"It's Just Everything Outside of the IDE that's the Problem":

Information Seeking by Software Developers with Visual Impairments

Kevin M. Storer*
University of California, Irvine
Irvine, CA, USA
storerk@uci.edu

Harini Sampath
Google Inc.
Seattle, WA, USA
harinis@google.com

M. Alice Merrick
Google Inc.
Seattle, WA, USA
amerrick@google.com

ABSTRACT

Many efforts to increase accessibility in coding for developers with visual impairments (DWVI) focus on supporting interactions with development tools. But, to understand how to appropriately modify and write source code, developers must seek information from a variety of disparate and highly technical sources. DWVI might benefit from technological support in this process. But, it is unclear what accessibility issues arise in technical information sources, whether accessibility impacts strategies for seeking technical information, or how best to support DWVI in information seeking. We conducted observations and interviews with twelve DWVI, to explore their information behaviors. We found that DWVI seek information in many of the same sources as their sighted peers, and accessibility issues in technical information sources were similar to those in nontechnical sources. But, despite these similarities, examining development as an information seeking process highlighted the role of contextual and social factors in determining accessibility for DWVI.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in accessibility**; • **Social and professional topics** → **People with disabilities**.

KEYWORDS

Software Development, Visual Impairments, Information Behavior, Accessibility

ACM Reference Format:
Kevin M. Storer, Harini Sampath, and M. Alice Merrick. 2021. "It's Just Everything Outside of the IDE that's the Problem":: Information Seeking by Software Developers with Visual Impairments. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3411764.3445090

1 INTRODUCTION

Digital technologies have produced fundamental changes in the global economy which make technical computing skills, like programming, a gateway to abundant and lucrative job opportunities [2, 3]. Broadening access to programming skills is important for ensuring these opportunities are equitably distributed and attainable for all members of society. The opportunities afforded by programming skills may be particularly important for people with visual impairments (PWVI), who currently experience significant economic inequities [1, 25, 55]. Accordingly, there has been broad academic interest in increasing the nonvisual accessibility of tools involved in programming, like Interactive Development Environments (IDEs) and source code [10, 26, 41, 53, 62, 67–71], diagrams [33, 34, 46], graphs [11, 12, 32, 44, 45, 48, 79], and even in the pedagogies used to teach Computer Science [36]. Improving accessibility in each of these areas is important for broadening participation of PWVI in technical occupations. However, providing individual, accessible tools—even if they offer a high level of usability—cannot necessarily guarantee the development experience of developers with visual impairments (DWVI) will be comparable to that of developers with no specified visual impairments (DWNSVI).

To appropriately modify and write source code, developers must seek and synthesize information across many disparate and highly technical information sources. Information seeking and synthesis is a particularly difficult aspect of professional coding, regardless of visual abilities. So, a variety of novel technologies have been designed specifically to support this process [16, 37–39, 52, 59, 61, 75, 80]. These tools have been built upon a large body of research examining the information behaviors of DWNSVI [17, 22, 30, 31, 47, 49, 51, 60]. DWVI might also benefit from support in information seeking. However, there is no analogous body of work examining the information behaviors of DWVI. But, outside the context of software development, studies of everyday web use show that PWVI who use screen readers have fundamentally different strategies for seeking and assessing information than people who use vision to navigate the web [5, 13, 14, 54, 65, 72]. So, it cannot be assumed that DWVIs' information behaviors mirror those of DWNSVI, nor that techniques for supporting DWNSVIs' information seeking and synthesis will benefit DWVI.

To explore how DWVI seek and synthesize information in their professional lives, we conducted observations and in-depth interviews with 12 professional DWVI. We asked them to search online API documentation for a package meeting a specific need, reflect on the decisions they made in searching for that package, and connect this to their use of technical information sources in their professional life. We used this task to scaffold in-depth interviews about other information sources used in coding, accessibility issues in each of these sources, and the factors considered when deciding which source to select.

We found that DWVI used many of the same information sources previously documented in studies of DWNSVI, and these technical

*Research completed while at Google Inc.

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05.
https://doi.org/10.1145/3411764.3445090

sources suffered from fundamental accessibility issues common to nontechnical sources. Yet, despite these similarities, we found that examining development as an information seeking process highlighted the role of contextual and social factors in determining accessibility for DWVI. We outline our findings below to compare our observations to previous explorations of DWNSVIs' information behaviors and accessibility issues in nontechnical information sources, and to identify ways in which accessibility in development may be impacted by factors outside of the IDE.

2 RELATED WORK

2.1 Nonvisual Software Development

Because of the growing need for PWVI to have technical computing skills in the shifting economy, there has been wide academic interest in supporting accessibility in computer programming and in STEM disciplines, more broadly. These efforts have addressed diverse domains, including increasing nonvisual accessibility in programming education (see [36], for a recent review), in scientific and mathematical graphs [11, 12, 32, 44, 45, 48, 79], and in UML models [33, 34, 46].

In this domain, a large portion of research efforts have been directed at increasing the nonvisual accessibility of source code and IDEs [53, 67, 69–71], because the line-by-line outputs of screen readers make it difficult for DWVI to gain high-level understanding of the structure of source code [6, 7, 9]. In response, many works have aimed to provide explicit information about nesting and hierarchical structure of source code. For example, Baker et al. [10] designed *StructJumper*, which generated an accessible tree structure to provide hierarchical information about the codebase that was more easily traversed with screen readers than a source file. Similarly, Smith et al. [68] constructed a hierarchical tree for conveying source code structure using unique auditory cues. Hutchinson and Metatla [41] tested the viability of different types of auditory cues, including speech and spearcons, for conveying code structure nonvisually. Potluri et al. [62] designed *CodeTalk*, which used auditory feedback to provide information about code structure, and to indicate syntax error highlights and changes in variable values as a program is executed. Alternatively, Falase et al. [26] explored the use of slide potentiometers to display such information about code structure and indentation levels through haptic channels.

While the highly visual structure of source code is an important problem to address in professional software development, it is likely not the only accessibility issue impacting DWVIs' development experience. Yet, very few works have empirically explored other barriers DWVI face in their work lives. In one exception, Mealin and Murphy-Hill [56] found evidence of a variety of other accessibility issues in development, including reliance on sighted coworkers creating embarrassment and difficulties managing the large amount of information gathered in development. Here, we build on this work by examining accessibility issues in information seeking and technical information sources, outside of source code navigation and IDEs.

2.2 Information Behaviors of Developers with No Specified Visual Impairments

Due to the information intensive nature of engineering professions, engineers were among the first groups of people to be studied as users of information [27]. Although many studies of engineers' information behaviors have been conducted outside the context of software development ([8, 50, 63], to name only a few), research suggests that information seeking in software development may be a particularly complex process. Developers have been found to employ a large number of information sources in their professional work, including technical blogs, expert others, online tutorials [51], and shared code bases [49, 64], to meet a diverse set of information needs, like just-in-time learning, translating syntax from one programming language to another, remembering implementation details [17] and maintaining mental models of shared code repositories [49]. Understanding the information needs of developers can be particularly complex, as needs have been found to depend on the programming task at hand [47, 51], the project phase [31, 51], and other contextual factors, like personal expertise and job role [31]. Even within a singular programming task, like code review, developers' information needs have been found to be particularly diverse and dependent upon individual priorities [60].

Studies of engineering professionals, outside the context of software development, consistently find that engineers prioritize saving time when seeking information and, as such, select information sources based on its "accessibility," here meaning ease of access, disregarding other important factors, like quality and breadth of information (see [27] for an overview). Studying software developers' information behaviors through theoretical models remains rare [30]. But, recognizing the importance of saving time for engineers, a large number of academic works presenting novel digital tools aimed at easing information seeking and synthesis for software developers. Many of these systems ease information seeking by integrating multiple, distributed information sources. For example, Brandt et al.'s *Blueprint* integrated web search functionality into the IDE [16]. Conversely, Hartmann et al.'s *D.mix* supported code editing functionality in the web browser [37]. Oney and Brandt's *Codelets* integrated code examples with interactive explanations into the IDE [59]. Similarly, Ponzanelli et al.'s *Seahawk* used a custom Eclipse plugin to display Stack Overflow insights in the IDE [61]. Hoffman et al.'s *Assieme* synthesized information in JAR files, API documentation, and webpages with examples and plain-language explanations [38]. Other systems aim to ease developers' information seeking, by scaffolding decision making. For instance, Buse and Zimmerman [22] explored whether automated software analysis tools might provide additional indicators to scaffold information seeking. Holmes et al.'s *Strathcona* automatically recommended relevant examples of API usage, based on the source code elements selected by developers [39]. Stylos and Myers' *Mica* augmented web search for API classes and methods, by providing appropriate results based on plain-language descriptions of desired code functionality [75]. Similarly, Liu et al.'s *Unakite* provided information about the tradeoffs of seemingly-equivalent codeblocks to assist developers in deciding which of many possible solutions best meets their needs [52].

Because of the complex information needs of developers, these efforts to scaffold information seeking and synthesis are important. Yet, each of the above works examined only DWNSVI. But, given the importance of information sources' ease of access for engineers [27], it is likely that DWVIs' information seeking strategies may be impacted by technical accessibility issues in the information sources employed in software development. So, it is not clear that these insights translate to supporting DWVI. Our study builds on this existing literature, by examining whether and how DWVIs' information behaviors differ from the information behaviors of DWNSVI, as previously documented in these works.

2.3 Information Behaviors of Web Users with Visual Impairments

The vast majority of research examining the information behaviors of PWVI has been conducted in the domain of casual web browsing. Because online information resources are prone to accessibility issues, many of these efforts have aimed to increase the overall accessibility of the web through technical augmentation ([15, 66, 78, 81] to name only a few).

Empirical studies of web browsing have shown that visual abilities and screen reader use produce differences in the information behaviors of web users with and without visual impairments. For example, Bigham et al. [13] found that accessibility issues, like missing alt text or heavy use of dynamic content, not only increase navigation time, but reduce PWVIs' overall engagement with inaccessible sites. But, Vigo et al. [78] found that it is difficult to determine whether a site returned by a web search will be accessible before a user navigates to it, causing PWVI to employ suboptimal information seeking strategies, like probing search results at random. Similarly, Sahib et al. [65] found that, because of the temporal cost of browsing the web with screen readers, PWVI have fundamentally different strategies than their sighted peers for formulating search queries and exploring and managing search results. Stockman and Metatla [72] found that differences in the ways web users with and without visual impairments traverse the web produce different conceptualizations of how web content is structured. Bigham et al. [14] found that inaccessibility in web content makes it difficult for web users with visual impairments to know what information is actually present in online sources. Additionally, Abdolrahmani and Kuber [5] found that people with and without visual impairments employ different strategies for assessing the credibility of online information sources.

While these studies demonstrate many ways in which visual abilities and screen reader usage influence information behaviors, they have primarily focused on casual web use and nontechnical information sources. So, these works suggest it is likely that the information behaviors of DWVI differ from those of their sighted peers. But, it cannot be assumed that these insights and suggestions for designing accessible web pages will translate directly to the highly technical information sources that are sought in coding. So, our work expands on these insights by examining whether and how visual abilities and screen reader use affect information behaviors within the specialized information sources used in software development.

3 METHODS

3.1 Participants

For our interviews, we recruited twelve DWVI, by contacting a previously curated list of people interested in participating in accessibility research. We observed that one member of the existing list also posted our study invitation to a mailing list for DWVI. We cannot distinguish the exact number of participants recruited from each source. In order to be included in this study, participants needed to indicate that they 1) regularly use screen readers to access digital devices and 2) write or modify source code as part of their professional work. There were no other inclusion criteria. Each participant received a $175 thank you gift for their time, commensurate with their specialized skill set.

Two participants identified as women and ten identified as men. Seven participants were born with their visual impairment, and only one had experienced a visual impairment for less than two years. Six participants described themselves as completely blind. Three interviewees described themselves as having some light perception. Two interviewees described themselves as low-vision. One participant declined to describe his visual abilities in detail. To be included in our study, participants were required to indicate modifying or writing source code as part of their job. But, job roles were diverse, and only four participants indicated working as software engineers. Similarly, to be included in our study, participants were required to use screen readers regularly. But, many used other assistive technologies, also. A full list of participant identifiers, demographics, visual abilities, job roles, and other assistive technology use is shown in Table 1.

3.2 Procedure

We conducted all sessions remotely using teleconferencing software, primarily due to local social distancing mandates. But, we note that remote interviews have been used in accessibility studies, prior to these mandates, to increase the geographic diversity and distance over which recruiting can occur [74]. Each remote meeting was scheduled for 105 minutes, including the introduction to the study, the research session in which data was collected, and a debriefing period in which participants were invited to ask questions about the session. Research sessions lasted between 94 minutes and 113 minutes, averaging 103 minutes each. In total, we collected and analyzed approximately 21 hours of data from our twelve participants.

Each research session consisted of two sections: 1) An observational section and 2) an in-depth interview section. During the observational section, participants were asked to share a window on their screen over the teleconferencing software and to perform a predefined task, which required navigating online API documentation for the open-source Go programming language, to find a package and function given specific criteria. Because we did not require our participants to have prior experience with Go, participants were asked to seek a function which performs simple String manipulation, which is a common programming task in many programming languages. We asked participants to think aloud during this process, as much as was comfortable for them. We additionally probed participants to reflect upon their experience in pivotal moments, or when we observed navigation issues. Observation

Table 1: Participant Demographics

ID	Job Description	Age	Gender	Visual Impairment, Duration	Other Assistive Devices
P1	Accessibility Specialist	61+	M	Light Perception, <2 Years	None
P2	Software Engineer	31-40	M	Completely Blind, >7 Years	None
P3	Accessibility Specialist	31-40	M	Low-Vision, >18 Years	Magnification, Braille Terminal
P4	Software Engineer	24-30	M	Light Perception, Since Birth	Braille Terminal
P5	Security Specialist	31-40	M	Completely Blind, Since Birth	None
P6	Software Engineer	31-40	M	Light Perception, Since Birth	None
P7	Accessibility Specialist	31-40	M	Completely Blind, Since Birth	Braille Terminal
P8	Web Developer	31-40	M	Light Perception, Since Birth	None
P9	Software Engineer	24-30	W	Low Vision, >7 Years	Magnification
P10	Accessibility Specialist	31-40	M	Not Reported, Since Birth	Braille Terminal
P11	Accessibility Specialist	31-40	M	Completely Blind, >18 Years	None
P12	Accessibility Specialist	31-40	W	Completely Blind, Since Birth	Braille Terminal

sections were designed to last 30 minutes and were ended around this time, regardless of whether the assigned task was completed.

We then used these observations to scaffold a longer conversation in in-depth interviews about information seeking while engaged in professional coding, which were designed to last 60 minutes. We adopted an approach to interviewing based on the Critical Incident Technique [29]. Broadly speaking, this technique asks participants to recount a recent or eventful time in which they were engaged in the behaviors of interest–here, software development. This approach has been shown to be a reliable method for gaining valid insights about in-situ behaviors retrospectively and has been used effectively in studies of information behaviors in other contexts [28, 77]. As participants recounted salient incidents of software development, we probed for more information about 1) what sources they used to find technical information, 2) the reasons they used these sources, 3) what types of accessibility issues they experienced using these sources, 4) whether alternative sources provided similar information, and 5) how they chose to use the selected source. However, Critical Incident interviews are only loosely structured, to encourage participants' free-form reflection [29]. So, often participants shared insights outside the four areas we directly probed.

Although our methods were designed to serve as sequential sections, participants often described their typical information seeking behaviors during the observation section and referred to the documentation example used in observations when describing their information behaviors during interviews. So, while the intended purpose of the observation section was to scaffold reflection during interviews, it was not strictly a priming exercise. Rather, informative data were gathered in both sections, and collected throughout the entirety of each 105 minute session. Here, the findings reported resulted primarily from our in-depth interviews. But, as participants often connected our predefined task to their typical information seeking practices, we report all generalizable findings on technical information source accessibility and information seeking together, regardless of the specific study section in which they arose.

3.3 Analysis

The first author moderated all research sessions and took notes throughout. The second and third author observed these sessions and submitted interview questions to the first author over a private messenger. Each research session was audio- and video- recorded and transcribed in full by a reputable third-party transcription service, yielding 313 pages in total. Within these transcripts and notes, we excerpted each passage in which participants indicated using a specific information source. The first author then conducted an inductive qualitative analysis, which consisted of three steps. First, he classified all excerpts into categories describing the types of information sources discussed. Second, he developed open codes to describe individual participants' use of each type of information source. Third, he developed thematic codes from these open codes to describe commonalities in the use of each type of information source across participants. Finally, the first author shared his codebook, containing thematic codes and representative quotes from participants, with the other members of the research team, who reviewed them for accuracy and clarity. After reviewing the codebook separately, team members met to discuss and confirm agreement together. No significant disagreements surfaced.

4 FINDINGS

We found that participants discussed seven types of information sources, outside of source code and IDEs. Four identified sources—Official Documentation, Blogs and Tutorials, Forums, and Textbooks—functioned strictly as information sources. Three identified sources—Web Search, Other People, and Memory—were described by participants as information sources, although they have alternative functions. There were strong commonalities in participants' descriptions of their use of strictly informational sources, typically expressed in terms of 1) the technical depth of information provided, and 2) the prior knowledge required to use them. But, information sources with alternative functions had highly specific uses. So, their utility was conceptualized differently than other identified information sources. Additionally, strictly informational sources had readily identifiable technical accessibility issues. But, the accessibility issues of information sources with alternative functions

were extremely nuanced. Consequently, selecting between strictly informational sources was typically described as a decision made based on participants' familiarity with a given topic. But, choices about employing information sources with alternative functions were more complex and dependent on personal and social factors.

4.1 Strictly Informational Sources

Across all participants, 1) Official Documentation, 2) Blogs and Tutorials, 3) Forums, and 4) Textbooks, were the most commonly discussed information sources which functioned strictly as information sources. Typically, the utility of these sources was expressed in terms of 1) the technical depth of information provided, and 2) the prior knowledge required to use them. Each of these sources suffered from technical accessibility issues, which impacted their usability for our participants.

4.1.1 Official Documentation. All twelve participants described using official documentation for programming languages and APIs in their software development activities. Collectively, participants indicated official documentation was useful for finding low-level technical details and required some prior knowledge to use.

Six participants {P2, P3, P4, P6, P9, P10} indicated that they used official documentation when they needed low-level technical details. For example, {P10} stated that he turns to official documentation when he needs to know *"what kind of information does the API provide, what kind of parameters can I use in my function when I call the API… and what kind of response the API will provide."* {P3} indicated using official documentation when *"the compiler isn't happy and you have to go dig for whatever it is that makes it work."* {P2} recounted an incident, in which *"yesterday, I was looking at [official documentation] that they provided for how to integrate stuff, improve the exact [application function] and stuff like that… some things are being deprecated so all that stuff is there on their [documentation]."*

Eight participants {P1, P2, P4, P6, P8, P10, P11, P12} indicated that official documentation requires prior knowledge to use. For instance, {P4} shared that *"If I'm looking at API docs, I generally know what the API is and what it does...I will generally start at a higher level...and then I can drill down."* Similarly, {P6} noted that he uses official documentation *"when I know I need to look at information on a specific package."* Conversely, {P11} recounted a time in which his lack of prior knowledge rendered official documentation unusable, stating, *"once I did find the documentation, I was like, 'OK, great. But, I'm new to [this language]—what does the actual syntax look like if I want to do operation X, here?'"*

Six participants {P1, P3, P4, P7, P8, P10}, identified specific accessibility issues in official documentation. In fact, {P4} noted that, *"Assuming that your IDE is accessible, [coding is] really a pretty level playing field—it's just everything outside of the IDE that's the problem...Like documentation."* Specific issues raised included differences in behaviors across screen readers {P1}, visual styling producing confusing markups {P3}, loading too much content degrading performance of screen readers {P7}, lack of proper headers {P8}, the use of inaccessible visual content {P10}, and inconsistent formatting across pages on a single site, which prevents learning document structure {P4, P10}.

4.1.2 Blogs and Tutorials. Eleven participants (all but {P4}) described using blogs and tutorials in their software development activities. Collectively, participants indicated blogs and tutorials were useful for finding high-level general information and did not require any prior knowledge to use.

Four participants {P1, P2, P3, P11} spoke to the use of blogs and tutorials for finding high-level general information. For instance, {P2} used tutorials to find *"some overview"* of a topic and {P3} indicated that a tutorial will show users *"all the basics."* {P1} pointed to the limitations of the high-level general information provided by tutorials, stating they *"start teaching you, 'OK, this is what a list is, this is how you initialize one'...But, they don't teach you every single method that's in a list."* Similarly, {P11} noted, *"they're not like the ground-up instruction or anything...It's not supposed to be an in-depth 'how-to-do-everything-with-[framework].'"*

Eight participants {P1, P2, P3, P5, P6, P7, P8, P9} indicated using blogs and tutorials in cases where they had little or no prior knowledge about an application or language. {P1} noted that he starts *"stressing out when you tell me things that you assume I should know. [So, in this case,] I started out by reading a blog post."* {P2} used tutorials in cases where he is working on *"something that I'm not familiar with, or maybe I know it's out there and I haven't used it...I'm just trying to learn."* Likewise, {P9} used tutorials *"to see how people have done it...to see if there's something simplified enough that I can understand it."*

Nine participants {P1, P2, P5, P6, P7, P8, P9, P10, P12} identified specific accessibility issues in blogs and tutorials. Most commonly, participants indicated that blogs and tutorials make heavy use of visual instructional content that is not properly described {P1, P2, P5, P6, P7, P8, P9, P12}. {P5} suggested that accessibility issues are especially prevalent in blogs and tutorials, noting *"you tend to see [issues] more in tutorials, because people will just take some screenshots and add a couple of paragraphs and call it done."* {P7} expressed desire for tutorials designed specifically for DWVI, noting *"I wish they would at least write the entire code in the transcription...[or show] how somebody using a screen reader would be [doing it], by recording that video using a screen reader, basically."* {P10} even indicated *"I have yet to find...a tutorial website that I am proud of"* in terms of its accessibility.

4.1.3 Forums. Eight participants {P2, P3, P4, P5, P6, P9, P10, P11} described using technical forums in their software development activities. Most often, participants discussed using Stack Overflow in particular. But, {P3} and {P10} also indicated using other unnamed forums. Collectively, participants indicated forums were useful for finding high-level general information and required some prior knowledge to use.

Five participants {P2, P3, P5, P10, P11} indicated using forums for finding high-level general information. For example, {P3} recalled using forums for finding only *"really simple examples."* Other participants used broad language to describe their needs when asked about their use of forums, like {P5} who shared forums are used to *"see if I can find something"* and {P10} who noted using forums for finding *"some stuff."* {P2} noted the limitations of the generality of the information available on forums, saying *"Stack Overflow only talks about certain things. Right? So, if people have questions and they have a certain question, maybe I don't have that question."*

Interestingly, although forums allow users to ask specific questions about low-level details, only {P11} indicated using this feature and *"apparently nobody knew [the answer] because I never heard anything back."*

Five participants {P2, P3, P5, P6, P10} noted that forums require some prior knowledge to be used effectively. In particular, participants needed to have effectively formulated and articulated the problems they were facing, in order to find analogous and relevant answers. For instance, {P3} searched for *"people posting in forums, having similar questions."* {P5} used forums to find *"something where someone is talking about trying to do the same thing [to] figure out what package they used."* Similarly, {P10} indicated turning to forums to determine *"how their function is working to make what I need work."*

Six participants {P4, P5, P6, P9, P10} remarked upon the accessibility of forums and, in particular, Stack Overflow. Of these participants, only {P10} felt negatively about forums' accessibility. He noted that these are not particular to Stack Overflow and did not identify specific issues. Rather, he suggested that *"the problem with any forum website is that accessibility is the last thing they think about."* {P4, P5, P6, P9} each indicated that Stack Overflow is particularly accessible due to their proper use of headings {P5, P9} and the use of LaTeX for *"math specific things...unless someone links to an off-site image...But, usually they get yelled at for it"* {P4}.

4.1.4 Textbooks. Seven participants {P1, P2, P3, P4, P9, P11, P12} described using textbooks in their software development activities. Collectively, participants indicated textbooks were useful for finding both high-level general information and low-level technical details, and did not require any prior knowledge to use.

Six participants {P2, P3, P4, P9, P11, P12} indicated using textbooks for finding high-level general information. Each of these participants described using textbooks for obtaining information about how to get started with a specific language or framework. Importantly, three of these participants also indicated using textbooks for finding low-level technical details {P2, P4, P12}. {P2} noted that textbooks can be especially useful in relation to other sources, because *"the book talks about [it] a little bit more in detail. So, yes, sometimes that information is presented somewhere [else, but] in the book it is presented very nicely with examples and stuff like that. So, I've found that approach very helpful."* Similarly, {P12} shared, *"If I need to look at a certain feature that we are working on developing...I look into my textbook [to figure out] the best way I could do it."*

Six participants {P2, P3, P4, P9, P11, P12} spoke to the fact that textbooks do not require prior knowledge to use. In fact, discussions of textbooks most often arouse in conversations about learning new skills. For instance, {P12} shared that before going to other sources, she would search for *"a textbook, or like a guide of some kind."* When his company was migrating to a new framework, {P2} shared that *"the very first thing I did was I downloaded a [framework-specific] book."* {P4}'s company was going through a similar migration, and he shared, *"that's also an interesting new thing that I need to learn, which is why I went and found a book on it."*

Generally, participants indicated using electronic textbooks {P1, P2, P4, P9, P11, P12}, sometimes from publishers of books for blind and low-vision readers {P1, P11} which were largely accessible, except one which misplaced line-break indicators in code examples {P11}. But, {P3} used a physical textbook to find information. He recounted a story of learning a new programming language to develop a personalized accessibility tool, sharing:

> *"I probably spent six months or seven months doing nothing but learning [this language] and reading the manual and reading every book on [it] that I could find...I mean it was two paper books...there is nothing [else] that gets people 'zero to 60'...So, we need to get all of the information that is in those books, into an accessible format. For me, I wouldn't have been able to learn all of this stuff without being able to use a little bit of a paper book. And as I was working on this project I started losing vision. And so it became an imperative for me to get this to work."*

So, while textbooks were especially useful resources, whether they were accessible largely depended on their format.

4.2 Sources with Alternative Functions

In addition to the strictly informational sources discussed above, participants also described using information sources which had other functions than providing information. Specifically, these sources most commonly included 1) Web Search, 2) Other People, and 3) Memory. In contrast to strictly information sources, sources with alternative functions often served highly-specific needs. As such, their usage and accessibility issues were described in unique ways.

4.2.1 Web Search. All twelve participants described using web search sites, like Google, in their recent software development activities. Though not squarely an information source itself, when asked *where* they found a specific piece of information, all twelve participants, at least once in our conversations, suggested they found it on web search. But, because participants were actually referring to finding information from sources *returned by* web search, when asked *what needs were met* by web search sites, they often described their needs in extremely vague terms. For example, web search was described as being used for *"looking for other things"* –{P1}, for getting *"some information"* –{P2}, for *"trying to figure out how to do something"* –{P4}, *"a means of getting the most relevant [information] I can"* –{P9}, to *"see if I can find a different [source]"* –{P10}, for *"plumbing the web"* –{P11}, and to *"see if I find any books, documents or whatever is available"* –{P12}.

But, because web search primarily served as a vehicle for connecting participants to the other discussed sources, participants' perceptions of its accessibility were nuanced. Where the accessibility of popular search engines was generally perceived positively {P4, P11}, the variability of the accessibility of the sources returned as search results created a frustrating experience. {P2} shared *"let's say I did a quick [search] and then I landed on this website [which] is not accessible—I have to find a way to navigate the website [or] it's going to completely stop me from gathering that information."* Additionally, {P11} noted that receiving results, only to discover they are inaccessible is *"entirely frustrating...because at that point, like, 'Oh, great! This search result sounded like exactly what I needed. Oh, great! They're very painful to use or I can't even read the contents,' and I have to go back."* So, while web search sites were largely accessible, the variable accessibility of returned results created an

mixed experience. Additionally, because the web search sites return valid results—their intended function—inaccessibility in the sources returned may be particularly frustrating, in that DWVI, like {P11}, may know they have found the right information, but are then prevented from retrieving it.

4.2.2 Other People. Eleven participants (all but {P4}) described using other people as information sources in their recent software development activities. Notably, other people sometimes served to provide information that could not feasibly be provided in any other format. For example, {P3} shared that he typically likes to work alone, but needs to exchange information with others because *"when I'm working with another human being, it's really to solve a problem that the machine has already failed to address. And so documentation can be great, you can interconnect all of these topics and teach us what we need to know. But, it can't actively problem solve."* Similarly, {P2} described exchanging information with other teammates to divide labor in *"weekly meetings [where] we talk about work we have done in the past week, what was assigned to us… if there is a request that came in, or stuff like that."* In these cases of collaborative problem-solving and distributing labor amongst teammates, it is unclear whether there are, or could be, alternative sources for the information provided.

Similarly, seven participants {P1, P2, P3, P6, P7, P8, P9} indicated using other people as information sources, because they had domain-specific knowledge. For example, {P9} described having "weekly meetings with my boss [because] I don't know much of anything about [this API]...because he knows a lot more [about it] than I do." {P3} used others as information sources to find mathematical information specific to the scientific field in which his application was being used. {P7} used designers working on his team to find information specific to their job role, including *"how the different flows work...and the reasoning behind it."* Additionally, {P1, P2, P6, P8} each used other DWVI as sources for finding information directly pertaining to accessibility issues in development. {P1} shared, for instance, *"I was talking to a deaf-blind [programmer]. And he said to me, [description of accessibility issue]. So, I said 'What you really should do, then, is [description of solution]."*

Perceptions of the (in)accessibility of using other people as an information source were particularly nuanced. Five participants {P1, P2, P6, P10} used other people as workarounds for inaccessible information sources, suggesting other people are particularly accessible. But, four participants {P7, P9, P11, P12} indicated times where sighted others *introduced* accessibility issues in their information seeking. In a positive example, {P2} noted that, because of his supportive coworkers, relying on sighted assistance is a perfectly acceptable workaround for inaccessibility in other sources. He shared, *"Honestly, when I was taking classes in school the same thing happened. If I was quiet, I was losing a lot of information. But, if I was speaking, I was getting more information...So, honestly, I don't feel, like, offended or anything if I can't see. I know there are people around me who can guide me or tell me what's going on. So, I don't feel odd, at all."* Conversely, {P12} indicated that whether other people are helpful or a hindrance *"depends on how your relationship is with them...Sometimes the new engineers, when they join, have no idea what to do, as far as assisting… It's not that they don't want to do it, it's just they don't know how to do it and, it's like, they don't want to offend somebody. So, they don't ask the questions to learn… So, I [ask] mainly who I'm close with."* In this way, whether other people were accessible or inaccessible information sources depended upon the empathy, knowledge, and collaborative abilities of sighted others.

4.2.3 Memory. Nine participants {P1, P2, P3, P4, P5, P6, P8, P9, P10} described using their memories as information sources in their recent software development activities.

Four participants {P3, P4, P5, P8} described using their memories for storing specific technical information. {P3} discussed having previously maintained mental models of codebases, which was no longer feasible as he is *"getting older,"* as did {P5}, who noted that it helps *"having written most of it."* {P8} described remembering syntax for command line operations because he *"started using computers back in the early DOS days. So, when I got in and it was like, 'oh, to switch directories you type cd/ blah, blah, blah,' I was like, 'I remember this stuff!'"* {P4} also stored some programming syntax in memory, noting *"if you commit changes to a repository 12 times a week, you're going to eventually remember the syntax to do that."*

Four participants {P1, P5, P6, P10} discussed using their memories to guide their information seeking. For example, {P5} indicated that, because he remembered that *"on [one] site, they have a pretty useful tutorial that touches on a lot of the different language features. So, I would jump to that, because I know that I've flicked through it before and I'm pretty sure that it has [the syntax I want]."* Similarly, {P6} noted that his information seeking was complete, because he remembered that the function he was seeking is in the language's standard library and does not require dependencies. {P1} was the only participant who discussed remembering which sources were accessible and which were not. He shared that, because one source is inaccessible, it *"makes me have to remember that in order to get the best experience, I need to open up [the accessible alternative source]."*

From the perspective of information retrieval, memory could be considered a highly accessible information source, as it does not require vision or screen readers to use. At a deeper level, however, both {P1} and {P9} spoke to situations where screen readers interfered with memory's usability. {P9} often used her vision while programming even though it strained her eyes, because using a screen reader is *"distracting more than anything...it takes longer to get into the zone with my screen reader talking to me. So, I sometimes just forgo it."* Although only explicitly indicated by {P1}, it should be noted that memory was also used to transfer information, gathered from the sources discussed above, into code within the IDE. So, importantly, screen readers interfering with memory introduced cognitive demands that complicated the use of gathered information. Reflecting on this difficulty, {P1} shared:

> *"With [one screen reader], when you land somewhere [in the IDE] you have to read the line by pressing a key. With [another] when [it's] done landing and it settles down, it reads the line for you. Which is very convenient, in my opinion, because...it's just one less thing you have to remember...It kind of sounds lazy. But, if your brain is trying to get wrapped around something you just got done learning and you come back [to the IDE], then it's just one more thing to have to remember."*

So, despite the inherent accessibility of retrieving information from memory, accessibility issues in other information sources can limit the viability of using memory as an information source, in this way.

4.3 Information Source Selection

When reflecting upon their choice of strictly informational sources, participants often provided clear justification for their decisions. Specifically, strictly informational sources were used and evaluated based on the technical depth of information provided and the prior knowledge required to use them. So, unsurprisingly, the most common factor identified as impacting participants' selection was familiarity with a given topic (noted by all but {P10}). When searching for information about an unfamiliar topic, participants typically described beginning their information seeking in sources which required little to no prior knowledge to use, like blogs and tutorials {P1, P2, P8, P11} and textbooks {P2, P3, P4, P9, P11, P12}. Conversely, participants typically described seeking low-level details in official documentation only after consulting other sources {P1, P2, P4, P5, P6, P8, P9, P11}. Forums were typically consulted before {P2, P4, P5, P6, P9} and in conjunction with {P2} official documentation. Notably, where textbooks often served as a starting point for seeking information about unfamiliar topics, because they also provided low-level technical details {P2, P4, P12} they were sometimes used again after consulting official documentation. {P2} shared a particularly illustrative example of this pattern of beginning information seeking in sources which provided general information and required little prior knowledge, and moving toward increasingly detailed sources which required more prior knowledge:

> *"When I found out [two years ago] that we were moving away from the [previous] platform and we were trying to implement the [current platform], the very first step I did–because [the current platform] was new to me– I downloaded a [text]book, which was talking about how to implement [the current platform]...[Now,] if I'm looking for something related to [the current platform], most of the time, Stack Overflow gives me good information...And if I have found some information that talks about [a similar problem] by not [my exact problem], and I know that information is not there, I need to go to [the official documentation]... And, if the same information is presented in both–on the [official documentation] as well as on some Stack overflow discussion...then, I'm 100% sure that is the right information… Occasionally, I do use [the textbook], especially if the information is not on the [official documentation], then the book talks about it in a little more details… I have found that approach very helpful."*

In contrast, we found no strong patterns in the selection of sources with alternative functions across participants. In the cases of Web Search and Memory, it is likely that these decisions happen without conscious reflection. However, in the decision to use Other People as information sources, it is likely we saw no clear patterns across participants because 1) Other People were typically described as providing information that could not be provided by another source, meaning there is no decision to make, and 2) participants' perceptions of the accessibility and usability of Other People as information sources were highly variable, depending on the social connections between them.

4.4 Accessibility in Information Seeking

Though each of the discussed information sources suffered from accessibility issues, (in)accessibility of individual sources was not often discussed as a factor in source selection decisions. But, that is not to suggest that (in)accessibility played no role in information seeking. Rather, participants typically described the (in)accessibility of sources as derailing information seeking in progress {P1, P2, P3, P5, P9, P10, P11, P12}. That is, because the (in)accessibility of a given information source was only identifiable at the time of its use, serving as a *"show stopper,"* in the words of {P2}, it may be impossible to factor (in)accessibility into *a priori* decision making. In fact, the three participants {P1, P5, P12} who did indicate that (in)accessibility affected their information source selection decision referred to remembering poor prior experiences with those sources in their reflections.

Notably, participants described several unique strategies for working around accessibility barriers arising during information seeking in development. Five participants {P1, P3, P6, P7, P8} described sharing and receiving information specific to accessibility in development from other DWVI, who they knew personally {P1, P8} or contacted through DWVI mailing lists {P1, P3, P6}. For instance, participants often described migrating to certain IDEs based on recommendations of other DWVI {P1, P3, P6, P8}. But, when asked about his use of a specific tutorial site, {P7} suggested he was *"just testing"* because he heard from other DWVI that it was particularly accessible. So, participants did not typically address factoring accessibility into their source selection decisions. But, interestingly, source selection may still be influenced by accessibility, via community recommendations.

Additionally, five participants {P1, P3, P4, P6, P8} used custom accessibility setups which they developed themselves {P3, P4, P8} or had been developed by personal contacts who are also DWVI {P1, P6, P8}. {P3} and {P4} each indicated using their custom setups to scaffold information seeking and synthesis. {P4} even noted that, *"I don't think that [developing your own accessibility tools] is something that most people have the ability to do for all kinds of reasons. Which is unfortunate [because] it does definitely make me a little bit more flexible."* It is likely participants with custom setups also used them for information seeking, even where not explicitly mentioned. So, in this way, how accessibility impacts information seeking may be deeply personal, as it drives the development of customized workarounds.

So, we did not see strong themes in how participants indicated that (in)accessibility of technical information sources impacted information seeking. But, our findings suggest that this is because (in)accessibility's role in information seeking is nuanced and personal, not necessarily because (in)accessibility plays no role.

5 DISCUSSION

5.1 Information Behaviors of Developers with Visual Impairments

Broadly speaking, we found that the information sources our participants used were highly similar to those used by DWNSVI, as

documented in prior literature. For example, that developers seek technical information in online tutorials [17, 31, 51], in forums [31, 51], in official programming language specification and API documentation [47], from other people [47, 49], through web searches [30] and in their own memory [17, 49], have each been documented in previous studies of DWNSVI. Likewise, we found similarities in the types of accessibility issues identified in technical information sources and those which are common in nontechnical sources. For instance, visual elements lacking proper textual descriptions, inappropriately nested or missing heading tags, and links whose purpose cannot be determined from context are each issues of noncompliance with established Web Content Accessibility Guidelines (WCAG) [43].

We note that this study is exploratory in nature, and future research should be conducted to confirm these similarities. Additionally, since the technical information sources discussed were noncompliant with WCAG, it is possible that higher-level issues were not discussed by, or were not apparent to, our participants in light of more fundamental issues. This may have obscured our ability to identify differences between technical and nontechnical sources. If WCAG compliance increases substantially in the future, there may be opportunities for research which revisits this question and explores considerations of accessible digital content which are particular to technical information sources.

Our findings differed most substantially from previous research in the domain of information source selection. Established understandings in Information Science suggest that engineering professionals use the information sources which are easiest for them to access, disregarding all other factors [27]. So, we expected that the (in)accessibility of individual information sources would be a significant factor in participants' source selection decisions. Instead, we found that the (in)accessibility of information sources was only known to participants retrospectively. So, many participants {P1, P2, P3, P5, P9, P10, P12} indicated that accessibility issues in a single source might *interrupt* their use of that source, prompting a search for a new source. But, they could not necessarily factor accessibility into their source selection decisions, unless they remembered which previously used sources were (in)accessible, like {P1, P5, P12}.

Vigo et al. [78] tested an approach to annotating hyperlinks with metadata reflecting the accessibility of the sites behind those links, to provide such information in advance. Interestingly, the authors did not find agreement in their sample of everyday web users with visual impairments about the utility of this tool. It is not clear whether this approach would be more useful for tech-savvy users, like DWVI. But, given that ease of access has been shown to be a significant factor in information source selection for engineering professionals [27], it is possible that the availability of such a tool would have changed our findings. There is room for future work testing this hypothesis and the utility of providing predictive accessibility metrics in a professional development environment.

5.2 Accessibility in Development as an Information Seeking Process

At a high level, the identified similarities between DWVIs' information behaviors and those previously documented in studies of DWNSVI could suggest that *being a software developer* plays a stronger role in information seeking than accessibility and visual abilities. But, that does not mean accessibility plays no role in understanding information seeking by DWVI. Here, we found that examining development as an information seeking process emphasized the importance of *when* and *why* accessibility issues arise, and *how* they are overcome, which indicate the influence of contextual and social factors that make some aspects of accessible development more critical and complex to address than others.

For example, because expertise was a driving factor in source selection, the information seeking process described by our participants more closely resembled a sequential and ongoing learning process, than a discrete act of probing a single information source. Our participants typically indicated beginning in resources that provided high-level general information which required little prior knowledge to use, like blogs and tutorials, and moved toward lower-level technical resources, like official documentation, only after gaining prior knowledge. In many ways, this finding is intuitive—sources which require little prior knowledge will be used when developers are facing unfamiliar problems.

However, within this context, understanding *when* accessibility issues arise in information seeking suggests issues in some sources may be more detrimental than others. In particular, we found blogs and tutorials were the sources most frequently noted to have accessibility issues {P1, P2, P5, P6, P7, P8, P9, P10, P12}, because they often contained visual media. Since they are instructional resources, it is unsurprising that blogs and tutorials made heavy use of visual information, which can be especially illustrative for novices. At the same time, because tutorials are directed at new learners, their accessibility issues likely arise in a moment when DWVI are already attempting to make sense of an unfamiliar problem. Additionally, information provided by blogs and tutorials was used to scaffold subsequent information seeking. So, accessibility issues in these sources may create gaps in knowledge which impact DWVIs' abilities to make effective use of the more technical information sources employed later, like official documentation. For these reasons, our findings suggest increasing the accessibility of instructional content, like blogs and tutorials, is especially important. Abdolrahmani et al. [4] previously argued that some technological problems are too large to address fully, such that accessible technology designers may have to prioritize the most critical issues. In this spirit, we find that universal increases in web accessibility are important for DWVI. But, focused investigation into increasing accessibility in online instructional information sources, like blogs and tutorials, may be especially critical for supporting accessible development.

Similarly, understanding *why* accessibility issues arise in information seeking highlighted the use of other people as information sources as a particularly complex source of accessibility issues in development. Mealin and Murphy-Hill [56] found previously that DWVI rely on sighted colleagues for obtaining visual information and that doing so causes embarrassment. But, our findings suggest that whether employing the vision of sighted colleagues as a workaround was perceived by DWVI as an embarrassing act of reliance or a typical team collaboration depended upon the knowledge and empathy of their sighted teammates. Consider, for instance, the difference between {P2}'s suggestion that asking for help does not make him *"feel odd, at all,"* and {P12} indicating that she asks for assistance from *"mainly who I'm close with"* because whether others

are helpful *"depends on how your relationship is with them."* At the same time, interacting with sighted coworkers may be a necessity in information seeking, as other people often served to provide information that may not be reasonably replicated by technological solutions, like distributing labor {P2}, engaging in collaborative problem solving {P3}, and offering domain-specific knowledge {P1, P2, P3, P6, P7, P8, P9}.

These findings suggest that, even in a highly accessible technical environment, professional DWVI may never truly be able to operate fully independently for the same reason DWNSVI may not be able to operate fully independently—the collaborative nature of their job. It is unlikely that increasing the technical accessibility of development tools and information sources can sidestep such socially produced accessibility issues. How social factors impact accessibility in environments shared by people with and without visual impairments has been explored in homes [18, 73, 74], schools [23, 57, 58] and office spaces [19]. But, collaborations between people with and without visual impairments in computer programming have primarily addressed teams of children in an educational capacity [42, 76]. Our findings indicate that interpersonal dynamics in professional development teams influence the overall accessibility of development for DWVI, and there is an opportunity for further investigation in this space.

Likewise, observed similarities in information behaviors could suggest previous approaches to easing information seeking and synthesis for DWNSVI [16, 37–39, 52, 59, 61, 75] may be adequate for supporting DWVI, if their interfaces are accessibly implemented. But, understanding *how* our participants overcame accessibility issues suggests that adopting the strategies that were previously successful for supporting DWNSVI may not be the only—or even best—approach to supporting DWVI. Specifically, we found that many participants had created custom accessibility setups for their development activities {P1, P3, P4, P6, P8}.

Creating accessible technologies *on behalf of* people with disabilities is often a necessity in HCI, because of differences in the technical capabilities of researchers and their anticipated end users. In the domain of accessible software development, however, many target users have the deep technical expertise required to create their own digital tools. {P4}, in particular, noted that designing custom accessibility tools is advantageous, *"because it does definitely make [him] a little bit more flexible."* Flexibility and customizability are especially important in designing for accessibility, because users' preferences and (dis)abilities are highly diverse. Even within our sample, which included only screen reader users, our interviewees had varying levels of visual acuity, which impacted their information behaviors. Consider for instance {P9} who used her vision when in the presence of others because screen readers distracted her, even though she preferred not to strain her eyes. Because of this diversity, accessible making [24, 40] and consumer-grade 3D-printing technologies [20, 21, 35] have been explored as routes toward creating personalized assistive devices in the physical world. Similarly, our findings suggest that, given the population of interest, accessibility in information seeking—and potentially other development activities—may be better supported by tools which scaffold DWVI in creating personalized software tools, than by imposing top-down, generalized design solutions.

So, although we did not observe strong differences in information behaviors between DWVI and those previously documented of DWNSVI, our findings suggest accessibility may impact DWVIs' software development experience, even where no behavioral differences are observable. As a collective, our findings suggest that, even in the highly technical domain of software development, understanding accessibility requires attending to contextual and social factors—outside of the IDE.

6 CONCLUSION

We conducted observations and interviews with twelve DWVI in order to identify the information sources they use in development, the common accessibility issues in these sources, and their strategies for selecting an information source. We compared our findings against previous studies of DWNSVIs' information behaviors and recommendations for designing nontechnical information sources. We found that, despite many superficial similarities, examining development as an information seeking process highlighted the role of contextual and social factors in determining the overall accessibility of the development process.

ACKNOWLEDGMENTS

The authors thank Andrew Macvean and Emerson Murphy-Hill for feedback on earlier drafts of this work and our participants for sharing their valuable insights into technical information seeking.

REFERENCES

[1] [n.d.]. 2017 Disability Statistics Annual Report. ([n. d.]), 48.
[2] [n.d.]. Coding Skills in the Job Market. https://www.burning-glass.com/research-project/coding-skills/
[3] [n.d.]. Software Developers : Occupational Outlook Handbook: : U.S. Bureau of Labor Statistics. https://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm#tab-6
[4] Ali Abdolrahmani, William Easley, Michele Williams, Stacy Branham, and Amy Hurst. 2017. Embracing Errors: Examining How Context of Use Impacts Blind Individuals' Acceptance of Navigation Aid Errors. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 4158–4169. https://doi.org/10.1145/3025453.3025528
[5] Ali Abdolrahmani and Ravi Kuber. 2016. Should I Trust It When I Cannot See It? Credibility Assessment for Blind Web Users. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '16)*. Association for Computing Machinery, New York, NY, USA, 191–199. https://doi.org/10.1145/2982142.2982173
[6] Khaled Albusays and Stephanie Ludi. 2016. Eliciting programming challenges faced by developers with visual impairments: exploratory study. In *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '16)*. Association for Computing Machinery, New York, NY, USA, 82–85. https://doi.org/10.1145/2897586.2897616
[7] Khaled Albusays, Stephanie Ludi, and Matt Huenerfauth. 2017. Interviews and Observation of Blind Software Developers at Work to Understand Code Navigation Challenges. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '17)*. Association for Computing Machinery, New York, NY, USA, 91–100. https://doi.org/10.1145/3132525.3132550
[8] Claire J Anderson, Myron Glassman, R Bruce McAfee, and Thomas Pinelli. 2001. An investigation of factors affecting how engineers and scientists seek information. *Journal of Engineering and Technology Management* 18, 2 (2001), 131–155.
[9] Ameer Armaly, Paige Rodeghero, and Collin McMillan. 2018. A Comparison of Program Comprehension Strategies by Blind and Sighted Programmers. *IEEE Transactions on Software Engineering* 44, 8 (Aug. 2018), 712–724. https://doi.org/10.1109/TSE.2017.2729548 Conference Name: IEEE Transactions on Software Engineering.
[10] Catherine M. Baker, Lauren R. Milne, and Richard E. Ladner. 2015. StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3043–3052. https://doi.org/10.1145/2702123.2702589

[11] Suzanne P. Balik, Sean P. Mealin, Matthias F. Stallmann, and Robert D. Rodman. 2013. GSK: universally accessible graph sketching. In *Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13)*. Association for Computing Machinery, New York, NY, USA, 221–226. https://doi.org/10.1145/2445196.2445266

[12] Suzanne P. Balik, Sean P. Mealin, Matthias F. Stallmann, Robert D. Rodman, Michelle L. Glatz, and Veronica J. Sigler. 2014. Including blind people in computing through access to graphs. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility (ASSETS '14)*. Association for Computing Machinery, New York, NY, USA, 91–98. https://doi.org/10.1145/2661334.2661364

[13] Jeffrey P. Bigham, 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 *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility (Assets '07)*. Association for Computing Machinery, New York, NY, USA, 51–58. https://doi.org/10.1145/1296843.1296854

[14] Jeffrey P. Bigham, Irene Lin, and Saiph Savage. 2017. The Effects of "Not Knowing What You Don't Know" on Web Accessibility for Blind Web Users. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '17)*. Association for Computing Machinery, New York, NY, USA, 101–109. https://doi.org/10.1145/3132525.3132533

[15] Jeffrey P. Bigham, Craig M. Prince, and Richard E. Ladner. 2008. WebAnywhere: a screen reader on-the-go. In *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility (W4A) (W4A '08)*. Association for Computing Machinery, New York, NY, USA, 73–82. https://doi.org/10.1145/1368044.1368060

[16] Joel Brandt, Mira Dontcheva, Marcos Weskamp, and Scott R. Klemmer. 2010. Example-centric programming: integrating web search into the development environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. Association for Computing Machinery, New York, NY, USA, 513–522. https://doi.org/10.1145/1753326.1753402

[17] Joel Brandt, Philip J. Guo, Joel Lewenstein, Mira Dontcheva, and Scott R. Klemmer. 2009. Two studies of opportunistic programming: interleaving web foraging, learning, and writing code. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. Association for Computing Machinery, New York, NY, USA, 1589–1598. https://doi.org/10.1145/1518701.1518944

[18] Stacy M. Branham and Shaun K. Kane. 2015. Collaborative Accessibility: How Blind and Sighted Companions Co-Create Accessible Home Spaces. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2373–2382. https://doi.org/10.1145/2702123.2702511

[19] Stacy M. Branham and Shaun K. Kane. 2015. The Invisible Work of Accessibility: How Blind Employees Manage Accessibility in Mixed-Ability Workplaces. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility (ASSETS '15)*. Association for Computing Machinery, New York, NY, USA, 163–171. https://doi.org/10.1145/2700648.2809864

[20] Erin Buehler, Stacy Branham, Abdullah Ali, Jeremy J Chang, Megan Kelly Hofmann, Amy Hurst, and Shaun K Kane. 2015. Sharing is caring: Assistive technology designs on thingiverse. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 525–534.

[21] Erin Buehler, Niara Comrie, Megan Hofmann, Samantha McDonald, and Amy Hurst. 2016. Investigating the implications of 3D printing in special education. *ACM Transactions on Accessible Computing (TACCESS)* 8, 3 (2016), 1–28.

[22] Raymond P. L. Buse and Thomas Zimmermann. 2012. Information needs for software development analytics. In *2012 34th International Conference on Software Engineering (ICSE)*. 987–996. https://doi.org/10.1109/ICSE.2012.6227122 ISSN: 1558-1225.

[23] Clare Cullen and Oussama Metatla. 2019. Co-designing Inclusive Multisensory Story Mapping with Children with Mixed Visual Abilities. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children (IDC '19)*. Association for Computing Machinery, New York, NY, USA, 361–373. https://doi.org/10.1145/3311927.3323146

[24] Maitraye Das, Katya Borgos-Rodriguez, and Anne Marie Piper. 2020. Weaving by touch: A case analysis of accessible making. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–15.

[25] Ph D. Edward C. Bell and Natalia M. Mino. 2015. Employment Outcomes for Blind and Visually Impaired Adults. (2015). https://www.nfb.org/images/nfb/publications/jbir/jbir15/jbir050202.html

[26] Olutayo Falase, Alexa F. Siu, and Sean Follmer. 2019. Tactile Code Skimmer: A Tool to Help Blind Programmers Feel the Structure of Code. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '19)*. Association for Computing Machinery, New York, NY, USA, 536–538. https://doi.org/10.1145/3308561.3354616

[27] Raya Fidel and Maurice Green. 2004. The many faces of accessibility: engineers' perception of information sources. *Information Processing & Management* 40, 3 (May 2004), 563–581. https://doi.org/10.1016/S0306-4573(03)00003-7

[28] Shelagh Fisher and Tony Oulton. 1999. The critical incident technique in library and information management research. *Education for information* 17, 2 (1999), 113–125.

[29] John C Flanagan. 1954. The critical incident technique. *Psychological bulletin* 51, 4 (1954), 327.

[30] Scott D. Fleming, Chris Scaffidi, David Piorkowski, Margaret Burnett, Rachel Bellamy, Joseph Lawrance, and Irwin Kwan. 2013. An Information Foraging Theory Perspective on Tools for Debugging, Refactoring, and Reuse Tasks. *ACM Transactions on Software Engineering and Methodology* 22, 2 (March 2013), 14:1–14:41. https://doi.org/10.1145/2430545.2430551

[31] Luanne Freund. 2015. Contextualizing the information-seeking behavior of software engineers. *Journal of the Association for Information Science and Technology* 66, 8 (2015), 1594–1605. https://doi.org/10.1002/asi.23278 _eprint: https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.23278.

[32] Nicholas A. Giudice, Hari Prasath Palani, Eric Brenner, and Kevin M. Kramer. 2012. Learning non-visual graphical information using a touch-based vibro-audio interface. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '12)*. Association for Computing Machinery, New York, NY, USA, 103–110. https://doi.org/10.1145/2384916.2384935

[33] Filipe Del Nero Grillo and Renata Pontin de Mattos Fortes. 2014. Tests with blind programmers using awmo: An accessible web modeling tool. In *International Conference on Universal Access in Human-Computer Interaction*. Springer, 104–113.

[34] Filipe Del Nero Grillo, Renata Pontin de Mattos Fortes, and Daniel Lucrédio. [n.d.]. Towards collaboration between sighted and visually impaired developers in the context of Model-Driven Engineering.

[35] Anhong Guo, Jeeeun Kim, Xiang'Anthony' Chen, Tom Yeh, Scott E Hudson, Jennifer Mankoff, and Jeffrey P Bigham. 2017. Facade: Auto-generating tactile interfaces to appliances. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 5826–5838.

[36] Alex Hadwen-Bennett, Sue Sentance, and Cecily Morrison. 2018. Making Programming Accessible to Learners with Visual Impairments: A Literature Review. *International Journal of Computer Science Education in Schools* 2, 2 (May 2018), 3–13. https://doi.org/10.21585/ijcses.v2i2.25 Number: 2.

[37] Björn Hartmann, Leslie Wu, Kevin Collins, and Scott R. Klemmer. 2007. Programming by a sample: rapidly creating web applications with d.mix. In *Proceedings of the 20th annual ACM symposium on User interface software and technology (UIST '07)*. Association for Computing Machinery, New York, NY, USA, 241–250. https://doi.org/10.1145/1294211.1294254

[38] Raphael Hoffmann, James Fogarty, and Daniel S. Weld. 2007. Assieme: finding and leveraging implicit references in a web search interface for programmers. In *Proceedings of the 20th annual ACM symposium on User interface software and technology (UIST '07)*. Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/1294211.1294216

[39] Reid Holmes, Robert J. Walker, and Gail C. Murphy. 2005. Strathcona example recommendation tool. In *Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering (ESEC/FSE-13)*. Association for Computing Machinery, New York, NY, USA, 237–240. https://doi.org/10.1145/1081706.1081744

[40] Amy Hurst and Jasmine Tobias. 2011. Empowering individuals with do-it-yourself assistive technology. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*. 11–18.

[41] Joe Hutchinson and Oussama Metatla. 2018. An Initial Investigation into Non-visual Code Structure Overview Through Speech, Non-speech and Spearcons. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3170427.3188696

[42] Gesu India, Geetha Ramakrishna, Joyojeet Pal, and Manohar Swaminathan. 2020. Conceptual Learning through Accessible Play: Project Torino and Computational Thinking for Blind Children in India. In *Proceedings of the 2020 International Conference on Information and Communication Technologies and Development (ICTD2020)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3392561.3394634

[43] W3C Web Accessibility Initiative (WAI). [n.d.]. Web Content Accessibility Guidelines (WCAG) Overview. https://www.w3.org/WAI/standards-guidelines/wcag/

[44] Chandrika Jayant, Matt Renzelmann, Dana Wen, Satria Krisnandi, Richard Ladner, and Dan Comden. 2007. Automated tactile graphics translation: in the field. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility (Assets '07)*. Association for Computing Machinery, New York, NY, USA, 75–82. https://doi.org/10.1145/1296843.1296858

[45] Andrea R. Kennel. 1996. Audiograf: a diagram-reader for the blind. In *Proceedings of the second annual ACM conference on Assistive technologies (Assets '96)*. Association for Computing Machinery, New York, NY, USA, 51–56. https://doi.org/10.1145/228347.228357

[46] Alasdair King, Paul Blenkhorn, David Crombie, Sijo Dijkstra, Gareth Evans, and John Wood. 2004. Presenting UML Software Engineering Diagrams to Blind People. In *Computers Helping People with Special Needs (Lecture Notes in Computer Science)*, Klaus Miesenberger, Joachim Klaus, Wolfgang L. Zagler, and Dominique Burger (Eds.). Springer, Berlin, Heidelberg, 522–529. https://doi.org/10.1007/978-3-540-27817-7_76

[47] Amy J. Ko, Robert DeLine, and Gina Venolia. 2007. Information Needs in Collocated Software Development Teams. In *29th International Conference on Software Engineering (ICSE'07)*. 344–353. https://doi.org/10.1109/ICSE.2007.45 ISSN: 1558-1225.

[48] Richard E. Ladner, Melody Y. Ivory, Rajesh Rao, Sheryl Burgstahler, Dan Comden, Sangyun Hahn, Matthew Renzelmann, Satria Krisnandi, Mahalakshmi Ramasamy, Beverly Slabosky, Andrew Martin, Amelia Lacenski, Stuart Olsen, and Dmitri Groce. 2005. Automating tactile graphics translation. In *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility (Assets '05)*. Association for Computing Machinery, New York, NY, USA, 150–157. https://doi.org/10.1145/1090785.1090814

[49] Thomas D. LaToza, Gina Venolia, and Robert DeLine. 2006. Maintaining mental models: a study of developer work habits. In *Proceedings of the 28th international conference on Software engineering (ICSE '06)*. Association for Computing Machinery, New York, NY, USA, 492–501. https://doi.org/10.1145/1134285.1134355

[50] Gloria J Leckie, Karen E Pettigrew, and Christian Sylvain. 1996. Modeling the information seeking of professionals: A general model derived from research on engineers, health care professionals, and lawyers. *The Library Quarterly* 66, 2 (1996), 161–193.

[51] Hongwei Li, Zhenchang Xing, Xin Peng, and Wenyun Zhao. 2013. What help do developers seek, when and how?. In *2013 20th Working Conference on Reverse Engineering (WCRE)*. 142–151. https://doi.org/10.1109/WCRE.2013.6671289 ISSN: 2375-5369.

[52] Michael Xieyang Liu, Jane Hsieh, Nathan Hahn, Angelina Zhou, Emily Deng, Shaun Burley, Cynthia Taylor, Aniket Kittur, and Brad A. Myers. 2019. Unakite: Scaffolding Developers' Decision-Making Using the Web. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 67–80. https://doi.org/10.1145/3332165.3347908

[53] Stephanie Ludi, Jamie Simpson, and Wil Merchant. 2016. Exploration of the Use of Auditory Cues in Code Comprehension and Navigation for Individuals with Visual Impairments in a Visual Programming Environment. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '16)*. Association for Computing Machinery, New York, NY, USA, 279–280. https://doi.org/10.1145/2982142.2982206

[54] Darren Lunn, Simon Harper, and Sean Bechhofer. 2011. Identifying Behavioral Strategies of Visually Impaired Users to Improve Access to Web Content. *ACM Transactions on Accessible Computing* 3, 4 (April 2011), 13:1–13:35. https://doi.org/10.1145/1952388.1952390

[55] Erin Martz and Yonghong Jade Xu. 2008. Person-related and service-related factors predicting employment of individuals with disabilities. *Journal of Vocational Rehabilitation* 28, 2 (Jan. 2008), 97–104. https://content.iospress.com/articles/journal-of-vocational-rehabilitation/jvr00408 Publisher: IOS Press.

[56] Sean Mealin and Emerson Murphy-Hill. 2012. An exploratory study of blind software developers. In *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 71–74. https://doi.org/10.1109/VLHCC.2012.6344485 ISSN: 1943-6106.

[57] Oussama Metatla, Sandra Bardot, Clare Cullen, Marcos Serrano, and Christophe Jouffrais. 2020. Robots for Inclusive Play: Co-designing an Educational Game With Visually Impaired and sighted Children. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[58] Oussama Metatla and Clare Cullen. 2018. “Bursting the Assistance Bubble”: Designing Inclusive Technology with Children with Mixed Visual Abilities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173920

[59] Stephen Oney and Joel Brandt. 2012. Codelets: linking interactive documentation and example code in the editor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2697–2706. https://doi.org/10.1145/2207676.2208664

[60] Luca Pascarella, Davide Spadini, Fabio Palomba, Magiel Bruntink, and Alberto Bacchelli. 2018. Information Needs in Contemporary Code Review. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (Nov. 2018), 135:1–135:27. https://doi.org/10.1145/3274404

[61] Luca Ponzanelli, Alberto Bacchelli, and Michele Lanza. 2013. Seahawk: Stack Overflow in the IDE. In *2013 35th International Conference on Software Engineering (ICSE)*. 1295–1298. https://doi.org/10.1109/ICSE.2013.6606701 ISSN: 1558-1225.

[62] Venkatesh Potluri, Priyan Vaithilingam, Suresh Iyengar, Y. Vidya, Manohar Swaminathan, and Gopal Srinivasa. 2018. CodeTalk: Improving Programming Environment Accessibility for Visually Impaired Developers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3173574.3174192

[63] Mark A Robinson. 2010. An empirical analysis of engineers' information behaviors. *Journal of the American Society for information Science and technology* 61, 4 (2010), 640–658.

[64] Caitlin Sadowski, Kathryn T. Stolee, and Sebastian Elbaum. 2015. How developers search for code: a case study. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2015)*. Association for Computing Machinery, New York, NY, USA, 191–201. https://doi.org/10.1145/2786805.2786855

[65] Nuzhah Gooda Sahib, Anastasios Tombros, and Tony Stockman. 2012. A comparative analysis of the information-seeking behavior of visually impaired and sighted searchers. *Journal of the American Society for Information Science and Technology* 63, 2 (2012), 377–391. https://doi.org/10.1002/asi.21696 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.21696.

[66] Daisuke Sato, Hironobu Takagi, Masatomo Kobayashi, Shinya Kawanaka, and Chieko Asakawa. 2010. Exploratory Analysis of Collaborative Web Accessibility Improvement. *ACM Transactions on Accessible Computing* 3, 2 (Nov. 2010), 5:1–5:30. https://doi.org/10.1145/1857920.1857922

[67] Emmanuel Schanzer, Sina Bahram, and Shriram Krishnamurthi. 2019. Accessible AST-Based Programming for Visually-Impaired Programmers. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 773–779. https://doi.org/10.1145/3287324.3287499

[68] Ann C. Smith, Justin S. Cook, Joan M. Francioni, Asif Hossain, Mohd Anwar, and M. Fayezur Rahman. 2003. Nonvisual tool for navigating hierarchical structures. *ACM SIGACCESS Accessibility and Computing* 77-78 (Sept. 2003), 133–139. https://doi.org/10.1145/1029014.1028654

[69] Andreas Stefik, Roger Alexander, Robert Patterson, and Jonathan Brown. 2007. WAD: A Feasibility study using the Wicked Audio Debugger. In *15th IEEE International Conference on Program Comprehension (ICPC '07)*. 69–80. https://doi.org/10.1109/ICPC.2007.42 ISSN: 1092-8138.

[70] Andreas Stefik, Andrew Haywood, Shahzada Mansoor, Brock Dunda, and Daniel Garcia. 2009. SODBeans. In *2009 IEEE 17th International Conference on Program Comprehension*. 293–294. https://doi.org/10.1109/ICPC.2009.5090064 ISSN: 1092-8138.

[71] Andreas Stefik, Christopher Hundhausen, and Robert Patterson. 2011. An empirical investigation into the design of auditory cues to enhance computer program comprehension. *International Journal of Human-Computer Studies* 69, 12 (Dec. 2011), 820–838. https://doi.org/10.1016/j.ijhcs.2011.07.002

[72] Tony Stockman and Oussama Metatla. 2008. The influence of screen-readers on web cognition. In *Proceeding of Accessible design in the digital world conference (ADDW 2008), York, UK*.

[73] Kevin M. Storer and Stacy M. Branham. 2019. "That's the Way Sighted People Do It": What Blind Parents Can Teach Technology Designers About Co-Reading with Children. In *Proceedings of the 2019 on Designing Interactive Systems Conference (DIS '19)*. Association for Computing Machinery, New York, NY, USA, 385–398. https://doi.org/10.1145/3322276.3322374

[74] Kevin M. Storer, Tejinder K. Judge, and Stacy M. Branham. 2020. "All in the Same Boat": Tradeoffs of Voice Assistant Ownership for Mixed-Visual-Ability Families. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376225

[75] J. Stylos and B.A. Myers. 2006. Mica: A Web-Search Tool for Finding API Components and Examples. In *Visual Languages and Human-Centric Computing (VL/HCC'06)*. 195–202. https://doi.org/10.1109/VLHCC.2006.32 ISSN: 1943-6106.

[76] Anja Thieme, Cecily Morrison, Nicolas Villar, Martin Grayson, and Siân Lindley. 2017. Enabling Collaboration in Learning Computer Programing Inclusive of Children with Vision Impairments. In *Proceedings of the 2017 Conference on Designing Interactive Systems (DIS '17)*. Association for Computing Machinery, New York, NY, USA, 739–752. https://doi.org/10.1145/3064663.3064689

[77] Christine Urquhart, Ann Light, Rhian Thomas, Anne Barker, Alison Yeoman, Jan Cooper, Chris Armstrong, Roger Fenton, Ray Lonsdale, and Siân Spink. 2003. Critical incident technique and explicitation interviewing in studies of information behavior. *Library & Information Science Research* 25, 1 (2003), 63–88.

[78] Markel Vigo, Barbara Leporini, and Fabio Paternò. 2009. Enriching web information scent for blind users. In *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility (Assets '09)*. Association for Computing Machinery, New York, NY, USA, 123–130. https://doi.org/10.1145/1639642.1639665

[79] B. N. Walker and L. M. Mauney. 2010. Universal Design of Auditory Graphs: A Comparison of Sonification Mappings for Visually Impaired and Sighted Listeners. *ACM Transactions on Accessible Computing* 2, 3 (March 2010), 12:1–12:16. https://doi.org/10.1145/1714458.1714459

[80] Doug Wightman, Zi Ye, Joel Brandt, and Roel Vertegaal. 2012. SnipMatch: using source code context to enhance snippet retrieval and parameterization. In *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12)*. Association for Computing Machinery, New York, NY, USA, 219–228. https://doi.org/10.1145/2380116.2380145

[81] Dongsong Zhang, Lina Zhou, Judith O. Uchidiuno, and Isil Y. Kilic. 2017. Personalized Assistive Web for Improving Mobile Web Browsing and Accessibility for Visually Impaired Users. *ACM Transactions on Accessible Computing* 10, 2 (April 2017), 6:1–6:22. https://doi.org/10.1145/3053733