/06/05/smartphone-ownership-2013/.

———. 2014. “African Americans and Technology Use.” *Pew Research Center: Internet, Science & Tech*. January 6. Accessed November 29, 2015. http://www.pewinternet.org/2014/01/06/african-americans-and-technology-use/.

———. 2015. “U.S. Smartphone Use in 2015.” *Pew Research Center: Internet, Science & Tech*. April 1. Accessed November 29. http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/.

Snow, Shane. 2010. “How Web Video SEO Is Finally Coming of Age.” Blog. *Mashable*. February 1. Accessed October 24, 2012. http://mashable.com/2010/02/01/web-video-seo/.

Sutton, Marcy. 2014. “JavaScript for Everybody: Accessibility of MVC Frameworks at JSConf EU.” *MarcySutton.com | Web Developer and Accessibility Advocate*. September. Accessed October 4, 2015. http://marcysutton.com/talk/jsconf-eu-2014/.

Sydell, Laura. 2010. “YouTube Launches Auto-Caption: A Boon for Deaf And Hearing Impaired.” Blog. *NPR: All Tech Considered*. March 4. Accessed October 27, 2011. http://www.npr.org/blogs/alltechconsidered/2010/03/youtube_launches_autocaption_f.html.

Thatcher, Jim. 2015. “Accessibility, Law and Target.com.” *JimThatcher.com*. Accessed October 4. http://jimthatcher.com/law-target.htm.

Twitter Bootstrap. 2014. “Twitter Bootstrap v3.3.0.” *GitHub*. October 29. Accessed October 4, 2015. https://github.com/twbs/bootstrap.

———. 2015. “Bootstrap · The World’s Most Popular Mobile-First and Responsive Front-End Framework.” Accessed November 26. http://getbootstrap.com/.

United Nations. 2006. “Convention on the Rights of Persons with Disabilities.” December 13. Accessed March 15, 2015. http://www.un.org/disabilities/convention/conventionfull.shtml.

———. 2015. “Ratifications and Signatories of the Convention and Its Optional Protocol.” Accessed March 16. http://www.un.org/disabilities/countries.asp?id=166.

United States Access Board. 2015. “Comparison Table of WCAG 2.0 to Existing 508 Standards.” *United States Access Board*. Accessed October 9. http://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-ict-refresh/background/comparison-table-of-wcag2-to-existing-508-standards.

US Census Bureau Public Information Office. 2010. “US Census Bureau Newsroom: Facts for Features & Special Editions: Facts for Features: 20th Anniversary of Americans with Disabilities Act: July 26.” May 26. Accessed July 31, 2012. http://www.census.gov/newsroom/releases/archives/facts_for_features_special_editions/cb10-ff13.html.

U.S. Department of Justice. 2010. “Fact Sheet Advance Notice of Proposed Rulemaking on Accessibility of Web Information and Services Provided by Entities Covered by the ADA.” August 3. Accessed April 14, 2011. http://www.ada.gov/anprm2010/factsht_web_anrpm_2010.htm.

———. 2012. “Statement of Interest of the United States of America in Opposition to Defendant's Motion for Judgment on the Pleadings.” May 15. Accessed November 29, 2015. http://www.ada.gov/briefs/netflix_SOI.pdf.

———. 2014. “Office of the Attorney General; Amendment of Americans with Disabilities Act Title II and Title III Regulations to Implement ADA Amendments Act of 2008.” *Federal Register*. January 30. Accessed September 16, 2015. https://www.federalregister.gov/articles/2014/01/30/2014-01668/office-of-the-attorney-general-amendment-of-americans-with-disabilities-act-title-ii-and-title-iii.

U.S. Equal Employment Opportunity Commission. 2009. “Section 902 Definition of the Term Disability.” November 21. Accessed April 14, 2011. http://www.eeoc.gov/policy/docs/902cm.html.

van de Poel, Ibo, and A. C. van Gorp. 2006. “The Need for Ethical Reflection in Engineering Design: The Relevance of Type of Design and Design Hierarchy.” *Science, Technology, and Human Values* 31(3): 333–60.

W3C. 2005. “How People with Disabilities Use the Web.” May 5. Accessed April 13, 2011. http://www.w3.org/WAI/EO/Drafts/PWD-Use-Web/.

———. 2014a. “WAI-ARIA Overview.” *The Website Accessibility Initiative*. June 12. Accessed October 2, 2015. https://www.w3.org/WAI/intro/aria.

———. 2014b. “HTML5.” October 28. Accessed November 29, 2015. http://www.w3.org/TR/html5/.

W3C WAI. 2014. “WAI-ARIA 1.0 Authoring Practices.” *Web Accessibility Initiative*. April 8. Accessed October 4, 2015. http://www.w3.org/WAI/PF/aria-practices/.

———. 2015a. “Introduction to Understanding WCAG 2.0.” *Understanding WCAG 2.0*. Accessed October 9. http://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html.

———. 2015b. “Understanding Conformance.” *Understanding WCAG 2.0*. Accessed October 9. http://www.w3.org/TR/UNDERSTANDING-WCAG20/conformance.html.

———. 2009c. “Web Content Accessibility Guidelines (WCAG) 2.0.” Organization. Accessed December 9. http://www.w3.org/TR/2008/REC-WCAG20-20081211/.

W3Techs. 2015. “Market Share Trends for Content Management Systems, October 2015.” *W3Techs Web Technology Surveys*. October. Accessed October 4. http://w3techs.com/technologies/history_overview/content_management.

Watson, Léonie. 2005. “What Is a Screen Reader?” *Humanising Technology Blog*. November 22. Accessed October 4, 2015. http://www.nomensa.com/blog/2005/what-is-a-screen-reader.

WCAG Working Group. 2015. “Participation in the Web Content Accessibility Guidelines Working Group.” *Web Accessibility Initiative*. Accessed October 9. http://www.w3.org/WAI/GL/participation.html.

WebAIM. 2013. “WebAIM: Links and Hypertext - Link Text and Appearance.” *Web Accessibility in Mind*. October 1. Accessed October 4, 2015. http://webaim.org/techniques/hypertext/link_text.

Winner, Langdon. 1988. *The Whale and the Reactor: A Search for Limits in an Age of High Technology*. Chicago, IL: University Of Chicago Press.

———. 1993. “Upon Opening the Black Box and Finding It Empty: Social Constructivism and the Philosophy of Technology.” *Science, Technology, and Human Values* 18(3): 362–78.

Wolbring, Gregor. 2008. “The Politics of Ableism.” *Development* 51: 252–58. doi:10.1057/dev.2008.17.

WordPress. 2015. “WordPress Features.” *WordPress*. Accessed October 5. https://wordpress.org/about/features/.

Zeldman, Jeffrey. 2003. *Designing with Web Standards*. 1st ed. Berkeley, CA: Peachpit Press.

APPENDICES

A: Quantitative Survey Questions

1. **Which of the following best describes your work environment?** ***(n=330)***

   a) I am a freelancer or a company of one. *(n=105)*

   b) I work for a company whose primary line of business is making websites. *(n=98)*

   c) I do website work for my company, whose primary line of business is something other than making websites. *(n=110)*

   d) Other: [free response] *(n=17)*

2. **What best describes your role within your work environment?** ***(n=327)***

   a) I am responsible for all elements of website design, including creating mockups, front-end, and back-end development. *(n=171)*

   b) I am only responsible for front-end design and implementation (mockups and/or HTML/CSS/JavaScript). *(n=55)*

   c) I am only responsible for back-end development (PHP/Rails/.NET/database/etc). *(n=57)*

   d) Other: [free response] *(n=44)*

3. **How many years of experience do you have in web design and/or development? *(n=328)***

   a) 0-1 *(n=22)*

   b) 2-5 *(n=126)*

   c) 5-9 *(n=94)*

   d) 10+ *(n=86)*

4. **What best describes your level of web-related education? *(n=327)***

   Select your highest level of education in a related field—IT, computer science, HCI, arts, marketing, etc.

   a) High School / GED *(n=46)*

   b) Some college, no degree earned *(n=98)*

   c) Associates (2-year degree) *(n=23)*

   d) Bachelors (4-year degree) *(n=136)*

   e) MBA *(n=7)*

   f) Masters *(n=13)*

   g) PhD *(n=4)*

5. **Please list any advanced certifications you hold, professional organizations you are a member of, or professional conferences you regularly attend. *(n=51)***

   e.g., Microsoft Certified DBA, Association for Computing Machinery (ACM)

member, JQuery conference, etc.

a) [free response]

6. **Which of the following best describes the type of school you attended? *(n=180)***

a) Trade school (ITT Tech or similar) *(n=13)*

b) Technology-focused college (MIT, RIT, or similar) *(n=54)*

c) Liberal arts college *(n=70)*

d) Other [free response] *(n=43)*[8]

8—For the purposes of analysis, some of the free responses were normalized to one of the three primary categories where it was appropriate to do so.

7. **Please indicate how much emphasis your college or trade school courses placed on the following areas:**

Table 1: Please indicate how much emphasis your college or trade school courses placed on the following areas

	1 (None)	**2 (Very Limited)**	**3 (Some)**	**4 (Fairly Strong)**	**5 (Central Focus)**
“Best Practices” Design	n=39	n=46	n=48	n=38	n=6
User Testing	n=54	n=57	n=41	n=21	n=4
Including Diverse Groups in User Testing	n=79	n=49	n=33	n=15	n=1
Including People with Disabilities in User Testing	n=106	n=39	n=22	n=9	n=0
Designing to W3C Language Standards (HTML/CSS/XML/etc)	n=58	n=42	n=38	n=28	n=11
Designing to Disability/Accessibility Standards (W3C WCAG, Section 508)	n=91	n=40	n=28	n=14	n=4

8. **Do you code to W3C standards for HTML, CSS, XML, and the like?** ***(n=322)***

a) Yes, whenever possible. *(n=258)*

b) I'd like to, but can't, due to time and budget constraints. *(n=32)*

c) I'd like to, but my manager doesn't think it's important. *(n=4)*

d) Only when it is required by the client. *(n=9)*

e) No, because I don't think the standards are useful. *(n=1)*

f) No, because I don't know how. *(n=7)*

g) No, because I didn't know these standards existed. *(n=0)*

h) Other: [free response] *(n=11)*

9. **Do you code to standards for web accessibility for people with disabilities? *(n=326)***

   Such as W3C's Web Content Accessibility Guidelines (WCAG) or Section 508 standards.

   a) Yes, whenever possible. *(n=114)*

   b) I'd like to, but can't, due to time and budget constraints. *(n=67)*

   c) I'd like to, but my manager doesn't think it's important. *(n=15)*

   d) Only when it is required by the client. *(n=81)*

   e) No, because I don't think the standards are useful. *(n=7)*

   f) No, because I don't know how. *(n=21)*

   g) No, because I didn't know these standards existed. *(n=15)*

   h) Other: [free response] *(n=6)*

10. **Which of the following best describes your organization's practices regarding user groups? *(n=329)***

    a) User groups are involved in every step of the design and development process. *(n=20)*

b) User groups are brought in at the alpha- or beta-testing stage. *(n=47)*

c) We employ user groups in a very limited capacity. *(n=99)*

d) We do not employ user groups. *(n=163)*

11. How important are the following in the websites you create?

Table 2: How important are the following in the websites you create?

	1 (Not Important)	**2 (Somewhat Important)**	**3 (Important)**	**4 (Very Important)**	**5 (Central Focus)**
Short Development Cycle	14	94	98	81	37
Sites Work for a Variety of Users	6	34	90	146	48
Sites Work in a Variety of Web Browsers	4	19	55	176	70
Sites Work for Users with Disabilities	93	116	60	45	8
Page Speed / Responsiveness	3	27	89	144	61
Expressing my Creativity	69	106	80	48	18
Standards Compliance	24	47	118	108	26

12. What automated testing or validation tools do you use?

(n=164)

a) [free response]

13. If you write code, what tool do you primarily use? *(n=320)*

a) Standalone Editor (Dreamweaver, etc) *(n=101)*

b) CMS Editor (WordPress, Drupal, etc) *(n=10)*

c) Text Editor (gEdit, jEdit, Xcode, Notepad, TextEdit, etc) *(n=209)*

14. What online communities do you frequent related to web design or development? *(n=287)*

e.g., Reddit, A List Apart, StackOverflow, etc.

a) [free response]

15. What city or geographic area do you work in? *(n=311)*

e.g., New York City, London, Sydney, Shanghai, etc

a) [free response][9]

9—For the purposes of analysis, free responses were normalized to country or region of origin by the investigator.