

# Understanding Peer-to-Peer Instructional Support in an Online Community for Blind Audio Producers

Abir Saha  
Northwestern University  
Evanston, IL, USA  
abir@u.northwestern.edu

Darren Gergle  
Northwestern University  
Evanston, IL, USA  
dgergle@northwestern.edu

Anne Marie Piper  
University of California, Irvine  
Irvine, CA, USA  
ampiper@uci.edu

## ABSTRACT

Exchange of peer-to-peer support in online Q&A communities plays an instrumental role in helping people learn and use complex software tools. While prior work has documented how disabled people support each other in finding accessible practices and workarounds in different contexts, research on understanding their participation in dedicated online Q&A communities has been limited. Through the analysis of 180 conversation threads consisting of 1140 posts in an online text-based Q&A community of blind and low-vision audio producers, we reveal various strategies members in this community use to formulate their queries and provide effective solutions regarding screen reader based navigation of complex graphical user interfaces. We reflect upon our findings to discuss the complexities blind and low-vision software users face in developing a shared understanding during collaborative troubleshooting through textual conversations and reimagine how online Q&A platforms could enhance peer-to-peer instructional support among screen reader users.

## CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in accessibility**; **Empirical studies in collaborative and social computing**.

## KEYWORDS

Accessibility; audio production; blind; low vision; screen reader; online forum; Q&A

### ACM Reference Format:

Abir Saha, Darren Gergle, and Anne Marie Piper. 2023. Understanding Peer-to-Peer Instructional Support in an Online Community for Blind Audio Producers. In *The 25th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '23)*, October 22–25, 2023, New York, NY, USA. ACM, New York, NY, USA, 15 pages. <https://doi.org/10.1145/3597638.3608399>

## 1 INTRODUCTION

Online Question and Answer (Q&A) communities have been instrumental in helping people learn and perform skilled work across various domains from computer programming [37, 45] to graphic

design [19] to music production [21]. Given the increasing use of software and hardware tools in these domains, online Q&A platforms play an important role in facilitating the ways in which people learn to use and understand complex tools required for various forms of skilled work. Prior research has studied how members of Q&A platforms such as *Yahoo! Answers*, *Quora*, *StackExchange*, and software-specific mailing lists help each other in learning to use and troubleshoot complex software required for computer-supported skilled work [2, 24, 25, 42, 65, 67, 68]. Engaging in these platforms has benefits beyond learning, as the peer-to-peer exchange of support in these spaces can foster a sense of community through constructive dialogue [51, 67]. Although limited work analyzes the practices of people with disabilities in dedicated Q&A spaces (see [29, 30, 61] for exceptions), how disabled people often help other in securing access and identifying workarounds to inaccessible environments and technologies has been well-established in the literature [3, 16, 26, 34, 36]. By analyzing online Q&A communities that are created for and by disabled people, we can better understand the kinds of support people are seeking, including help with pervasive accessibility issues, as well as how experts support others in learning to use complex software required for skilled work.

The present study focuses on how blind and low-vision screen reader users engage in online Q&A as part of learning the tools and practices of audio production. Audio production is a particularly compelling domain of inquiry, as prior work has documented the growing interest in the field—both professionally and as a hobby—among blind and low-vision people as well as pervasive accessibility issues with audio production software and hardware [39, 50, 58]. Specifically, audio production software—often called Digital Audio Workstations (DAWs)—contain complicated Graphical User Interfaces (GUIs) that are geared towards sighted users, leading to accessibility challenges, workflow disruptions, and inefficiencies for screen reader users [50, 58]. Given these difficulties, experienced blind audio producers often create accessible tutorials that can help beginners and seasoned professionals alike develop further competency and expertise in using complex audio production interfaces [57]. While peer-to-peer learning in online Q&A communities involving sighted people has been studied extensively (e.g., [2, 45, 55, 60, 66–68]), we know less about how blind and low-vision people use similar spaces to exchange instructional support and develop technical skills and expertise. To help bridge this gap in the literature, we analyze one specific online text-based Q&A community: a publicly available mailing list in which blind and low-vision screen reader users seek and provide instructional support regarding accessible audio production tools and practices. This community was initiated in 2011 and discusses the use of Logic

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

ASSETS '23, October 22–25, 2023, New York, NY, USA

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0220-4/23/10...\$15.00

<https://doi.org/10.1145/3597638.3608399>

Pro, a popular DAW on macOS with VoiceOver screen reader. Our analysis addresses three primary research questions:

- **RQ1:** *What types of discussions take place in this community?*
- **RQ2:** *For discussions related to seeking help, with which aspects do learners request help and how do they formulate their help requests?*
- **RQ3:** *How do community members support help-seekers in understanding how to use and navigate the DAW with a screen reader?*

We report findings from an analysis of 180 conversation threads consisting of 1140 individual posts from this online text-based Q&A community initiated and managed by blind audio producers. Our analysis reveals four primary types of discussions that occur in this community, including understanding navigation and interaction within the DAW, product reviews and recommendations, promotional and sales activities, and advocacy and sustaining accessible tools and tutorials. Analysis of posts related to navigation and interaction with the DAW reveal how help-seekers formulate their questions and ask for assistance in learning to use audio production software. Finally, we detail different strategies experienced helpers adopt to support help-seekers in addressing challenges of navigating and interacting with complex GUIs and how both parties resolve misalignments that arise not only in help-seekers' understanding of screen reader-based navigation of DAWs but also in the communication between helpers and help-seekers.

Our paper makes three key contributions. First, we provide an empirical overview of the different kinds of discussions supported by an online Q&A community of blind and low-vision audio producers, highlighting topics ranging from peer-to-peer support regarding use of software tools to coordinating on sustaining accessible learning materials and advocating for software accessibility. Second, we contribute a detailed analysis of different strategies for formulating help requests and providing effective solutions regarding screen reader based GUI navigation, illustrating the complexity of collaborative troubleshooting in this context. Finally, we reflect upon the different factors that aid or disrupt the development of shared understanding among screen reader users in troubleshooting contexts and discuss opportunities for reimagining the ways in which online platforms enable peer-to-peer instructional support among screen reader users.

## 2 RELATED WORK

Our work builds upon research on online Q&A communities and online help-seeking and peer support among disabled people.

### 2.1 Online Q&A Communities and Practices

Over the past two decades, there has been an emergence of online Q&A forums, which are online spaces where people with shared interests pose questions about their activities and problems, and receive answers or comments from a broad set of individuals. Prior work has examined participation on Q&A platforms e.g., *Yahoo! Answers* [2], *Quora* [48, 68], *StackExchange* [19, 21, 37, 45], and mailing lists [24, 67]. This work indicates that the success of online Q&A platforms hinges upon users' perceptions about the answers provided, incentives for answering questions, and approach to designing these forums. Harper et al. [25] proposed categorizing the

kinds of questions asked within these spaces as conversational, discursive, or informational. Others identified many different types of questions, including factual (seeking objective data), advice (seeking recommendations), opinion (seeking others' viewpoints), and non-questions (spam) [27, 37, 42]. Prior work has also characterized various types of users in these spaces, including lurkers, help-seekers (askers), and help-givers (responders) [65]. As is the case with peer-production communities like Wikipedia [31], a few highly active users contribute a majority of the content (i.e., questions and answers) posted within these spaces [65]. Nevertheless, online Q&A platforms rely on a combination of intrinsic and extrinsic motivation to participate, including visible reputation markers and connections to one's interests and identities [27, 67].

Researchers have also investigated the help-seeking and help-giving strategies enacted on these online communities [59]. For instance, past work has characterized how the same people engage differently in two separate Q&A venues i.e., a mailing list on R programming language versus the StackExchange network on the same topic [67]. Others highlighted what types of career advice responses are most valued in an online career support community [66], how instructors make use of online Q&A forums for troubleshooting and seeking help regarding online teaching [55], and the differences in argumentation style between experts and newcomers on a programming language related forum [64]. Hudson et al. [28] examined several software help forums to uncover that use of non-standard terminology and lack of visual demonstration and information on current system state in a help-seeking post tend to prompt requests for further clarification from helpers.

Another relevant branch of research involves how people engage in social media question asking (SMQA), where the participants facilitate Q&A activities through social media platforms (e.g., Facebook groups) that are not specifically designed with help-seeking in mind, as opposed to dedicated Q&A forums like *Quora* and *StackExchange*. These studies look at people's reasons for turning to social media for help-seeking and the factors affecting the outcomes of these efforts [41], the goals and strategies of social media help-seeking [44], and help-seekers' expectations when they tag acquaintances in their posts [20]. One distinction is that much of the SMQA literature focuses on social support (e.g., in the context of health [44]) rather than peer learning and instructional support among professionals and hobbyists.

### 2.2 Online Help-Seeking Practices and Peer Support Among People with Disabilities

Prior research within Human-Computer Interaction (HCI), accessibility, and Disability Studies highlights how accessibility is created among groups of people with and without disabilities [4, 10, 15, 16, 34] and the ways in which disabled people are agents in securing access [3, 18, 26, 36]. As part of navigating disability experiences, scholars have begun to analyze the online help-seeking practices among disabled people. Most closely related to our work are studies of blind and low-vision programmers who engage in help-seeking online. For example, blind and low-vision programmers often experience difficulty finding support from others who are knowledgeable about accessibility issues [47], even in online Q&A forums dedicated to programming support [62]. To better understand the

help-seeking behavior of novices, Johnson et al. [29] studied a text-based forum for blind programmers called Program-L. They detail the community norms such as introducing oneself (and disclosing disability), describing one's setup (e.g., IDE and screen reader), and making requests for informational (i.e., knowledge-based) support and practical assistance.

Beyond these few studies on help-seeking related to skilled work, other research has examined how disabled people give and receive peer support in online spaces related to other goals. For example, early work explored how blind people ask visual questions by posting photos and receiving answers from sighted people through crowdsourcing applications (e.g., VizWiz [5]) and social networking sites [9]. Others have examined parenting groups on Facebook to understand co-reading practices of blind parents and their children [61] and blind podcasters' content on interactions with voice-activated personal assistants (e.g., Amazon Echo, Apple Siri) [1]. Similarly, Gotfrid et al. [23] analyzed online forum conversations among disabled knitters to understand accessibility in knitting practices. Still others have examined how disabled people and their care partners (e.g., parents, spouses, adult children) seek and receive social support through online communities, such as an online forum for people with dementia [30] and a video-based platform for parents of children with developmental disabilities [7].

Despite extensive work on Q&A communities in general and online peer support practices among people with disabilities, we know less about how disabled people—particularly blind and low-vision screen reader users—engage in online peer-to-peer support as part of learning to use and troubleshoot accessibility issues with complex software tools.

### 2.3 Accessibility in Audio Production

A growing body of work within HCI and accessible computing has been investigating accessibility in audio content creation for blind and low-vision people. Research in this space has primarily focused on designing novel and multimodal tools using sonification [39], haptic feedback [49, 63], and tangible and tabletop objects [46] to support blind people in audio editing and music composition. To a lesser extent, researchers have also investigated how blind and low vision audio producers and music composers learn and use existing audio production software and music technology [50, 57, 58]. They have detailed the ways blind audio producers pieced create workflows that combine a multitude of semi-accessible tools. These studies also highlighted how seasoned blind audio producers maintained and engaged in several online spaces (e.g., WhatsApp, online forums, mailing lists, etc.) for discussions related to accessible audio production, where they used their hard-earned expertise with accessible music technologies to help other screen reader users learn these tools. Our work extends this research by providing a detailed account of how peer-learning takes place in an online text-based Q&A community of blind audio producers.

## 3 METHOD

### 3.1 Context of Study: An Online Q&A Community for Blind Audio Producers

Prior research (e.g., [58]) revealed the existence of multiple publicly available online communities for blind and low vision audio

producers, each centered around discussing accessible interactions with DAW software (e.g., Logic Pro, REAPER, or Pro Tools). After reviewing these online spaces, we decided to focus our analysis on an online community for blind and low-vision audio producers who use Logic Pro using the VoiceOver screen reader. Members can start a new thread with a subject line and original post, and all subsequent posts made in response to the original post (i.e., replies) will appear as a single thread. This platform is primarily used like a mailing list where sending out a new email and replying to an existing email are equivalent to creating a new conversation thread and replying to that thread respectively. Members communicate almost exclusively using text and sometimes send audio files or external links as attachments – but no images. The community started on January 2011 and is active as of the writing of this paper.

We chose to focus our analysis on this community over others because understanding and interpreting conversations in these Q&A platforms requires technical knowledge of the particular DAW and screen reader. As such, we focused on the online community that discusses a DAW-screen reader pairing (Logic Pro and VoiceOver) with which the research team is most experienced.

### 3.2 Data Collection

We collected data during August 2020. Using the search filter functionality on the mailing list, we manually collected links of all conversation threads posted over a six-month period between February 1, 2020 and July 31, 2020. This resulted in our corpus consisting of 180 conversation threads (respectively 18, 27, 32, 37, 36, and 30 threads in each of the six months). Each thread contained between 1 and 38 posts including the original post and replies (median 4 posts per thread), resulting in a total of 1140 posts in our corpus. For each thread, we collected the subject, date of posting, original post, and subsequent replies on the thread. We decided to limit our analysis to six months of activity because of the complexity of discussion (posts and replies) within each thread, our corpus being comparable in size to what other studies on online communities analyzed [29, 30, 61], and because we began to observe common patterns and themes across our data.

### 3.3 Data Analysis

We analyzed the data in two stages. First, to address RQ1 (i.e., the types of discussion made within this space), the first-author performed an initial open coding on our entire corpus based on the subject and first post of each thread. Initial codes included, for example, “selling musical instruments”, “communicating accessibility issues to DAW developers”, “seeking recommendations for accessible plugins” and more. Although we had a sense of some types of conversations that occurred in this forum, we kept an open mind about other possible discussion topics. The three authors regularly met to discuss these open codes and then refined and reworked the codes into four categories that captured the main topics of discussion: (1) understanding navigation and interaction within audio production software, (2) product reviews and recommendations, (3) promotional and sales activities, and (4) advocating for and sustaining accessible tools and tutorials. Once we developed this coding scheme, the first and third authors independently coded a random

sample of 20% of the conversation threads in our corpus. We calculated inter-rater reliability using Cohen’s Kappa (unweighted) which yielded  $\kappa = 0.93$ , indicating high agreement among coders [33]. We resolved any disagreements through discussion and the first author then coded the rest of the threads. We report frequency of discussion types and examples in Section 4.1.

Next, to answer RQ2 and RQ3, we analyzed discussion threads related to understanding navigation and interaction within audio production software (total 138 threads). We took a bottom-up approach in our data analysis at this stage, where we did not start with preconceived codes or themes but relied on the data to guide our coding process. Following a thematic analysis approach [11], the co-authors first familiarized themselves with the data by looking at conversation threads together to develop a shared understanding. Following this, we performed an initial open coding of the threads, paying particular attention to types of questions asked, challenges described by help-seekers, and strategies used by helpers to communicate possible solutions. We also produced analytic memos for a deeper analysis of selected threads that contained rich interaction between helpers and help-seekers. These memos allowed us to draw connections between initial queries, follow-up clarification requests and suggested solutions, identify potential sources of misalignment between helpers and help-seekers, and develop common themes across our data. While the first author led the coding process, the research team met regularly to discuss the codes and analytic memos, through which we reached researcher agreement on themes. Through iterative comparison between our codes, data, and memos, we reworked these themes into the findings presented in Sections 4.2 and 4.3. While data analysis for RQ1 involved a closed-ended coding scheme and required inter-rater reliability, our approach to RQ2 and RQ3 followed established practices of inductive thematic analysis where we relied on an iterative process of discussion and refinement [38] to identify and report the most salient and cross-cutting themes.

All co-authors of this paper are sighted. The first author who led data coding and preparation of analytic memos has experience with multiple DAWs and screen readers and has worked closely with blind and low vision audio producers for four years.

### 3.4 Ethical Considerations

We obtained approval from our university’s Institutional Review Board to perform analysis of public discussions on online platforms. We critically reflected upon the ethics of conducting research using publicly available data shared by people who did not provide their explicit consent and how data collection and reporting approaches can harm marginalized communities [35, 69]. Following established practices for preserving anonymity and privacy of individuals, we de-identify and paraphrase original text for excerpts reported in the paper [12, 40]. As sighted researchers, we did not join the mailing list which was primarily a space for blind and low-vision audio producers. Hence, we did not collect information about total number of members and their identities and grounded our analysis only on publicly available data. Since we did not know members’ gender identities and preferred pronouns, we use they/them pronouns when describing their activities. Note that the ‘helper’ or ‘help-seeker’ designations used in this paper do not apply universally

to any given community member but on a thread-by-thread basis depending on whether a member is seeking or providing help in that particular thread. That is, a help-seeker in one thread can be the helper in another thread, and vice versa.

## 4 FINDINGS

Our overarching goal is to understand how peer-to-peer instructional support takes place in this online community among blind audio producers. We structure our presentation of findings following our three research questions. First, we briefly describe the types of discussions taking place in this community (RQ1). Then given our analytic interest in instructional support, we turn our attention to a more detailed analysis of the kinds of help-seeking requests made within this community and how members formulated their requests for help (RQ2). Finally, we describe the strategies helpers used to support help-seekers, elaborating on examples of their conversations and interactions with audio production tools to illustrate the complexity of collaborative troubleshooting in this context (RQ3). Given that the posts include audio production and screen reader related terms that general readers may not be familiar with, we provide a glossary of repeatedly-used terms.

### 4.1 What Types of Discussions Take Place in This Community?

Our analysis revealed that blind audio producers in this community participated in four broad categories of conversations: (1) understanding navigation and interaction within audio production tools, (2) product reviews and recommendations, (3) promotional and sales activities, and (4) advocating for and sustaining accessible tools and tutorials. Overall, 178 out of 180 conversation threads in our corpus corresponded to one of these four categories, while the two remaining threads were created by a member for testing purposes and did not have any discussion topic.

*4.1.1 Navigation and Interaction Within Audio Production Tools.* The most prominent activity on this forum (76.7% i.e., 138 out of 180 threads) involved asking questions and providing answers regarding accessible ways to navigate and interact with Logic Pro, which was the primary DAW of interest among community members, as well as other audio production tools. These conversation threads typically received high engagement from members with multiple back-and-forth questions and answers. Our analysis revealed interesting and varying instances of how members attempted to discuss and resolve the complexities they faced associated with screen reader based GUI navigation on the DAW, such as learning how to perform a task, keyboard shortcuts or features not working as expected, certain features disappearing from the interface, and new features appearing after software updates. Given our focus on understanding instructional support, we detail these discussions in Sections 4.2 and 4.3.

*4.1.2 Product Reviews and Recommendations.* Roughly 12% (23 out of 180 threads) of our data were related to reviews and recommendations for audio production tools. A common example of such activity was exchanging reviews and recommendations on how accessible certain software and hardware tools were for blind and low-vision users. One poster said: “*On Pro Tools, many plugins are*

not fully accessible without extended use of scripting. How is the accessibility in Logic for parameters and settings of plugins like <Names>?” Members also sought out feedback regarding new software versions before installing updates on their computer, especially since software updates are known to break or reduce screen reader support in DAWs and other software [58]. One member asked: “Is it safe to update Logic Pro to the latest version? I’ve not updated it for a while. Are there any known VoiceOver bugs?”

**4.1.3 Promotional and Sales Activities.** While the forum was primarily used for asking and answering questions about accessible use of audio production tools, it also served as a space for blind audio producers to advertise their content, services, and equipment to people with shared interests and needs (9 threads or 5% of our data). For example, some members shared music they had created while others advertised their teaching and professional services. One person posted: “Wrote a jazz piece and decided to record using Logic Pro. If interested, here’s the YouTube link.” Another posted, “I’m available for one-on-one training on how to use Logic Pro. Write me off the forum if interested.” Some also posted ads to sell used equipment such as computers, musical instruments, and more.

**4.1.4 Advocating for and Sustaining Accessible Tools and Tutorials.** Our data also included instances of individual and collective efforts to advocate for better screen reader support on audio production tools and facilitate the onboarding of blind and low-vision learners. While only eight discussion threads (about 4%) were in this category, threads like these play an important role in improving accessibility of audio production tools and sustaining accessible resources [57]. Members shared their experience with contacting software developers and the status and outcome of their advocacy efforts. One poster said, “I’ve been in contact with <App> support. They told me the Mac app still isn’t accessible, but they’re planning to include VoiceOver support.”

We also observed threads where members coordinated their strategy for collectively approaching software developers. An expert member posted ideas on joining forces to share blind users’ needs to Apple (developer of Logic Pro). This member also shared a Google Form to crowdsource a list of accessibility issues in Logic and made this list available for public review in a GitHub repository.

*“First, we can encourage as many folks as possible to reach out to Apple and let them know there is a large customer base who care about accessibility for Logic. Second, we should approach Apple as a community with more consistency in our message. If we come up with a collective message from within the community, it might be helpful in addition to individual voices.”*

Beyond advocacy efforts, members initiated and joined collaborative efforts around creating, sharing, and curating accessible audio production tutorials so that others can easily find them. One poster asked another member, “Hey! If you make a tutorial, care if I share it on <Website> and the YouTube channel so its easily accessible from search results?” Because these tutorials are rooted in the experiences of blind audio producers, they are more likely to resonate with the challenges faced by new learners who are just getting started with DAWs using screen readers [57].



Figure 1: Screenshot of a DAW called Logic Pro.

## 4.2 How Do New Learners Seek Help in This Community?

Working on audio production (e.g., editing, mixing, mastering) involves simultaneously paying attention to multiple audio tracks, controlling a number of parameters (e.g., volume levels) and more—all while navigating a complex DAW interface (Figure 1) in combination with third party plugins, digital instruments, and hardware tools. While the use of a DAW’s graphical user interface has a steep learning curve for sighted and blind users alike, sighted users have access to persistent visual cues and representations (e.g., audio waveforms) on the GUI and can glance over multiple windows and use the mouse pointer to discover and control a myriad of information. Such visual UIs, however, are not typically designed with much consideration for screen reader users. Due to a screen reader’s sequential navigation of UI elements and ephemeral nature of audio feedback, screen reader users face unique experiences and challenges with discovering and keeping track of information across different parts of a DAW’s complex GUI. Furthermore, certain information and features may not have screen reader support at all, while several other less accessible features might require the use of workarounds that are not common knowledge for all users. To make matters worse, mainstream tutorials and guides primarily focus on mouse-based navigation of GUIs and do not account for the unique experiences of screen reader users. All of this makes learning how to use a DAW particularly difficult for new learners who are blind, many of whom turn to this online community to receive support from experienced helpers.

Our analysis revealed that help-seekers in this community asked a variety of questions to receive support in navigating and interacting with the DAW and associated software and hardware tools. Our analysis revealed three high-level types of questions posters asked regarding interaction with the DAW: (1) developing a preliminary understanding of a topic; (2) expanding or verifying current understanding of a topic; and (3) troubleshooting a problem encountered while using the DAW. Here, a topic may involve using a feature (e.g., a GUI element, keyboard shortcut) and/or performing a task using the DAW. Below we detail the categories of questions including some archetypes of common questions that help-seekers asked and how they structured these questions.

**4.2.1 Developing a Preliminary Understanding.** The questions in this category mainly involved help-seekers looking to gain knowledge about a GUI element, feature, or a DAW workflow they had not used before. Help-seekers formulated their questions in ways that would make it easier for helpers to understand the context of their use and provide necessary suggestions accordingly. Most help-seekers posted help requests that specified what they wanted to achieve (*goal*) and their current level of experience and knowledge (*prior knowledge*) before posing their main *query*. Occasionally, they added why they wanted to learn a new feature or a task (*motivation*) and how learning to use (or not being able to use) this feature would affect their work (*impact*). Below are a few examples of such posts.

**What is <a feature>? What can I do with <a feature>?** “As a new user of Logic Pro (*prior knowledge*), could anyone explain what ‘Event List’ is (*query*)?”

“Can anybody explain explain all use cases of the shortcut Command+Shift+E? (*query*) Since I can’t afford to hire someone to get my audio mixed, I thought I should learn about it (*motivation*).”

**How to do <a task>?** “I would like to mute out portions of an audio track instead of trimming it out (*goal*). Could folks share step-by-step guidance on how to do it (*query*). Having some instructions rather than figuring it out myself would save me a lot of time (*impact*).”

**4.2.2 Expanding or Verifying Current Understanding.** For this category of questions, help-seekers might have previously developed some understanding and first-hand experience with a feature or task on the DAW and they sought to build upon their existing knowledge by learning new use cases of a feature or more efficient ways of doing a task. They might also seek to verify their understanding or speculation about various use cases of a feature or task. For such questions, in addition to using the strategies described in the previous category (*goal, prior knowledge, motivation, query, impact*), help-seekers also incorporated in their questions how they currently use a certain GUI element, keyboard shortcut, or workflow (*current practice*) and/or their *speculation* about the potential use cases of a feature or task.

**Can I use <a feature/task> for a specific/different <use case>?** “... duplicating a track and having another instrument play the same tune (*current practice*), can you apply that same concept with vocals? Or do you have to manually sing background vocals? (*query*) I want to create a track with background vocals (*goal*). So I thought if I could duplicate the main track and simply apply some effects to it (*speculation*)?”

**Is what I know about <a feature/task> correct?** “Is my understanding correct that if I want minor chords I’ll need to use the ‘Event List’ and change the third of the major chords (*speculation, prior knowledge*)?”

**What is a more efficient or accessible way to do <a task>?** “With <App>... when I try to add plugins to a track (*goal*)... none of the plugins are labeled. So I’d need to open each plugin just to learn which one is loaded, and if I accidentally load the wrong one, I’d have to find and choose a new plugin and hope it’s the

correct one (*prior knowledge, current practice*)... Does anyone know a better way to deal with plugins(*query*)? My current workflow is starting to frustrate me and I’ve been dealing with this for nearly a month (*impact*).”

**4.2.3 Troubleshooting.** One of the most common types of questions help-seekers asked were about troubleshooting problems regarding their navigation and interaction with the DAW. For relatively experienced DAW users who are blind or low vision, as we will see in Section 4.3, troubleshooting by themselves may come with several challenges, including screen readers providing incomplete information about the system state or not sharing sufficient feedback on the result of their actions. By effectively describing a problem, help-seekers can leverage the prior experience of seasoned helpers within the community, who may have experienced similar problems themselves in the past, to identify the likely cause. To frame their questions for troubleshooting, help-seekers explained what task they wanted to perform (*goal*), what the *configuration of the system* was, what *problem* occurred, what *error messages* they received from their screen reader if any, what *steps or actions* they performed before the problem occurred or afterwards to address the problem, and how the problems affected their work (*impact*). Considering this contextual information, help-seekers asked why the problem was occurring (*query about reason*), what the problem or screen reader feedback meant (*query about meaning*), and how to solve the problem (*query about solution*).

**Why is <a problem> occurring?** “When I want to send someone a project file (*goal*), I save it with the typical Control+Option+Space shortcut on folder, click save, and compress the saved project into a zip file and send it (*actions performed before problem*). But after the other person receives and tries to open it, why does it fail to open (*query about reason*)?”

**How to solve <a problem>?** “I am recording flute into a mic (*goal*) through <App>, maximizing input level (*system configuration*), recording while listening to playback through my headphones (*actions performed before problem*). The problem is that the volume of the flute seems too loud (*problem*), which prevents me from hearing the background instruments (*impact*). How can I decrease the flute’s output volume without reducing the input volume so that it doesn’t dominate what I’m hearing (*query about solution*).”

“I have the same shortcuts configured (*system configuration*) and I even went to ‘select’ menu and chose ‘deselect all’ (*actions tried to address problem*) but it didn’t work (*problem*). Got any other suggestions or alternative tools? (*query about solution*)”

**What does <a screen reader feedback> mean?** “I was making a beat today (*goal*). The message pops up while working on the project, “error synchronizing audio and MIDI” (*screen reader feedback*). What does that mean? (*query about meaning*) How do I resolve it? (*query about solution*)”

If help-seekers found a solution to the problem on their own, occasionally they added the solution to their original thread to

support other new learners. For example, one help-seeker followed up on their own query, “I’ve solved it. Why didn’t I find it before!... I didn’t have all my tracks lined up correctly, so simple!” In many cases, however, troubleshooting required back-and-forth conversations between helpers and help-seekers, with experienced helpers adopting various strategies to provide effective suggestions, which we detail next.

### 4.3 How Do Experienced Members Support Help-seekers?

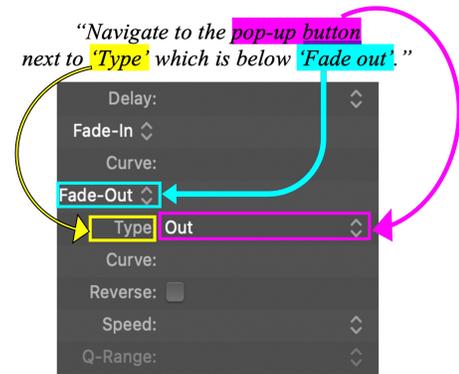
In response to help-seekers’ troubleshooting-related questions, helpers in this community provided step-by-step instructions on identifying the causes of the issues and ways to solve those, often requiring multiple iterations of back-and-forth replies and other members chiming in. We identified six key strategies helpers used to support help-seekers: (1) teaching others to use clear, standard terminology, (2) providing navigational signposting to help them locate GUI elements, (3) sharing strategies for extracting information by using screen reader and keyboard focus, (4) inferring system state in the absence of screen reader feedback, (5) resolving incorrect mental models of how features work, and (6) augmenting text-based support with additional resources.

#### 4.3.1 Teaching Help-seekers to Use Clear, Standard Terminology.

Screen reader feedback always includes specific terminology and names for different actions and GUI elements. If help-seekers describe actions using vague terminology or refer to GUI elements using unfamiliar or non-standard names, it can prevent helpers from understanding their questions and providing the support help-seekers need [28]. Thus, one of the first things helpers do is to teach help-seekers standard terminologies and exact names of GUI elements that are embedded in the DAW or its documentation and conveyed in screen reader announcements. In one thread, a poster asked: “How do you lock the mouse?” In response, another poster clarified that ‘mouse lock’ is not a standard term and provided the standard term instead: “I think it’s referred to as ‘mouse down,’ not ‘mouse lock.’” Helpers also teach help-seekers to distinguish between and avoid confusion regarding GUI elements that have similar names. For example, when suggesting the use of an action that is called ‘Create 2 Automation Points’, a helper clarified that he was not referring to performing another similar-named action titled ‘Create 1 Automation Point’ twice—and also provided an explanation about the difference between these two actions.

#### 4.3.2 Providing Navigational Signposting to Help Locate GUI Elements Using a Screen Reader.

Figuring out how to locate GUI elements is a prerequisite for navigating a complex DAW using a screen reader. Blind and low-vision users need to sequentially move the screen reader focus from one GUI element to another (buttons, menu items, read-only text etc.) that are organized in a hierarchical tree-like structure (i.e., the accessibility tree or hierarchy<sup>1</sup>). While a sighted user might glance across the display to locate spatially separated elements, screen reader users must move up, down, or across different branches and levels of the accessibility tree to reach



**Figure 2: ‘Type’ and ‘Fade out’ are two GUI elements spatially positioned nearby a particular pop-up button of interest on the DAW Logic Pro.**

the same elements—a time-consuming and potentially confusing process for users with less experience [54].

Given the complexities of screen reader navigation of DAW interfaces for relatively inexperienced help-seekers, one way helpers supported help-seekers in locating GUI elements they were not familiar with was by providing navigational signposting—or orienting linguistic cues that would help understand the hierarchical location and relationships of various elements. Helpers often referred to the name and type of a parent element in the accessibility tree, particularly if the name and location of the parent element were considered common knowledge. For example, when a help-seeker needed to identify the location of a dialog box named “Apple loop,” a helper suggested: “File > Export > Apple Loop will get you the dialog.” Referring to less common menu items like ‘Export’ and ‘Apple Loop’ without additional signposting (i.e., the well-known ‘File’ menu) would likely result in confusion on the help-seeker’s part and require additional turns of conversation.

Elements that were buried deep inside the GUI or in a portion of the GUI the help-seeker had not explored beforehand often required additional signposting, such as referencing other nearby elements. Even if none of the nearby elements were well-known, mentioning those might be helpful because if the help-seeker ran into those elements, they would be able to figure out that their desired element was nearby and narrow down their scope of search to a specific region of the GUI instead of traversing the entire accessibility tree. For instance, a helper specified the location of a pop-up button by saying “navigate to the pop-up button next to ‘Type’ which is below ‘Fade out.’” Here, ‘Type’ and ‘Fade out’ are two GUI elements spatially positioned nearby the particular pop-up button of interest (see Figure 2). Also interesting is the use of directional terms ‘next to’ and ‘below’—for two reasons. First, such usage of directional terms on the helper’s part indicates an assumption that the help-seeker had an understanding of how ‘next to’ and ‘below’ indicate sideways and vertical navigation of the screen reader cursor. Moreover, these terms suggest that in this particular case, the help-seeker would need to explore both sideways and vertically, because moving only in one direction (e.g., vertically) might result in the help-seeker forming an incorrect understanding of how the GUI element of

<sup>1</sup>Accessibility Objects and the Accessibility Hierarchy. <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/Accessibility/cocoaAXOverview/cocoaAXOverview.html>

interest is spatially organized with respect to nearby elements and eventually failing to find the desired GUI element.

We found many such instances of navigational signposting, where helpers referred to a variety of GUI elements by specifying their names and types (e.g., “*Amp Type pop-up button*” or “*Tracks radio button*”) or by triangulating the location of an element with respect to nearby elements (e.g., “*Region Inspector - usually the first table in the ‘Inspector’ just past where it says ‘Regions,’*” or “*the ‘Discloser’ triangle next to where it says ‘More.’*”). Collectively, these examples show the importance of strategic navigational signposting provided by helpers to support help-seekers in navigating and successfully locating obscure GUI elements and complement prior work that documented the use of similar signposts or landmarks by screen reader users in other contexts such as web browsing [8].

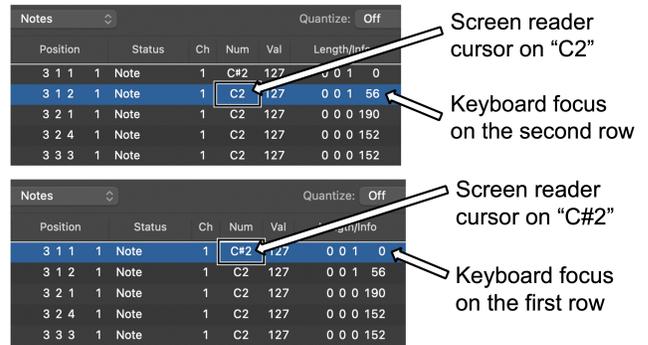
**4.3.3 Sharing Strategies for Extracting Information Through Screen Reader and Keyboard Focus.** In addition to navigating to a certain GUI element, users must also understand how to access relevant information from the element as part of accomplishing a task on a DAW. Efficiently performing an audio production task requires a user to figure out ways to maximize the amount of information that can be obtained about GUI elements in a DAW, which is uniquely challenging because often screen readers do not announce all available information about a GUI element that is visually available to sighted users. One efficient approach that helpers shared involved using the screen reader cursor and keyboard focus strategically to receive different types of auditory feedback. Important to note is that the screen reader cursor is not the same as keyboard focus.<sup>2</sup> Keyboard focus can only select and switch between certain GUI elements that users can interact with (e.g., buttons, sliders, text input fields, links etc.). Many other non-interactive elements such as text headings, paragraphs, and read-only warning text in a dialog box cannot receive keyboard focus. In contrast, the screen reader cursor can move between all interactive and non-interactive elements of an accessible user interface.

Even when both the screen reader cursor and keyboard focus can access an element, in some cases there is value in choosing to use one over the other. A key goal for helpers is to make inexperienced help-seekers aware of what kind of information to expect when using these two different modes of interaction. Consider the following excerpt where the helper is explaining how to traverse the ‘Event List’ feature that is organized as a table with each row containing a distinct event:

*“You can navigate through the table (by changing screen reader cursor) with Control+Option+Up and Down arrows to hear screen reader speak (the name of the musical note for each event on the list) as you navigate to them, or use just up and down arrows (for moving keyboard focus) to not hear screen reader speak but just the note played.”*

Here, the helper suggests navigating between the rows by moving the keyboard focus (using either up/down arrows) or the screen reader cursor (using Control+Option+Up/Down arrows). Importantly, each of these interactions in this case provides different information, which can be useful depending on what the help-seeker

<sup>2</sup>Depending on a user’s settings, both screen reader cursor and keyboard focus can move together in many situations, but that is not always the case.



**Figure 3: Navigating the Event List feature in a tabular format by moving screen reader cursor and keyboard focus.**

wants to achieve. Figure 3 illustrates this example. The keyboard focus (highlighted in blue) is on the second row of Event List, and the screen reader cursor (denoted by the black rectangle) is on a particular element ‘C2’ on the second row - which is the name of the musical note associated with this event. In this situation, if one moves the screen reader focus upwards using Control+Option+Up arrow, the screen reader will speak out the name of the note associated with the event in the first row, i.e., ‘C#2’. Alternatively, if one presses only the Up arrow to move the keyboard focus to the first row, screen reader will not give any spoken feedback, and instead the musical note C#2 will be played.

This example highlights how helpers teach others to combine different modes of interaction to extract information about GUI elements. We observed additional instances of this kind of guidance. For example, in the mixing window of Logic Pro, using Up and Down arrows would move the keyboard focus from one track to another but will not announce which track currently has the focus. In one thread, a helper suggested that the help-seeker could indirectly find out the current track under keyboard focus by pressing the ‘M’ key to toggle the muted/unmuted status of the currently selected track, because this action will announce both the muted/unmuted status and the name of the currently selected track. Overall, these instances reveal the challenges associated with obtaining and maximizing information about GUI elements, and how experienced helpers provide strategies to help screen reader users more effectively extract information from the DAW interface.

**4.3.4 Inferring System State in the Absence of Screen Reader Feedback.** One of the challenges for screen reader users is maintaining awareness of the system state as they move through a GUI and act on elements. Sighted users can see real-time changes on the GUI that make them aware of the effect of their actions (e.g., after pressing a keyboard shortcut, buttons turn red to indicate it has been toggled off). For blind users, sometimes the screen reader announces auditory feedback about changes to a GUI element (e.g., ‘mute checkbox, checked’ or ‘mute checkbox, unchecked’). At other times however, due to inconsistent and incomplete screen reader support, the screen reader does not provide feedback when GUI elements are acted upon. This can lead to difficulties in tracking the system state for users who are not familiar with the behavior of individual GUI elements. In such cases, expert helpers offer

workarounds to monitor the state of the element. For example, a helper explained how to keep track of the location of keyboard focus in a feature that does not provide auditory feedback: *“The Left and Right arrows move you back and forth between the 16 steps of a repeating musical pattern... I counted using my Left and Right arrows so I could tell what step I was in out of 16.”* Here, the helper’s explanation indicates that the screen reader does not directly announce which of the 16 steps is currently selected, therefore the helper shared their manual strategy of keeping track of the current step under keyboard focus by counting how many times they have pressed the Left/Right arrows.

While expert screen reader users might have developed strategies like the one described above to infer system state in the absence of direct screen reader feedback, this issue can exacerbate the troubleshooting process for inexperienced users. We see that help-seekers often run into situations where they cannot develop situational awareness about problems they are facing because they are uncertain about why there is no screen reader feedback after pressing a particular keyboard shortcut, which may be because screen reader feedback was not implemented for the associated action or because the keyboard shortcut did not have the intended effect or did not work at all. When help-seekers mention such problems in their posts, helpers use their prior experience and contextual knowledge to support help-seekers in inferring system state, even if the underlying causes are not readily apparent from screen reader announcements alone or simply there is no screen reader feedback. Bigham et al. [6] describe this as “not knowing what you don’t know,” which is a pervasive problem in web accessibility and appears in numerous threads within our corpus.

Others experience confusion when an action they performed previously suddenly no longer works, and they do not have screen reader feedback to help understand what is causing the problem. Consider the following conversation where the help-seeker is facing a problem with a specific screen reader shortcut not generating the intended effect, despite the screen reader focus being on the correct GUI element.

**Help-seeker:** *A couple versions ago, I could press Control+Option+Shift+Space (shortcut for simulating mouse click) on the Duration text field and edit it. With the updates, are we no longer able to edit this field with Control+Option+Shift+Space?*

**Helper:** *Try full screening Logic Pro so that there is less chance of another window being in the way.*

At first, the help-seeker wonders if the feature no longer works as it used to because of software updates—a common problem for screen reader users [16]. Instead, the helper offers another possibility given that the keyboard shortcut mentioned in help-seeker’s post is a screen reader specific shortcut for simulating mouse click. The helper deduces that the shortcut might not have worked on the desired text field because another application’s window was possibly covering the DAW, so the mouse click was taking place on the wrong application’s window—unbeknownst to the help-seeker because there is no screen reader feedback that announces this. Therefore, the helper suggests full screening Logic Pro window to mitigate the chance of another application’s window getting in the way. Even without direct access to the help-seeker’s GUI, this

experienced helper made an educated guess regarding the source of the problem based on their experiential knowledge and prior experience dealing with similar problems. Indeed, we found other instances of help-seekers facing similar problems (e.g., simulating mouse click brought screen reader focus to an unintended window, or text fields were no longer editable after simulating a double-click) where a helper was able to infer the likely cause based on their prior knowledge of simulating mouse actions.

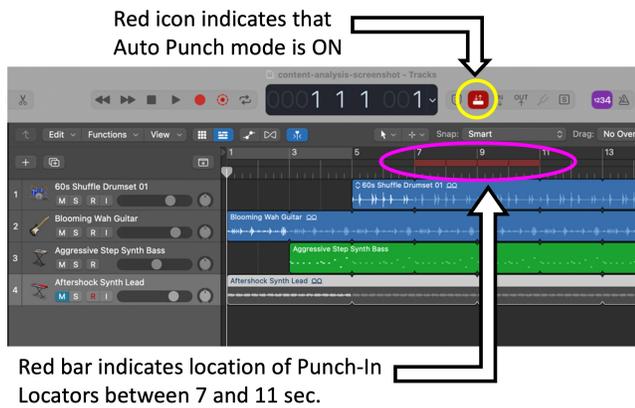
Overall, the above examples illustrate that the complexities associated with maintaining awareness of system state in the absence of screen reader feedback often leads to misunderstandings and requires troubleshooting with helpers who offer guidance based on both inference and their prior experience.

#### 4.3.5 Resolving Incorrect Mental Models of How Features Work.

Another form of support helpers provided help-seekers involves resolving incorrect mental models of how the GUI works. As demonstrated in Section 4.3.4, the absence and incomplete nature of screen reader feedback can result in an inaccurate understanding of the state of GUI and requires screen reader users to learn strategies to infer the current state. In more extreme cases, poor screen reader feedback may even result in incorrect mental models about the scope and functionality of certain features. Although expert screen reader users may have developed more accurate mental models about the scope and functionality of a DAW’s feature, this issue becomes particularly challenging for inexperienced help-seekers. Thus, helpers must identify help-seekers’ inaccurate mental models by parsing their posts and attempt to correct their misconceptions.

As an illustrative example, we detail a help-seeker’s question about Punch-In Locators and the complex back-and-forth exchange among multiple helpers. For context, Logic Pro has a specific pair of features called Auto-Punch mode and Punch-In Locators, where a user can place a start locator (similar to a marker) at one timestamp and an end locator at another timestamp on the audio timeline. Turning on the Auto-Punch mode automatically sets the two Punch-In Locators at random points on the audio timeline, and their positions can be later adjusted using keyboard shortcuts. Auto-Punch mode remains activated unless a screen reader user turns it off with a keyboard shortcut (see Figure 4). When Punch-In Locators are set on the audio timeline and the Auto-Punch mode is enabled, if a user attempts to record audio (e.g., an instrument or vocal) on any track, recording will take place only inside the region of time marked by the Punch-In Locators (i.e., between the start and end markers), but nothing will be recorded outside of the region marked by the Punch-In locators. Although a single pair of Punch-In Locators apply globally to all tracks, a screen reader only announces when Punch-In Locators are added or removed but does not announce the global scope of these locators. As a result, a relatively inexperienced help-seeker may form an incorrect mental model of how these locators work and where they apply (i.e., globally on all tracks or on a track-by-track basis). To make things worse, when the Auto-Punch mode is turned on using its designated keyboard shortcut, no screen reader feedback is provided, making it easy to unintentionally place and activate Punch-In locators. Below we detail a back-and-forth exchange related to an incorrect understanding of this feature.

A help-seeker begins by asking, *“I’m trying to record a piano on a track. The track is a software instrument set up as it should be...*



**Figure 4: Visual cues for the Auto Punch mode and Punch-In Locators available to sighted users. Red icon at the top indicates that the Auto Punch mode is ON. Red bar on the audio timeline indicates the presence of Punch-In Locators between 7 sec and 11 sec. The position of this red bar on the DAW’s global audio timeline indicates that Punch-In Locators apply to all tracks, not just a specific track.**

*I hear the instrument as I play, and yet nothing is recorded, why?”* A first helper (Helper-1) asks, *“Is Auto-Punch mode enabled and the Punch-In locators outside the position you want to record?”* Here, Helper-1 suspects that the source of problem could be that the Auto-Punch mode is enabled and there are Punch-In locators on the help-seeker’s project that they are not aware of, which is preventing the recording from taking place anywhere except between the Punch-In Locators. In response, the help-seeker claims that there should be no Punch-In Locators remaining on their project, because they have already removed the associated track where they may have set Punch-In Locators. They posted, *“I haven’t set any Punch-In locators. And even if I had set any Punch-In Locators by mistake, they should have disappeared with the track I deleted, thinking that the problem was with the track. But the problem still persists even after making a new track.”* This response shows the help-seeker’s misconception about how Punch-In Locators work, thinking that they are track-specific and should be removed when a track is removed when in fact they are global and apply to all tracks.

Another helper (Helper-2) joins the conversation and suspects the same problem as Helper-1. They suggest, *“It might be that Auto-Punch is on. Try turning Auto-Punch off... To do this, hit Control+Option+Command+P.”* The help-seeker, however, affirms their belief that the Auto-Punch mode is already deactivated after checking certain information on the DAW. They explain, *“When I look in the control bar, I see an area that says Auto-Punch layout item or some such...is turned off.”* Importantly, the help-seeker’s uncertain wording here suggests they might have confused another GUI item for the Auto-Punch mode button. Inconsistent and ambiguous screen reader feedback may have reinforced the help-seeker’s inaccurate mental model of the system state. Helper-2 suggests again that the help-seeker try the shortcut to make sure the Auto-Punch is turned off. In response, the help-seeker confirms that Auto-Punch is indeed off, saying, *“I’ve gone into the project again and checked. The*

*Auto-Punch is off... And it has always been like this in this project... Logic Pro simply refuses to record one single note even if I hear the piano... I cannot see any logical reason why it should be this way.”*

Not giving up on the issue, the help-seeker shares their troubleshooting steps by posting *“Update: I made another instrument track just to check if the problem is global or local. That is if it’s like this on every track or just a single one. Turns out the problem is global, I cannot record on any track... My confusion is now monumental.”* Here, upon debugging by trying to record on newly created tracks, the help-seeker finally identifies that the problem is not track-specific, rather it continues to persist even for new tracks. While the help-seeker has now been able to form a partial understanding of the problem (i.e., the scope is global rather than track-specific), they still cannot identify the source of the problem, which confuses them even more. After multiple responses from other helpers, the help-seeker finally responds with a resolution, *“The problem I had with not being able to record on a track is now solved. Turns out that the person (Helper-2) who wrote about Auto-Punch mode was completely right, it was turned on. How I did it I cannot understand...”*

The “monumental” confusion experienced by this help-seeker was enabled by their misunderstanding of the system state (i.e., that the Auto-Punch mode was on) as well as their incorrect mental model about the scope of Punch-In Locators. Even with repeated attempts by multiple helpers, it was exceedingly difficult for the help-seeker to rectify their inaccurate understanding of the system state and mental model of how the Punch-In Locator feature works. This is only one of many instances we observed related to helpers supporting help-seekers in resolving incorrect mental models of how the system works. Other instances involved help-seekers’ confusion about setting markers in a track and misunderstanding how copying a sound file from the file browser to Logic Pro works, both of which also point to the lack of clear and effective screen reader feedback contributes to users developing incorrect mental models, requiring expert helpers to leverage their experiential knowledge for sorting out these misconceptions.

#### 4.3.6 Augmenting Text-Based Support with Additional Resources.

In some situations, helpers and help-seekers did not manage to resolve confusions by describing problems and communicating suggestions through their text-based conversations alone. These situations included help-seekers struggling to replicate the helpers’ instructions, helpers not getting adequate situational awareness about the system state from their conversation with help-seekers, and helpers deeming a solution to be too complex to explain using text alone. In such cases, helpers and help-seekers moved to alternate modalities and mediums for sharing instructional support [43]. This would involve synchronous meetings over videoconferencing tools like Zoom, exchange of voice messages on WhatsApp, sending audio project files back and forth, making new audio tutorials, or referring help-seekers to existing resources (e.g., user manuals, audio/video tutorials produced by others) for performing the tasks at hand. In one thread, a help-seeker asked how to use a feature called ‘Marquee Selection Tool.’ In reply, a helper provided step-by-step written instructions for this tool. However, the help-seeker responded that they were still struggling with understanding the instructions. The helper then asked the help-seeker to send the audio project file so that they could make an audio tutorial on how to

use the feature using the help-seeker's file. Similarly, help-seekers sometimes shared audio recordings they made of screen reader announcements while replicating problematic tasks that needed troubleshooting. As another example, when a help-seeker asked how to use a feature called 'Punch Recording,' a helper shared that they found it difficult to explain this feature in written form, saying *"I do it all the time, but I can't seem to explain in writing."* Instead, they recorded an audio tutorial covering the basics of that particular feature and shared it on the conversation thread.

## 5 DISCUSSION

Our analysis details how blind and low-vision screen reader users engage in peer-to-peer instructional support in an online text-based Q&A community centered around audio production software. While community members posted about a variety of topics, including product reviews and recommendations, offers of available audio editing services and equipment, and advocacy for accessible tools and resources, the overwhelming majority of posts asked for help navigating and interacting within audio production tools. We turn our attention to a discussion of key insights drawn from the ways in which help-seekers and helpers work together to achieve shared understanding when resolving technical issues as well as future opportunities for designing more effective tools for peer-to-peer instructional support among screen reader users.

### 5.1 Achieving Shared Understanding During Collaborative Troubleshooting Among Screen Reader Users

Prior work shows that screen readers often do not support the same level of awareness about the GUI state as visual representation does in terms of glanceability, discoverability, and gestalt perception [17, 53, 54]. Our findings also show evidence of this: we observed instances where help-seekers could not develop a complete awareness about their system state from the screen reader translation of sophisticated visual feedback alone, primarily because the auditory feedback was inadequate or missing and did not allow for a way to verify the task state, despite the information being visually available. As an example, in Section 4.3.4, the help-seeker had no way to know whether another application's window was covering their DAW—a likely reason behind their simulation of mouse action not working—since screen reader does not provide active feedback for when interfaces overlap. Nevertheless, an interesting aspect in this scenario is that the helper was able to draw upon their prior experience in predicting the source of the help-seeker's problem. Here, we can see that for expert helpers, experiential knowledge may sometimes compensate for the lack of feedback in understanding system state. However, for the help-seeker, the lack of experiential knowledge would not have stood in the way of understanding system state if they received real time feedback about the overlapping windows—something sighted users readily receive by visually noticing the spatial positioning of different windows.

Our analysis provides evidence of how such asymmetry between visual information in a GUI and sparse auditory feedback through screen readers [17] can both delay and disrupt achieving shared understandings between helpers and help-seekers. In Section 4.3.5, we notice how the lack of screen reader feedback on the system

state makes it more difficult for both the help-seeker and helpers to develop a common understanding about the source of the problem around the presence/absence of Punch-In Locators. With persistent visual cues, a sighted user would have easily seen that the Punch-In Locators were global and cannot be removed simply by deleting a track. However, since this information is not readily provided by the screen reader to the help-seeker, the helper also had to include additional description of how Punch-In Locators worked to successfully achieve mutual understanding with the help-seeker. Thus, helpers often need to anticipate the sources of misalignments in both communication and knowledge based on their prior experiences and seek to provide additional information to avoid or repair those misalignments [14].

To this end, our analysis of conversations in this community show a number of distinct conversational norms and practices adopted by the blind and low vision members to facilitate the process of achieving a shared understanding about a graphical user interface. For example, establishing effective navigational signposting of GUI elements provides a unique strategy enacted by screen reader users for other screen reader users, as explained in Section 4.3.2. While collaboration among sighted people also involves the use of directional terms (e.g., top-left, near the bottom) and deictic moves (e.g., here, there) to refer to particular regions of a GUI [13, 22], these terms often carry little meaning for screen reader users in reconstructing a meaningful spatial structure of a complex visual interface. Therefore, contextualizing the position of an element by referring to its nearby and parent/child elements in the hierarchical accessibility tree as well as using standard element names and types are strategic moves to maximize the chances of achieving a shared understanding between multiple screen reader users in an asynchronous text-based conversational medium.

Overall, this interplay between screen reader announcements about system state (or lack thereof), experiential knowledge, and conversational practices designed to scaffold novice users raises important questions about the process of achieving shared understanding of complex, visually-oriented tools among screen reader users. To this end, future research could seek to extend our current understanding of the conversational grounding process in peer-to-peer support among blind and low vision software users, a concept that up until now has mostly been studied in the context of collaboration between sighted people [13, 22, 32, 56].

### 5.2 Reimagining Peer-to-Peer Instructional Support for Screen Reader Users

Understanding the practices and norms of this particular online community reveals opportunities for reimagining the ways in which online platforms enable peer-to-peer instructional support among screen reader users.

**5.2.1 Learning From Conversational Norms.** Our findings in Sections 4.2 and 4.3 reveal conversational norms that help-seekers and helpers use to formulate their questions and responses in this community. For example, help-seekers often include their prior knowledge, current practice, and goal in addition to their specific query. Helpers' responses also contain useful information regarding navigating to and obtaining information from GUI elements, which often describe step-by-step solutions and offer keyboard shortcuts.

Future work could leverage these conversational norms that are rooted in screen reader users' shared experience with software tools to prompt posters to provide additional information if they did not and to automatically generate additional details that complement helpers' instructions. For example, such a system could parse posts to look for non-standard terminology and suggest correct terms, which would potentially reduce confusion and the need for additional turns of conversation for clarification. As another example, future systems could parse helpers' instructions regarding a navigational or information retrieval task and generate audio recordings of what the screen reader announcements and associated keyboard actions at various stages of the task would sound like.

**5.2.2 Leveraging Traces of Screen Reader Activity to Support Shared Awareness.** The text-based nature of this online community, which many members use like a mailing list, provides an accessible format for seeking and receiving support among blind and low vision screen reader users. Although text-based discussion communities have many benefits, such as their accessibility and searchability, community members sometimes had difficulty communicating their problems and solutions and resolving misunderstandings using text alone. As our findings reveal, some helpers and help-seekers would switch to other richer communication channels where they could hear the other person's screen reader. To this end, future work could explore features that allow capturing and sharing text or audio recordings of screen reader announcements. Help-seekers' screen reader announcements provide helpers with more context about their DAW's system state, any error messages they might be getting, and more. Conversely, by listening to screen reader feedback shared by helpers, help-seekers can get a concrete understanding of what the navigation or interaction with GUI elements sound like and how information is presented within different GUI elements [57]. Of equal importance is knowing the keyboard actions preceding each screen reader announcement to understand what kind of announcement (or lack thereof) to expect in certain situations. Many screen readers already store recent history of screen reader speech (last several announcements) and allow users to copy and share them in textual form. Apple's VoiceOver screen reader can also output the most recent announcement as an audio file. Just as sighted people share screenshots of GUIs with helpers when they encounter a problem, future work could explore richer multimodal traces of screen reader activity as a way of providing additional context when requesting and providing help.

**5.2.3 Embedding Conversational Activity and Instruction Within Software Tools.** Another direction for future work involves enhancing the ways in which screen reader users can discuss and reference parts of a complex visual interface. Towards this end, peer support could be embedded within the DAW environment, such as how Pro Tools (a prominent DAW) recently introduced cloud-based features that allow people to chat while working within a shared project. Similar kinds of integrated support could be critical for blind and low vision audio producers who are just entering the industry or looking to advance their skills. However, few interactive tools of this nature exist for screen reader users, and without creating new tools that emphasize screen reader access from the start, existing systems (e.g., the Pro Tools embedded chat feature) are likely to

widen existing gaps in access. Some other work has explored interactive tutorials for screen reader users. For example, the Tutoria11y system [57] captures a blind instructor's explanations in an audio file and plays their instructions step-by-step as a learner progresses through various screen reader commands. Developing future systems that enable seamless integration of instructor and helper activity within the environment of interest is a promising approach for supporting screen reader users in learning to use complex software required for various forms of skilled work. Another promising approach for supporting such on-demand instruction within software tools for screen reader users could be the use of modern artificial intelligence technologies. As an example, AI-based systems such as large language models (LLMs) are becoming increasingly popular in help-seeking contexts such as getting coding tips and identifying mistakes in code [52]. However, it remains to be seen how well these platforms support blind and low vision users while seeking help with their coding problems as well as in other contexts that involve learning software tools, especially considering the unique challenges associated with learning to navigate complex GUIs using screen reader technology. Our findings about misalignments between helpers and help-seekers, and the conversational strategies they use to resolve such issues, might provide starting points for both studying and improving the design of AI-based help-seeking platforms for screen reader users.

## 6 CONCLUSION

In this paper, we present results from an analysis of Q&A conversations between blind audio producers who seek and provide help related to using complex audio editing software with a screen reader. Our findings detail the kinds of questions posters ask in this online community, the vast majority of which are related to understanding how to navigate and interact with the Logic Pro DAW using VoiceOver. An in-depth analysis of help-seekers' queries reveals the kinds of instructional support screen reader users are seeking and how they formulate their requests for support. In turn, helpers use a variety of strategies to support collaborative troubleshooting with help-seekers and achieve a shared understanding of various problems and solutions. Reflecting upon our findings, we discuss the complexities of how help-seekers and helpers reach a shared understanding during troubleshooting as well as opportunities for designing new systems that enable more effective peer-to-peer instructional support for screen reader users.

## ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under Grant No. IIS-1901456. We are grateful to Maitraye Das, Lotus Zhang, our colleagues from Accessibility Research Collective at UCI, and our reviewers for their thoughtful comments on earlier drafts of this paper. We also thank Marcelo Worsley, Shaun Kane, and our colleagues from CollabLab at Northwestern University and Inclusive Design Lab at the University of Washington for their continuous engagement in our research.

## REFERENCES

- [1] Ali Abdolrahmani, Kevin M. Storer, Antony Rishin Mukkath Roy, Ravi Kuber, and Stacy M. Branham. 2020. Blind Leading the Sighted: Drawing Design Insights from Blind Users towards More Productivity-Oriented Voice Interfaces. *ACM*

- Trans. Access. Comput.* 12, 4, Article 18 (jan 2020), 35 pages. <https://doi.org/10.1145/3368426>
- [2] Lada A. Adamic, Jun Zhang, Eytan Bakshy, and Mark S. Ackerman. 2008. Knowledge Sharing and Yahoo Answers: Everyone Knows Something. In *Proceedings of the 17th International Conference on World Wide Web (Beijing, China) (WWW '08)*. Association for Computing Machinery, New York, NY, USA, 665–674. <https://doi.org/10.1145/1367497.1367587>
  - [3] Cynthia L. Bennett, Erin Brady, and Stacy M. Branham. 2018. Interdependence as a Frame for Assistive Technology Research and Design. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility (Galway, Ireland) (ASSETS '18)*. Association for Computing Machinery, New York, NY, USA, 161–173. <https://doi.org/10.1145/3234695.3236348>
  - [4] Cynthia L. Bennett, Daniela K. Rosner, and Alex S. Taylor. 2020. The Care Work of Access. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–15. <https://doi.org/10.1145/3313831.3376568>
  - [5] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samuel White, and Tom Yeh. 2010. VizWiz: Nearly Real-Time Answers to Visual Questions. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology (New York, New York, USA) (UIST '10)*. Association for Computing Machinery, New York, NY, USA, 333–342. <https://doi.org/10.1145/1866029.1866080>
  - [6] 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 (Baltimore, Maryland, USA) (ASSETS '17)*. Association for Computing Machinery, New York, NY, USA, 101–109. <https://doi.org/10.1145/3132525.3132533>
  - [7] Katya Borgos-Rodriguez, Kathryn E. Ringland, and Anne Marie Piper. 2019. MyAutosomeFamilyLife: Analyzing Parents of Children with Developmental Disabilities on YouTube. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 94 (nov 2019), 26 pages. <https://doi.org/10.1145/3359196>
  - [8] Yevgen Borodin, Jeffrey P. Bigham, Glenn Dausch, and I. V. Ramakrishnan. 2010. More than Meets the Eye: A Survey of Screen-Reader Browsing Strategies. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) (Raleigh, North Carolina) (W4A '10)*. Association for Computing Machinery, New York, NY, USA, Article 13, 10 pages. <https://doi.org/10.1145/1805986.1806005>
  - [9] Erin L. Brady, Yu Zhong, Meredith Ringel Morris, and Jeffrey P. Bigham. 2013. Investigating the Appropriateness of Social Network Question Asking as a Resource for Blind Users. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (San Antonio, Texas, USA) (CSCW '13)*. Association for Computing Machinery, New York, NY, USA, 1225–1236. <https://doi.org/10.1145/2441776.2441915>
  - [10] Stacy M. Branham and Shaun K. Kane. 2015. Collaborative Accessibility: How Blind and Sighted Companions Co-Creatable Accessible Home Spaces. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (Seoul, Republic of Korea) (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2373–2382. <https://doi.org/10.1145/2702123.2702511>
  - [11] Virginia Braun, Victoria Clarke, Nikki Hayfield, and Gareth Terry. 2019. *Thematic Analysis*. Springer Singapore, Singapore, 843–860. [https://doi.org/10.1007/978-981-10-5251-4\\_103](https://doi.org/10.1007/978-981-10-5251-4_103)
  - [12] Amy Bruckman. 2002. Studying the Amateur Artist: A Perspective on Disguising Data Collected in Human Subjects Research on the Internet. *Ethics and Information Technology* 4 (2002), 217–231. <https://doi.org/10.1023/A:1021316409277>
  - [13] Herbert H. Clark and Susan E. Brennan. 1992. Grounding in communication. In *Readings in Groupware and Computer Supported Cooperative Work: Assisting Human-Human Collaboration*, Ronald Baecker (Ed.), Morgan Kaufmann.
  - [14] Gregorio Convertino, Helena M. Mentis, Mary Beth Rosson, John M. Carroll, Aleksandra Slavkovic, and Craig H. Ganoe. 2008. Articulating Common Ground in Cooperative Work: Content and Process. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Florence, Italy) (CHI '08)*. Association for Computing Machinery, New York, NY, USA, 1637–1646. <https://doi.org/10.1145/1357054.1357310>
  - [15] 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 (Honolulu, HI, USA) (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–15. <https://doi.org/10.1145/3313831.3376477>
  - [16] Maitraye Das, Darren Gergle, and Anne Marie Piper. 2019. "It Doesn't Win You Friends": Understanding Accessibility in Collaborative Writing for People with Vision Impairments. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 191 (nov 2019), 26 pages. <https://doi.org/10.1145/3359293>
  - [17] Maitraye Das, Thomas Barlow McHugh, Anne Marie Piper, and Darren Gergle. 2022. Co11ab: Augmenting Accessibility in Synchronous Collaborative Writing for People with Vision Impairments. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (New Orleans, LA, USA) (CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 196, 18 pages. <https://doi.org/10.1145/3491102.3501918>
  - [18] Maitraye Das, John Tang, Kathryn E. Ringland, and Anne Marie Piper. 2021. Towards Accessible Remote Work: Understanding Work-from-Home Practices of Neurodivergent Professionals. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 183 (apr 2021), 30 pages. <https://doi.org/10.1145/3449282>
  - [19] Patrick Marcel Joseph Dubois, Mahya Maftouni, Parmit K. Chilana, Joanna McGrenere, and Andrea Bunt. 2020. Gender Differences in Graphic Design Q&As: How Community and Site Characteristics Contribute to Gender Gaps in Answering Questions. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW2, Article 113 (oct 2020), 26 pages. <https://doi.org/10.1145/3415184>
  - [20] Hasan Shahid Ferdous, Dipto Das, and Farhana Murtaza Choudhury. 2018. Social Media Question Asking (SMQA): Whom Do We Tag and Why?. In *Proceedings of the 30th Australian Conference on Computer-Human Interaction (Melbourne, Australia) (OzCHI '18)*. Association for Computing Machinery, New York, NY, USA, 12–22. <https://doi.org/10.1145/3292147.3292173>
  - [21] Hengyi Fu and Yun Fan. 2016. Music Information Seeking via Social Q&A: An Analysis of Questions in Music StackExchange Community. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries (Newark, New Jersey, USA) (JCDL '16)*. Association for Computing Machinery, New York, NY, USA, 139–142. <https://doi.org/10.1145/2910896.2910914>
  - [22] Darren Gergle, Robert E. Kraut, and Susan R. Fussell. 2013. Using Visual Information for Grounding and Awareness in Collaborative Tasks. *Human-Computer Interaction* 28, 1 (2013), 1–39. <https://doi.org/10.1080/07370024.2012.678246>
  - [23] Taylor Goffrid, Kelly Mack, Kathryn J Lum, Evelyn Yang, Jessica Hodgins, Scott E Hudson, and Jennifer Mankoff. 2021. Stitching Together the Experiences of Disabled Knitters. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (Yokohama, Japan) (CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 488, 14 pages. <https://doi.org/10.1145/3411764.3445521>
  - [24] Anja Guzzi, Alberto Bacchelli, Michele Lanza, Martin Pinzger, and Arie van Deursen. 2013. Communication in open source software development mailing lists. In *2013 10th Working Conference on Mining Software Repositories (MSR)*. 277–286. <https://doi.org/10.1109/MSR.2013.6624039>
  - [25] F. Maxwell Harper, Daniel Moy, and Joseph A. Konstan. 2009. Facts or Friends? Distinguishing Informational and Conversational Questions in Social Q&A Sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Boston, MA, USA) (CHI '09)*. Association for Computing Machinery, New York, NY, USA, 759–768. <https://doi.org/10.1145/1518701.1518819>
  - [26] Megan Hofmann, Devva Kasnitz, Jennifer Mankoff, and Cynthia L. Bennett. 2020. Living Disability Theory: Reflections on Access, Research, and Design. In *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility (Virtual Event, Greece) (ASSETS '20)*. Association for Computing Machinery, New York, NY, USA, Article 4, 13 pages. <https://doi.org/10.1145/3373625.3416996>
  - [27] Gary Hsieh, Robert E. Kraut, and Scott E. Hudson. 2010. Why Pay? Exploring How Financial Incentives Are Used for Question & Answer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Atlanta, Georgia, USA) (CHI '10)*. Association for Computing Machinery, New York, NY, USA, 305–314. <https://doi.org/10.1145/1753326.1753373>
  - [28] Nathaniel Hudson, Parmit K. Chilana, Xiaoyu Guo, Jason Day, and Edmund Liu. 2015. Understanding triggers for clarification requests in community-based software help forums. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 189–193. <https://doi.org/10.1109/VLHCC.2015.7357216>
  - [29] Jazette Johnson, Andrew Begel, Richard Ladner, and Denae Ford. 2022. Program-L: Online Help Seeking Behaviors by Blind and Low Vision Programmers. In *2022 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 1–6. <https://doi.org/10.1109/VLHCC53370.2022.9833106>
  - [30] Jazette Johnson, Rebecca W. Black, and Gillian R. Hayes. 2020. Roles in the Discussion: An Analysis of Social Support in an Online Forum for People with Dementia. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW2, Article 127 (oct 2020), 30 pages. <https://doi.org/10.1145/3415198>
  - [31] Aniket Kittur and Robert E. Kraut. 2008. Harnessing the Wisdom of Crowds in Wikipedia: Quality through Coordination. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (San Diego, CA, USA) (CSCW '08)*. Association for Computing Machinery, New York, NY, USA, 37–46. <https://doi.org/10.1145/1460563.1460572>
  - [32] Robert E. Kraut, Darren Gergle, and Susan R. Fussell. 2002. The Use of Visual Information in Shared Visual Spaces: Informing the Development of Virtual Co-Presence. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (New Orleans, Louisiana, USA) (CSCW '02)*. Association for Computing Machinery, New York, NY, USA, 31–40. <https://doi.org/10.1145/587078.587084>
  - [33] J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33, 1 (1977), 159–174. <https://doi.org/10.2307/2529310>
  - [34] Kelly Mack, Maitraye Das, Dhruv Jain, Danielle Bragg, John Tang, Andrew Begel, Erin Beneteau, Josh Urban Davis, Abraham Glasser, Joon Sung Park, and Venkatesh Potluri. 2021. Mixed Abilities and Varied Experiences: A Group

- Autoethnography of a Virtual Summer Internship. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility* (Virtual Event, USA) (ASSETS '21). Association for Computing Machinery, New York, NY, USA, Article 21, 13 pages. <https://doi.org/10.1145/3441852.3471199>
- [35] Kelly Mack, Emma McDonnell, Venkatesh Potluri, Maggie Xu, Jailyn Zabala, Jeffrey Bigham, Jennifer Mankoff, and Cynthia Bennett. 2022. Anticipate and Adjust: Cultivating Access in Human-Centered Methods. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 603, 18 pages. <https://doi.org/10.1145/3491102.3501882>
- [36] Kelly Mack, Emma J. McDonnell, Leah Findlater, and Heather D. Evans. 2022. Chronically Under-Addressed: Considerations for HCI Accessibility Practice with Chronically Ill People. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility* (Athens, Greece) (ASSETS '22). Association for Computing Machinery, New York, NY, USA, Article 9, 15 pages. <https://doi.org/10.1145/3517428.3544803>
- [37] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. 2011. Design Lessons from the Fastest Q&A Site in the West. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) (CHI '11). Association for Computing Machinery, New York, NY, USA, 2857–2866. <https://doi.org/10.1145/1978942.1979366>
- [38] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. 2019. Reliability and Inter-Rater Reliability in Qualitative Research: Norms and Guidelines for CSCW and HCI Practice. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 72 (nov 2019), 23 pages. <https://doi.org/10.1145/3359174>
- [39] Oussama Metatla, Fiore Martin, Adam Parkinson, Nick Bryan-Kinns, Tony Stockman, and Atau Tanaka. 2016. Audio-Haptic Interfaces for Digital Audio Workstations. *Journal on Multimodal User Interfaces* 10, 3 (Sept. 2016), 247–258. <https://doi.org/10.1007/s12193-016-0217-8>
- [40] Megan A. Moreno, Natalie Goniu, Peter S. Moreno, and Douglas Diekema. 2013. Ethics of social media research: Common concerns and practical considerations. *Cyberpsychology, Behavior, and Social Networking* 16, 9 (2013), 708–713. <https://doi.org/10.1089/cyber.2012.0334>
- [41] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. 2010. What Do People Ask Their Social Networks, and Why? A Survey Study of Status Message Q&A Behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) (CHI '10). Association for Computing Machinery, New York, NY, USA, 1739–1748. <https://doi.org/10.1145/1753326.1753587>
- [42] Kevin Kyung Nam, Mark S. Ackerman, and Lada A. Adamic. 2009. Questions in, Knowledge in? A Study of Naver's Question Answering Community. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) (CHI '09). Association for Computing Machinery, New York, NY, USA, 779–788. <https://doi.org/10.1145/1518701.1518821>
- [43] Bonnie A. Nardi, Steve Whittaker, and Erin Bradner. 2000. Interaction and Outeraction: Instant Messaging in Action. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work* (Philadelphia, Pennsylvania, USA) (CSCW '00). Association for Computing Machinery, New York, NY, USA, 79–88. <https://doi.org/10.1145/358916.358975>
- [44] Mark W. Newman, Debra Lauterbach, Sean A. Munson, Paul Resnick, and Margaret E. Morris. 2011. It's Not That I Don't Have Problems, I'm Just Not Putting Them on Facebook: Challenges and Opportunities in Using Online Social Networks for Health. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work* (Hangzhou, China) (CSCW '11). Association for Computing Machinery, New York, NY, USA, 341–350. <https://doi.org/10.1145/1958824.1958876>
- [45] Nigini Oliveira, Michael Muller, Nazareno Andrade, and Katharina Reinecke. 2018. The Exchange in StackExchange: Divergences between Stack Overflow and Its Culturally Diverse Participants. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 130 (nov 2018), 22 pages. <https://doi.org/10.1145/3274399>
- [46] Shotaro Omori and Ikuko Eguchi Yairi. 2013. Collaborative Music Application for Visually Impaired People with Tangible Objects on Table. In *ASSETS'13*. ACM Press, 1–2. <https://doi.org/10.1145/2513383.2513403>
- [47] Maulishree Pandey, Vaishnav Kameswaran, Hrishikesh V. Rao, Sile O'Modhrain, and Steve Oney. 2021. Understanding Accessibility and Collaboration in Programming for People with Visual Impairments. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 129 (apr 2021), 30 pages. <https://doi.org/10.1145/3449203>
- [48] Sharoda A. Paul, Lichan Hong, and Ed H. Chi. 2012. Who is Authoritative? Understanding Reputation Mechanisms in Quora. *ArXiv abs/1204.3724* (2012).
- [49] William Payne, Fahiha Ahmed, Michael Gardell, R. Luke DuBois, and Amy Hurst. 2022. SoundCells: Designing a Browser-Based Music Technology for Braille and Print Notation. In *Proceedings of the 19th International Web for All Conference* (Lyon, France) (W4A '22). Association for Computing Machinery, New York, NY, USA, Article 14, 12 pages. <https://doi.org/10.1145/3493612.3520462>
- [50] William Christopher Payne, Alex Yixuan Xu, Fahiha Ahmed, Lisa Ye, and Amy Hurst. 2020. How Blind and Visually Impaired Composers, Producers, and Songwriters Leverage and Adapt Music Technology. In *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility* (Virtual Event, Greece) (ASSETS '20). Association for Computing Machinery, New York, NY, USA, Article 35, 12 pages. <https://doi.org/10.1145/3373625.3417002>
- [51] Louise F. Pendry and Jessica Salvatore. 2015. Individual and social benefits of online discussion forums. *Computers in Human Behavior* 50 (2015), 211–220. <https://doi.org/10.1016/j.chb.2015.03.067>
- [52] Jeffrey M. Perkel. 2023. Six Tips for Better Coding with ChatGPT. *Nature* 618, 7964 (June 2023), 422–423. <https://doi.org/10.1038/d41586-023-01833-0>
- [53] Venkatesh Potluri, Tadashi E. Grindeland, Jon E. Froehlich, and Jennifer Mankoff. 2021. Examining Visual Semantic Understanding in Blind and Low-Vision Technology Users. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 35, 14 pages. <https://doi.org/10.1145/3411764.3445040>
- [54] 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* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–11. <https://doi.org/10.1145/3173574.3174192>
- [55] Kavana Ramesh, Laton Vermette, and Parmit K. Chilana. 2021. Setting up, Troubleshooting, and Innovating on the Delivery of Online Instruction: A Case Study of an LMS Q&A Forum. In *Proceedings of the Eighth ACM Conference on Learning @ Scale* (Virtual Event, Germany) (L@S '21). Association for Computing Machinery, New York, NY, USA, 59–67. <https://doi.org/10.1145/3430895.3460133>
- [56] Abhishek Ranjan, Jeremy P. Birnholtz, and Ravin Balakrishnan. 2007. Dynamic Shared Visual Spaces: Experimenting with Automatic Camera Control in a Remote Repair Task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '07). Association for Computing Machinery, New York, NY, USA, 1177–1186. <https://doi.org/10.1145/1240624.1240802>
- [57] Abir Saha, Thomas Barlow McHugh, and Anne Marie Piper. 2023. Tutoria11y: Enhancing Accessible Interactive Tutorial Creation by Blind Audio Producers. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 220, 14 pages. <https://doi.org/10.1145/3544548.3580698>
- [58] Abir Saha and Anne Marie Piper. 2020. Understanding Audio Production Practices of People with Vision Impairments. In *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility* (Virtual Event, Greece) (ASSETS '20). Association for Computing Machinery, New York, NY, USA, Article 36, 13 pages. <https://doi.org/10.1145/3373625.3416993>
- [59] Vandana Singh and Michael B. Twidale. 2008. The Confusion of Crowds: Non-Dyadic Help Interactions. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work* (San Diego, CA, USA) (CSCW '08). Association for Computing Machinery, New York, NY, USA, 699–702. <https://doi.org/10.1145/1460563.1460670>
- [60] Sulayman K. Sowe, Ioannis Stamelos, and Lefteris Angelis. 2008. Understanding knowledge sharing activities in free/open source software projects: An empirical study. *Journal of Systems and Software* 81, 3 (2008), 431–446. <https://doi.org/10.1016/j.jss.2007.03.086> Selected Papers from the 2006 Brazilian Symposia on Databases and on Software Engineering.
- [61] 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* (San Diego, CA, USA) (DIS '19). Association for Computing Machinery, New York, NY, USA, 385–398. <https://doi.org/10.1145/3322276.3322374>
- [62] Kevin M. Storer, Harini Sampath, and M. Alice Alice Merrick. 2021. "It's Just Everything Outside of the IDE That's the Problem": Information Seeking by Software Developers with Visual Impairments. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 487, 12 pages. <https://doi.org/10.1145/3411764.3445090>
- [63] Atau Tanaka and Adam Parkinson. 2016. Haptic Wave: A Cross-Modal Interface for Visually Impaired Audio Producers. In *CHI'16*. ACM Press, 2150–2161. <https://doi.org/10.1145/2858036.2858304>
- [64] Hon Jie Teo and A. Johri. 2013. Experts learn more (than Newcomers): An exploratory study of argumentation in an online help forum. *Conference on Computer-Supported Collaborative Learning Conference* 1 (01 2013), 470–477.
- [65] Hon Jie Teo and Aditya Johri. 2014. Fast, Functional, and Fitting: Expert Response Dynamics and Response Quality in an Online Newcomer Help Forum. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (Baltimore, Maryland, USA) (CSCW '14). Association for Computing Machinery, New York, NY, USA, 332–341. <https://doi.org/10.1145/2531602.2531731>
- [66] Maria Tomprou, Laura Dabbish, Robert E. Kraut, and Fannie Liu. 2019. Career Mentoring in Online Communities: Seeking and Receiving Advice from an Online Community. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland UK) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3290605.3300883>

- [67] Bogdan Vasilescu, Alexander Serebrenik, Prem Devanbu, and Vladimir Filkov. 2014. How Social Q&A Sites Are Changing Knowledge Sharing in Open Source Software Communities. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (Baltimore, Maryland, USA) (*CSCW '14*). Association for Computing Machinery, New York, NY, USA, 342–354. <https://doi.org/10.1145/2531602.2531659>
- [68] Gang Wang, Konark Gill, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. 2013. Wisdom in the Social Crowd: An Analysis of Quora. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) (*WWW '13*). Association for Computing Machinery, New York, NY, USA, 1341–1352. <https://doi.org/10.1145/2488388.2488506>
- [69] Anon Ymous, Katta Spiel, Os Keyes, Rua M. Williams, Judith Good, Eva Hornecker, and Cynthia L. Bennett. 2020. "I Am Just Terrified of My Future" — Epistemic Violence in Disability Related Technology Research. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI EA '20*). Association for Computing Machinery, New York, NY, USA, 1–16. <https://doi.org/10.1145/3334480.3381828>

## GLOSSARY

**Auto-Punch mode** is a special recording mode in which a portion of an audio track can be re-recorded during playback, without touching any other part of the track. 9

**DAW** or Digital Audio Workstation is a software tool used for recording, editing, and producing audio content. Some common DAWs are Garageband, Logic Pro, etc. 1–5, 7–10

**Event List** shows a list of events such as where a musical note or a audio clip starts on the timeline, length of the clip and so on in a table format. 8

**Logic Pro** is a DAW developed by Apple. 1, 3–5, 7–10

**Punch-In Locator** marks the start or end point on the timeline for recording in Auto Punch mode and limits what part of a audio track will be overwritten. 9