"*I Would Just Ask Someone*": Learning Feature-Rich Design Software in the Modern Workplace

Kimia Kiani
Simon Fraser University
Burnaby, BC, Canada
kimia_kiani@sfu.ca

Parmit K. Chilana
Simon Fraser University
Burnaby, BC, Canada
pchilana@cs.sfu.ca

Andrea Bunt
University of Manitoba
Winnipeg, MB, Canada
bunt@cs.manitoba.ca

Tovi Grossman
University of Toronto, Autodesk Research
Toronto, ON, Canada
tovi@dgp.toronto.edu

George Fitzmaurice
Autodesk Research
Toronto, ON, Canada
george.firzmaurice@autodesk.com

***Abstract*— Design professionals increasingly need to learn and use multiple feature-rich software to design complex artifacts in a variety of domains. In this work, we aimed to understand how design professionals learn new software features and seek help in today's modern workplace of interconnected spaces and increased access to shared online learning resources. We conducted 20 interviews with design professionals working in different disciplines such as architecture and planning, construction, structural design, and broadcast media, who used complex feature-rich applications on a daily basis. We found that despite the wide availability of learning resources on the web and decades of research dedicated to workplace knowledge management tools, design professionals still rely heavily on interpersonal help-seeking. Furthermore, while companies have devised a variety of technological channels to answer employees' help requests, such intermediate tools were often abandoned, and help-seeking conversations shifted to face-to-face learning episodes. Our findings point to design opportunities and shortcomings in the design of remote communication tools when seeking help for feature-rich software applications.**

Keywords— Software help-seeking, software learnability, workplace help-seeking, interpersonal help-seeking, design tools

I. Introduction

Many industries such as architecture and construction, computer games, animation, broadcast media, and manufacturing, make extensive use of feature-rich and complex design software (e.g., Revit, Maya, SolidWorks, AutoCAD, Adobe Creative Suite). These design software applications offer large feature sets, often numbering in the hundreds or thousands of commands. Designers, architects, engineers (design professionals, henceforth) must not only have the technical skills in using such feature-rich design software, but also need to understand specific workflows related to their domain of design. Furthermore, design professionals also have to keep up with software-related practices that are often established for use within specific organizations.

In the past, design professionals would often receive dedicated training on the feature-rich software they were expected to use in their careers [1]. However, as the rate of change in design software is rapidly accelerating, it is becoming difficult to receive comprehensive training in advance. Consequently, designers face the onus of learning and troubleshooting such software on-the-job. Moreover, with the ongoing development of new tools, features, and industry practices (such as the use of generative design tools that automate parts of the design process [2,3]), even the more experienced software users can benefit from learning about workflows that are more efficient than the ones they are currently using. The overall result is a need for design professionals to continuously develop and upgrade their software skill sets.

A number of innovations in human-computer interaction (HCI) have proposed ways of facilitating user interaction with feature-rich software and improving software learnability [4–13]. With the proliferation of online learning and troubleshooting resources, recent studies have also shed light on software help-seeking practices and use of resources such as tutorials, videos, and forums [14]. However, much of this work has been conducted in individual and self-directed recreational contexts of software use, where strategies such as "trial and error" [15,16] can be commonly used to achieve a desired goal. But, what about design professionals who use feature-rich software every day and often work on high-stakes commercial projects? How do they seek help and develop on-the-job software skills?

Prior works in HCI and computer-supported cooperative work (CSCW) have long suggested that colleagues play a key role in workplace help-seeking and information sharing (e.g., [17,18]). Indeed, today's communication and collaboration tools that are pervasive in modern workplaces offer several facilities for better connecting and interacting with co-located and remote colleagues. These include asynchronous and synchronous communication channels (e.g., Slack, Microsoft Teams), lightweight screen-sharing and video-conferencing tools (e.g., Skype for Business, Zoom), and document sharing and knowledge management tools (e.g., SharePoint, Confluence, Google Drive). But, we currently lack insights into if and how these modern tools are being used in software learning and help-seeking activities and whether or not they are actually useful. We argue that understanding how users are appropriating currently available tools in their day-to-day work environments will provide grounded insights to inform the next generation of innovations in software help-seeking.

This work was supported by a Mitacs Accelerate grant.

978-1-7281-6901-9/20/$31.00 ©2020 IEEE

In this paper, we investigate the practices that design professionals use to informally learn software skills on-the-job and how they interact with their colleagues during this process. Through semi-structured interviews with 20 design professionals across 8 small, medium, and large organizations, we provide a comprehensive analysis of on-the-job software learning and help-seeking episodes, focusing on the role of collaborative knowledge-sharing and remote communication.

Among our key findings, we found that most modern communication and collaboration tools failed in learning and help-seeking scenarios and that design professionals preferred in-person assistance. We also found that despite years of research and progress in improving remote communication and knowledge sharing within organizations, many of the recommended features were either not available or not used in the context of software learning and help-seeking. Furthermore, we observed that even though there was repetition and redundancy in the help-seeking needs among design professionals, there was little to no documentation or archival of help-seeking episodes, leading to knowledge sharing that was largely ephemeral, without opportunities for reuse.

The insights from our study offer several opportunities for design software to embrace and foster social means of helping design professionals adopt the know-how and best practices they need to get their work done and upgrade their skills with constantly evolving software features. In particular, our findings highlight the importance of further understanding and designing for workers who serve as "hubs of knowledge" and can have a dramatic impact on workplace help-seeking and learning.

Our paper makes the following contributions:

- provides insights into how design professionals learn and seek help for feature-rich software in today's workplace, highlighting the importance of in-person assistance, despite the presence of modern communication technologies.
- identifies opportunities for user-centered tools to address social forms of learning and help-seeking that can bring users closer to in-person types of assistance.

II. Related Work

To contextualize our findings, we draw upon prior literature on self-directed help-seeking practices and workplace learning and expertise sharing tools.

A. Studies of Self-Directed Software Help-Seeking Practices

Software learning and help-seeking has a long history in HCI. Past work has documented the many challenges that users face when learning to use feature-rich software and their associated help resources [14,16,19–23]. Prior work has also shown that it is common for users to learn subsets of functionality as dictated by the tasks they commonly perform [24,25]. Over the years there has been a proliferation of self-directed learning and help resources (e.g., manuals, online tutorials, videos, forums) [26,27] that can help end users resolve learning issues and challenges that they face with feature-rich software. However, it is well-established that most users are hesitant to spend dedicated time reading manuals or documentation as a source of help [15,28–30]. A key shortcoming of these resources is that they become quickly outdated as the software evolves [14]. Documentation can also lack coverage on the variety of methods and workflows of performing a given task, causing more confusion for end users [31]. Finally, users can experience difficulties locating relevant help from the wide pool of resources available online [14].

Although numerous HCI projects have aimed to improve the help-seeking experience of feature-rich application users (e.g., [4–13]), usually the focus has been on individual learning scenarios, with little to no focus on software help-seeking that occurs in the workplace. In addition to supporting individual help-seeking, recent work in HCI have also recognized the importance of social and crowdsourcing approaches that allow users to connect with one another as a means of accessing more targeted or personalized help (e.g., see [32–38]). These approaches have been studied through short deployments or laboratory studies, and do not offer insights into how people learn software on-the-job and how they approach help-seeking. Our work complements this existing body of research by providing insights on help-seeking interactions with different feature-rich applications across a variety of modern workplace settings (e.g., small to large companies) and disciplines (e.g., architecture, construction, broadcast media).

B. Workplace Learning and Expertise Sharing

HCI researchers have long demonstrated the importance of informal workplace learning [39–45]. Ackerman et al. [39] used the term "expertise sharing" to describe knowledge that is transferred between colleagues through discussion. Several studies have noted the challenges with expertise sharing, such as "blind leading the blind" [46], and in externalizing important knowledge before people retire or leave the company [47]. Prior studies have also shown that many workplace learning episodes are through interpersonal help-seeking among colleagues [48,49] or where employees serendipitously become aware of new tools through workplace interactions [50]. Furthermore, Twidale's work on over-the-shoulder-learning (OTSL) [18] explores situations where employees purposefully ask for technical help and prefer to see demonstrations from others.

A classic example of a tool that supports workplace collaboration and communication is *Answer Garden* [17,51,52] that was introduced in the early 90s. Answer Garden was designed as a Q&A repository for archiving knowledge in organizational settings. Pipek et al. [53] further studied the challenges with Answer Garden and the difficulties in the reuse of information. Similar themes were observed in other works, such as the Organization-Wide Learning (OWL) [54] system, another recommender tool for workplace knowledge sharing.

In addition, a number of collaborative tools that enable interpersonal workplace interactions (including for remote work) have been introduced and/or studied over the years, such as email lists [55], and synchronous chats [52,56–58]. Other studies have explored remote and distributed collaboration [59–62] and workforce perceptions and preferences of remote work [63]. Identifying who to ask for help has also been studied in the CSCW literature [40,64,65]. There have also been attempts for tools for automating the evaluation of expertise levels [66] and identifying experts in the workplace [67].

Many of these fundamental works, unfortunately, date back

prior to the prevalence of modern communication, collaboration, and online help-seeking tools. It is critical to revisit these topics to understand, what, if any, impact tools inspired by this body of research have had on software help-seeking behaviors in the modern workplace. Our research provides this updated insight by investigating how today's knowledge workers are actually utilizing these various communication tools to support their software learning in the workplace (if at all) and how their learning and help-seeking practices have evolved.

III. METHOD

We used a semi-structured interview approach and collected participant accounts of help-seeking and help-giving interactions during which a design professional needed to learn or troubleshoot a feature-rich application. These interactions could either be interpersonal (through a face-to-face conversation or through a remote communication channel, such as email or instant messaging) or non-personal and through the use of help resources such as videos, text tutorials, and forums.

A. Participants and Recruitment

To recruit a diverse range of design professionals, we reached out to companies representing different industries such as architecture and construction, media and entertainment, consumer goods. We recruited 20 participants (14M/6F) from 8 different companies (C1 to C8) using feature-rich applications on a daily basis (Table 1, C1-P01 means Participant 1 who works at Company 1).

The companies varied from startups designing wearable devices, with 1-60 employees all co-located, to long-established organizations in architecture with more than 5000 employees worldwide. We conducted one-on-one semi-structured interviews with participants who were frequent users of well-known and widely used complex feature-rich software. Out of these twenty interviews, seventeen were conducted in-person, at either the participants' workplace or an agreed-upon location. Three were coordinated over Skype because of the remote location of the participants.

TABLE 1. DEMOGRAPHIC DETAILS OF PARTICIPANTS

Company & Participant	Gender & Age	Job Title	Experience
C1-P01	M, 31-40	Technical Director	2-5 years
C2-P02	F, 19-24	Electrical Designer	1-2 years
C2-P03	M, 19-24	BIM (Building Information Modeling) Technician	2-5 years
C2-P04	M, 31,40	Mechanical Designer	2-5 years
C2-P05	M, 31-40	Mechanical Designer	1-2 years
C2-P06	M, 25-30	Project Engineer, Mechanical	2-5 years
C3-P07	M, 41-50	Architect	<1 year
C3-P08	M, 51-60	Senior Architecture Technologist	<1 year
C3-P09	F, 41-50	Architect	<1 year
C3-P10	M, 31-40	BIM Manager	2-5 years
C4-P11	M, 41-50	Director, Industrial and Mechanical Design	<1 year
C5-P12	M, 31-40	BIM Manager	2-5 years
C6-P13	F, 31-40	Architecture	2-5 years
C7-P14	F, 31-40	BIM Lead	2-5 years
C7-P15	F, 25-30	Geotechnical Designer	2-5 years
C7-P16	M, 25-30	Mechanical Designer	2-5 years
C7-P17	F, 25-30	Senior Structural Engineer	5-10 years
C7-P18	M, 31-40	Senior Technician	2-5 years
C7-P19	M, 25-30	Architect	5-10 years
C8-P20	M, 25-30	Principle Hardware Engineer	<1 year

We asked participants to assess their expertise level with their most regularly used feature-rich software. Participants reported a high degree of variation in their software expertise levels (novice, intermediate, advanced) and their learning approaches. Most of the participants (14/20) mentioned that they were novices when they first joined their current or previous company and learned their primary design software on-the-job by attending company-led workshops, use of online resources, trial and error, and asking colleagues for help. Only a few (3/20) participants mentioned learning their primary software through official courses and university programs. All participants had either improved in their expertise level or maintained the same level of expertise since joining their current company.

B. Semi Structured Interviews

Each interview session began with a brief pre-interview questionnaire, in which we asked the participants to provide demographic information about themselves, such as gender, age, educational background, current job position, how long they have been employed in the company, feature-rich design applications that they use, and the size of the team they regularly work with. We next asked our participants some warm-up questions about their job description and responsibilities. We inquired about their team, company, physical workspace and their software experience and training. We then asked participants to walk us through instances of peer-to-peer help-seeking and help-giving, their use of other resources to resolve their help needs, and the process of on-boarding and off-boarding in the company. Each session lasted approximately one hour. Following the interview, we asked participants to fill out a short questionnaire about their methods of help-seeking (peer-to-peer and the other external methods), the usefulness of these methods, and how regularly they rely on each method. All participants received $20 gift cards at the end of their interview.

C. Data Analysis

We used the ATLAS.ti software and a data-driven inductive analysis approach [68] consisting of open coding to identify initial themes. We were further used axial coding to refine the initial codes and analyze all of the interviews. The first author later used the revised coding scheme to analyze all twenty interviews and regularly discussed results with the other authors.

We analyzed each interpersonal help-seeking and help-giving episode according to a number of different dimensions, some of which were based on Twidale's 2005 work on OTSL [18]. For example, we identified the help-seeker, the type of help request (adopting and extending categorizations proposed in prior work [20,69]), the interpersonal help-seeking channel used (if any), the help-giver, whether the communication was synchronous or asynchronous, and the number of people involved. We noted whether the help episode was successful, whether it was archived, and how detailed the answer was. We also identified reasons for asking a particular help-giver (proximity, expertise level, team membership.).

IV. RESULTS

Based on our analysis of interviews, we distilled key results on how design professionals learned feature-rich software in the workplace, focusing in particular on their interpersonal help-seeking experiences.

A. *Overview of help requests and help resources*

We first provide an overview of the types of software-related questions that came up for design professionals and the types of help resources and communication tools that they used.

1) Types of help requests

A total of 43 instances of software help requests were described by our participants (24 help-seeking instances and 19 help-giving instances) that varied by topic and complexity. The questions ranged from being as simple as locating a feature in the user interface to more complex questions related to troubleshooting design models (summarized in Table 2).

In almost half of the help instances (19/43), the help-seekers described a need for procedural help. In some cases, they had a hunch that a feature was available within the software, but did not know the list of actions necessary to perform that task. Another class of help instances (13/43) was related to troubleshooting. In these cases, the participants explained how they knew the actions and appropriate commands necessary to perform a task, but had ended up in a non-deterministic state during which they could not solve the issue. In 8/43 of the help episodes, the help-seeker wanted to know if a particular workflow was possible with the software or whether or not a particular feature actually existed. The remaining help requests (3/43) were about confirming whether the help-seeker performed a task correctly or whether it adhered to company standards.

TABLE 2. TYPES OF DIFFERENT HELP REQUESTS PARTICIPANTS ELABORATED ON DURING THE INTERVIEWS

Types of Help Requests	Explanation and Example
Procedural Help (19/43)	When the help-seeker knows some tools exist for completing a task but does not know the procedure: P01: *"... for some sort of a chocolate milk commercial…I had to simulate this liquid pouring and that's actually one of the most complex things, even today in 3D animation. So, I banged my head against the wall for quite a few days and then eventually I couldn't get it done!"*
Troubleshooting (13/43)	When the help-seeker ends up in a non-deterministic state and cannot solve the issue on their own: P03: *"We had a model, everything was showing up, people started to sync that model, and the background on the mechanical side started to disappear…it was weird [...]Electrical [team] was not affected…was only a couple of mechanical drawings that had this issue."*
Determining Possibilities (8/43)	When the help-seeker wants to know what is possible or not possible with a feature-rich application: P20: *"I noticed that one of my coworkers had found a cool sub-menu that shows you the visual state of each part. and I just asked him, "Oh, that's cool. How did you do that?" And then, he showed me which sub-menus the option was hidden under."*
Getting Confirmation and Verifying Actions (3/43)	When the help-seeker wants verification or confirmation that standards have been met: P19: *"making sure that the model was at the right location […] that's always a tricky thing for us. So, often times we've had our BIM team set that up […] I like to have them check and lock it."*

For each of these types of help requests, design professionals had many ways of finding answers. We shed light on some of the resources they used, and the role of interpersonal help.

2) Use of help resources and inter-personal communication

From our interviews, we learned that design professionals had access to various help and learning resources either provided by the software vendor (e.g., Autodesk Design Academy) or internally within their team or organization (e.g., Pinnacle guided tutorials, internal courses on Moodle, Q&As over Yammer, internal manuals, documentation, and Wiki pages). Participants also elaborated on how they could also access thousands of online learning resources, such as tutorials and videos, (e.g., Lynda.com and YouTube), Q&A forums (e.g., StackExchange), and other text-based articles and discussions.

Based on the results of our post-interview questionnaires, the most frequently-used resources were forums (13/20) and video tutorials (12/20). Online text tutorials and company software documentation were used less often, but most people still used them a few times per month. The usage frequency of different sources of help among our participants is summarized in Fig 1. The majority of participants reported external resources to be useful (10-14/20 depending on the resource type) and indicated that company documentation was moderately useful (14/20).

Participants explained that most design companies today offer some form of mandatory or optional software-related group training and workshops. However, design professionals were concerned about the effectiveness of such training and workshops where a certain level of expertise was assumed:

P10: *They [workshops] approach it from a sort of a beginner, intermediate, advanced level. I don't think that that's how one should approach the software or the learning of it [...] There are certain things that once you simply demystify them slightly, they become very introductory. So, it really depends...*

Despite the plethora of self-guided software help resources available online today (forums, videos, tutorials, documentation), it was somewhat surprising to see how many design professionals still preferred to talk to someone. In fact, almost all participants (18/20) asked colleagues fo r help at least a few times per week, with more than half (11/20) saying that they asked questions a few times a day.

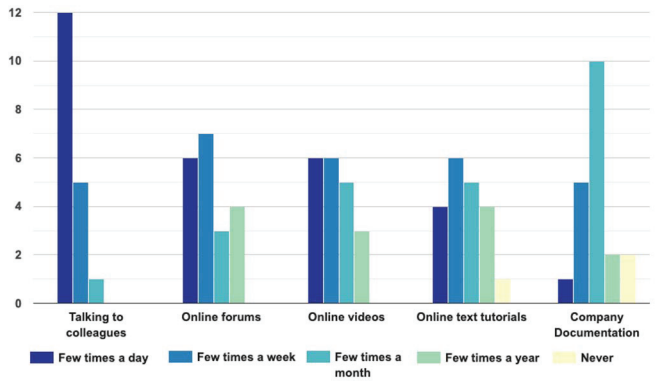

Fig. 1. Usage frequency of different sources of help among our participants

The type of help received from colleagues varied: in some cases (10/43) the help-giver fully performed the task for the help-seeker while in other cases (10/43) they only provided guidance on the necessary actions or gave a partial answer (10/43). In other instances (8/43), the help-seeker and the help-giver figured out the solution collaboratively. The remaining 5/43 episodes did not result in a clear resolution.

The two common ways for participants to learn and seek help from colleagues was through remote communication tools or through face-to-face conversations. We next discuss the difficulties of using remote communication tools for help-seeking and why design professionals still preferred in-person assistance.

B. *Difficulities in fully resolving help needs using remote communication tools*

As modern workplaces are increasingly interconnected with technologies, we learned that many colleagues were talking to each other using remote channels. Participants mentioned nine distinct communication tools in their interviews (summarized in Table 3). Some of these tools enabled users to share their models and designs while others were mainly used for real-time or synchronous conversations.

We found that about half of our participants (9/20) initiated their help-seeking episodes through a remote communication tool, such as instant messaging through Slack or Skype video conferencing or through an internal forum:

P15: *We also have an [internal] online forum... you can kind of post a question and people from the other offices all around the world will answer you and give you some help on how to do the task that you're trying to do…*

Participants described several advantages of using remote channels of help-seeking even when they were co-located, such as having more time to describe and understand the help request:

P10: *[Sometimes, co-workers] send me a message on Ryver [a messaging app]. That usually gives me some lead time to prepare a response, do a quick bit of research, so that we don't just sit there staring at their screen while they try to do it.*

However, we found that remote channels were eventually abandoned in the later stages of help-seeking. In fact, even though 15 out of 43 help-seeking episodes were initiated via a remote request, in most cases (10/15) the conversation shifted from being remote to becoming a face-to-face interaction:

P17: *We were using Skype, and [I] then sent him also like the context and what I tried and then asked him to come over to look at it. And then sometime later in the day when he had time, he came over to my computer and we played around with it...*

In the other 5/15 remote help-seeking episodes, face-to-face assistance was not an option because of the long distance between the help-seeker and the help-giver, and they struggled to resolve the issue using the same remote channel.

Our overall analysis revealed three key reasons why remote communication tools failed in the context of software help-seeking, as described below.

1) Difficulties in visually explaining software issues

The main reason that design professionals often gave up on these remote tools was the visual nature of help requests and answers when working with feature-rich design software, as has been seen other works [18]. Help-seekers struggled to find the appropriate vocabulary and language to explain something visually and receive visual explanations in return:

TABLE 3. REASONS WHY DIFFERENT REMOTE COMMUNICATION TOOLS FAILED IN SOFTWARE HELP-SEEKING SITUATIONS

Tool	Reasons for Failure
Email	Explaining the problem and actions necessary can be difficult and lengthy through text Overhead of attaching sscreenshots to complement text Emails can eventually get lost in inboxes
Phone call	Challenging to put all the instructions into words Cannot show visually how the actions should take place
Instant messaging (e.g., Slack, Ryver, Skype messaging)	Explaining the problem and actions necessary can be difficult and lengthy through text Overhead of attaching sscreenshots or remote desktop sessions to complement text Significant delays if the help-giver is not available for synchronous conversations
Video Conferencing (e.g., Skype, MS Teams)	Requires more overhead in planning video conferencing Can be difficult to coordinate screen sharing and pinpoint the exact issue in the software
Q&A Tools (e.g., Yammer)	Explaining the problem and actions necessary can be difficult and lengthy through text Overhead of attaching sscreenshots to complement text Some potential help-givers might not be following the Q&A stream where the question was asked Answers can get stale over time and not be relevant

P05: *If it's something you're not sure of, something you not sure how to use Revit for. It's very difficult to explain that on Skype or some kind of message sharing thing.*

P13: S*ometimes we'll just share a screen, or on Microsoft Teams we'll just chat, but then we'll have to do a screenshot and then sort of like a rough notes sketch over our screenshot of what we're trying to achieve, so it's just an added component because now we have to do a screengrab, and then try and sketch over it with our notes to actually show the other person what we're trying to achieve...*

2) Limited support for group expertise sharing

In 7 of the in-person help request instances, we found that multiple people were involved in the diagnosis and resolution (one instance showed that at least 10 colleagues were involved in the resolution). P01 gave an example of his experience:

P01: A few of us were working the same project at the same time…whether I was working on a 3D model and I wanted some opinion and the guy next to me was writing a piece of code to automate something and wanted my input, or whatever it was, [communication] was quite constant.

Even when designers were not working on the same project, we found that this type of constant communication was common in open workspaces where colleagues could overhear questions and answers and join the conversation. For example, P15 elaborated on one instance where a more senior colleague spontaneously joined a conversation:

P15: One of the senior managers who was quite experienced in automation with AutoCAD [gave] his input on what would be the most efficient way to solve the problem and what we had already kind of created and available that we could repurpose to use for this task.

In contrast to asking for help through co-located colleagues, remote tools offered limited functionality for supporting such

multi-way conversations and organic group expertise sharing. For example, end-to-end chats could not easily be shared with other team members, leaving out other help-seekers with similar questions and even other potential help-givers. And, as P10 explained, with multiple channels and conversations on the same topic, *"the information is disparate…it's just a mess."*

3) Lack of support for immediate help needs

Many help needs required an immediate answer and in some cases help-givers did not have time to completely resolve a request. In one case, a participant described how the help-giver had to first perform a complicated prerequisite task for resolving the issue at hand. Since the help-giver did not have time, the answer got severely delayed (P08: *"I had to wait maybe two days"*).

When the communication occurred over email, internal forums, or other asynchronous mediums, it not only took long, but the context could also get lost over time:

P14: *I can send an email to someone or I can post on an online forum and people might not have the same urgency or feel as motivated about a subject or want to find an answer with me as I do being on the ground being like I've got a real client sitting here and I've got real drawings and a real team that depend upon this answer*

All but one of the 43 help-seeking instances were synchronous. A common motivation for this was being able to ask follow-up, related questions:

P04: *If someone is explaining an answer to you, you know full well that when you get that answer you're going to ask about five more questions during them answering the question that you have.*

But, participants felt that sustaining such conversations remotely was more difficult and felt unnatural compared to talking to colleagues face-to-face.

C. *In-person help is most useful, but has a social cost*

Given the challenges of working with remote tools, most of our participants resorted to in-person help when available. In 31 out of 43 help-seeking instances, the help-seeker asked for help from a help-giver because of their higher level of expertise level or seniority at the company. In 12 instances, help-seekers asked their question from someone sitting at close proximity to them. In 10 instances, help-seekers identified a team member and 2 instances of their choice of help-giver was based on the context of their current and/or prior conversations:

P19: [I] think remote working is, is great, but there are certain tasks that are better when need that are just better when you have people in the same team. (e.g., Can you give me an example? construction drawings?). I think you need at least the beginning to be working in the same bay as people.

Participants often resorted to in-person assistance because they could not only get an answer to their question, but also benefit from tacit software knowledge sharing that is common between co-located colleagues [41,70]. Most of our participants appreciated learning from their colleague's expertise and experiences with specific features.

P17: Often times when you seek help from someone, a colleague, you end up having a discussion on the context surrounding the problem, and… that turns into a discussion about the past experiences or similar experiences [...] there's a bit of a branching and you end up talking about something that's related but also potentially really useful.

Another benefit of seeking help in person was that it was easier for participants to demonstrate their problem, which led to quicker mutual understanding and resolution:

P02: *Asking colleagues is faster because it's straight to the point, I can show them what I'm looking at and what I want. That's very straightforward.*

Despite the benefits of interpersonal help-seeking, design professionals gave several examples of how soliciting advice and tips comes at a social cost, as has been shown in other works [39,71]. For example, participants explained how they did not want to admit their lack of software knowledge nor did they want to appear incompetent or ill-informed in front of their colleagues:

P01: *A challenge of asking a colleague for help was that you don't want to help the people you don't know. You wanted to be careful about what you ask….sometimes, I would waste a lot of time online trying to find an answer to something that the guy next to me could easily tell me. But I didn't want to tell them that I don't know this or I have forgotten.*

The degree of social belonging to a company seemed to influence participants' tendencies to seek interpersonal help. For example, five of our participants had been at their current company for less than one year. Compared to other participants, these newer employees did not have the advantage of knowing everyone. They were also more motivated to prove themselves and make a good impression. These factors seem to affect their willingness to seek help from their colleagues. For example, P07, a recent hire at C3, had questions about the availability different feature-rich tools within Revit. Although he acknowledged that his collaborative workspace environment was positive, he was still hesitant to ask others:

P07: *I didn't find these [Revit features] [in] online [resources], maybe they have it, I don't know [...] I haven't checked that with my colleagues. It's something that I've done for myself [...] I didn't ask people for that, maybe I should ask!*

Even though it is common for employees to transfer practices and procedures to one another through word of mouth, our participants explained how this could also cause some issues down the road when the help-giver may be unsure or may in fact be following a "*bad practice*":

P10: *There have been times in the past where bad practices have kind of spread through teams unwittingly. That can cause clean up time in models [and] model slow down….*

D. *Hubs of knowledge are helpful, but overburdened with repetitive requests*

Given the social costs of interpersonal help-seeking, we learned that at many of the companies (5/8) were making an effort to centralize in-person knowledge-sharing. In fact, our participants who worked at these companies described the emergence of workplace "hubs of knowledge" who were employees designated to answer software-related and/or project-related questions. In fact, four of our participants played the role of a hub of knowledge within their organization and noted that it was part of their job duties to help other employees.

One advantage of having these hubs of knowledge was that design professionals could get on-the-fly feedback and

verification on the changes they had made, avoiding potential mistakes and pitfalls:

P02: *Because we have our mechanical and electrical [parts] of the same model, if I want to change, I don't want to affect the other people working on it as well. If I want to change, I'll send an email to the big manager, and then he asks everyone, is everything okay?*

But, despite the benefits of having access to these hubs of knowledge for help-seekers, we learned that these helpers sometimes became overburdened with frequent help requests throughout the day. For example, P03 mentioned that different colleagues asked him questions about "every hour and a half", every day. Furthermore, these hubs of knowledge were repeatedly asked the same questions, but they did not have any formal way of recording or re-accessing this information.

This observation resonates with prior research that has shown that there can often be a lot of overlap in software help-seeking Q&A [38,72] as people tend to face issues that other users have already experienced. Over time, several tools have been developed to support Q&A repositories and knowledge archives within companies. However, we learned that most help-seeking and learning-related knowledge was not formally archived by our participants. In fact, in only 7/43 of the cases, the question and/or the answer was recorded at all. In some cases (3/7), the information was temporarily saved within an instant messaging application through which the participant sought help. In other cases (4/7 episodes), participants took notes for their personal use (in either a paper-based notebook or a digitized application such as MS OneNote or Google Docs).

P17: *So I know a lot of people put things in OneNote, and I definitely have like a general OneNote in which I just toss things that are useful or interesting links or something like that.*

We also learned that designers would re- ask for help for the same question rather than make efforts to record the answer:

P09: *We do that all the time [re-asking colleagues] constantly… when you're doing color schedules, you can add the area, the types. I always forget where the drop-down is.*

Such behavior is similar to the concept of "production bias" [29] that has been used to explain why software users focus on figuring out the features they need just-in-time rather than investing in fully learning the application. Furthermore, not only did our participants not invest in learning, they also had no mechanism to reference the issue if it ever came up again. In more complicated cases where the help-giver fully performed the tasks, the help-seeker might not learn how to perform the task and could difficulties in re-performing those actions:

P19: *I've always had trouble with the geolocation stuff. Bringing in, making sure that the model was at the right location based on some CAD data [...] So, often times we've had our BIM team set that up. But it would be nice to have knowledge of that within the team. I think we're lacking in that sense. I struggle with it. So, I'm sure other people do.*

The lack of archiving was also problematic for the hubs of knowledge who were often dealing with repetitive questions and still had no process for archiving their conversations. So, why don't people bother to record and archive these help episodes and the knowledge that they gain? Above all, archiving this information for future use is time-consuming. As mentioned before, most design professionals were working in fast-paced environments and did not take time off their projects to save information for probable future use cases.

P20: *I mean, nobody is going to spend time documenting a bug on somebody else's software [...] we've got our own problems, right?*

The lack of archived knowledge was even more problematic when an employee left the company as they would take company-specific software knowledge with them:

P15: They [colleagues] spent a couple of days going through all of the project documentation and checking it and checking all of their assumptions…in the end, they came back and ended up with the original answer. But, it took a couple of days' worth of time and they weren't able to contact the original person that did the calculations…

Overall, participants explained that capturing and archiving all of the learning-related knowledge was a big challenge. We found an interesting paradox between the need for design professionals and companies to be able re-access knowledge and save time, but also not invest in formal practices to archive.

V. Discussion

Our study provides insights into how design professionals learn and seek help for feature-rich software in today's modern, tech-savvy workplaces. Although modern channels of communication have opened up new ways for knowledge workers to stay connected in the workplace, these remote channels largely fail in the context of learning and seeking help for software features. As a result, most remote help-seeking interactions turn into face-to-face conversations, but there are challenges in sustaining and archiving such information exchanges for future re-use.

We now reflect on the larger context of these findings and identify opportunities to design social forms of learning and help-seeking that can bring users closer to in-person assistance.

A. Larger Context of the findings

One insight from our study is that when knowledge workers are learning new software, their help requests can be multi-faceted and require immediate and verified answers. Although modern resources, such as videos, tutorials, and forums are popular for learning software, they do not capture company best practices or task-specific help needs. Compared to software developers [50], we found that the vocabulary problem appears to be even more acute for designers as their help requests are more visual in nature and difficult to articulate in natural language. Even when organizations have wikis, manuals, and other forms of internal documentation, either users do not want to take the time to read them or they do not know how to find them. In such cases, interpersonal help-seeking may be the only way for such users to obtain satisfactory and timely answers. In fact, we found that face-to-face conversations were still preferred and widely used, corroborating with findings from early 1990s research on workplace knowledge sharing [48,49]. What was particularly surprising was that even though remote communication tools have vastly improved in the last few years and allow knowledge workers to have easy access to their colleagues, these tools were eventually abandoned in the later stages of help-seeking episodes.

These findings are intriguing given decades of research and progress in facilitating remote communication and shared work.

Systems such as Answer Garden [17] were introduced in the early 1990s, but even thirty years later, the features advocated by such tools are not widely available or not used in practice. Although we have seen recent progress towards supporting the help needs of knowledge workers through in-context knowledge sharing and Q&A tools (e.g., [33,37]), there can still be barriers in establishing common ground. Furthermore, there is limited ability for multiple co-workers to "overhear" a conversation and organically expand the group discussion (which was deemed to be a key benefit of face-to-face interactions). We also found that the idea of "production bias" [29] still holds true as help seekers focus on increasing the throughput of their work and rarely spend time on archiving their knowledge for reuse.

We acknowledge that not all of the barriers and challenges that knowledge workers face can necessarily be solved by introducing yet another tool. However, it is also not feasible to replace everything with face-to-face conversations—these conversations are difficult to scale and do not work well in workplace arrangements where remote work in increasingly encouraged and even necessary. In this context, our findings point to several opportunities for design to foster more effective remote software help-seeking in interconnected workplaces.

B. Designing user-centered tools for "hubs of knowledge"

We learned that many organizations are now formalizing the concept of "hubs of knowledge" for internal knowledge sharing. Most design professionals acknowledged that this was a useful practice and that these hubs were playing an instrumental role in everyday help-seeking situations. In particular, having these dedicated hubs helped alleviate some of the social costs of relying on colleagues. However, few efforts in HCI have targeted design for these hubs of knowledge and the everyday support that they provide to software learners. Similar to prior research on expertise location in different domains [64,73,74], future work can explore the design of user-centered tools for identifying knowledge hubs, especially in large workplaces. In addition to automatically identifying a relevant software expert, it could also be useful to investigate ways to facilitate a subsequent interpersonal, face-to-face, help request with that user.

C. Designing in-context knowledge sharing and auto-archiving tools

Another opportunity for future work is to explore personal informal management tools that can help hubs of knowledge manage repeat requests. Based on our results, knowledge archiving in the context of software help was almost completely absent. Without adequate archival, knowledge hubs can be required to provide redundant help, which can be a waste of valuable time and expertise. Future work can explore automated systems to identify when a help seeking episode might have occurred and remind participants later in the day to archive their information exchange. Another consideration may be what should happen when a hub of knowledge leaves the team?

We also see a need for lightweight knowledge sharing tools that can allow users to express help needs and share knowledge in context. Some initial work in this area is trying to tackle this problem by building in-context memory aids (e.g., [38]), but such innovations do not take into account the multiple channels and tools that today's knowledge workers may be using on-the-job. We also need tools that support cross application help-seeking and knowledge transfer, such as RePlay [8], that support help needs with multiple tools. More HCI research is needed to better understand how we can make such knowledge sharing successful across different applications, platforms, and versions.

D. Supporting organic group expertise sharing

As noted in our results, many help-seeking episodes started with the use of a remote communication tool, but eventually ended up as a face-to-face conversation. In many cases, the key advantage of in-person help-seeking was that multiple experts could often chime into the conversation by serendipity. For example, in co-located scenarios, some employees could "overhear" conversations by other team members and help find a quicker resolution to an issue. Although many modern remote communication tools have provisions to allow people to participate in group chats or multi-channel conversations, most of our participants indicated that there were few opportunities for such organic group expertise sharing. Future work can explore designs that may support this type of organic group participation in the context of software help-seeking. In building tools that allow users to "overhear" and join an ongoing conversation, a key consideration may be how comfortable an employee may be with remote group notifications and interruptions.

E. Study Limitations

Our study has several limitations and some caution should be used in interpreting our results. For example, we used convenience sampling for this study and among our participants we had employees who could work remotely occasionally, but we did not have participants who were permanently remote. One future direction for more detailed analysis could be reaching out to remote employees. Future work could also look at people who are reskilling, coming back from retirement and/or changing jobs in different companies. Although our participants came from a variety of organizations, all of them were located in North America. Also, almost all of them were focused on using design software and the extent to which our findings generalize to all types of feature-rich software should be further investigated. Furthermore, we did not investigate individual personalities of help-seekers or help-givers or measure their preferences for different formats of help; our insights suggest that it would be worth studying such personalities and preferences in more detail and how they evolve over time in an organization.

VI. CONCLUSIONS

We carried out 20 interviews with design professionals to investigate how they learn and seek help for feature-rich design software in modern workplaces. We learned that designers rely heavily on interpersonal interactions and that modern remote communication tools are not always adequate in the context of seeking help. There needs to be an ongoing assessment of remote forms of communication and a better understanding of the long-term effects of introducing help-seeking innovations in the workplace. More broadly, there is an opportunity to re-consider and better support the ephemeral social help-seeking needs of knowledge workers as many of them struggle with learning and using different aspects of feature-rich software every day.

References

[1] L. Becker, "In our own hands: the democratization of graphic design," *University of California Press*, p. 26.

[2] K. Shea, R. Aish, and M. Gourtovaia, "Towards integrated performance-driven generative design tools," *Automation in Construction*, vol. 14, no. 2, pp. 253–264, Mar. 2005, doi: 10.1016/j.autcon.2004.07.002.

[3] "So. Algorithms Are Designing Chairs Now | WIRED." [Online]. Available: https://www.wired.com/2016/10/elbo-chair-autodesk-algorithm/. [Accessed: 01-Oct-2019].

[4] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," *Found. Trends Hum.-Comput. Interact.*, vol. 4, no. 2, pp. 81–173, Feb. 2011, doi: 10.1561/1100000009.

[5] D. K. Farkas, "The role of balloon help," *SIGDOC Asterisk J. Comput. Doc.*, vol. 17, no. 2, pp. 3–19, May 1993, doi: 10.1145/154425.154426.

[6] A. Fourney, B. Lafreniere, P. Chilana, and M. Terry, "InterTwine: creating interapplication information scent to support coordinated use of software," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2014, pp. 429–438, doi: 10.1145/2642918.2647420.

[7] C. A. Fraser, M. Dontcheva, H. Winnemöller, S. Ehrlich, and S. Klemmer, "DiscoverySpace: suggesting actions in complex software," in *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, New York, NY, USA, 2016, pp. 1221–1232, doi: 10.1145/2901790.2901849.

[8] C. A. Fraser, T. J. Ngoon, M. Dontcheva, and S. Klemmer, "RePlay: contextually presenting learning videos across software applications," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2019, pp. 297:1–297:13, doi: 10.1145/3290605.3300527.

[9] T. Grossman and G. Fitzmaurice, "ToolClips: an investigation of contextual video assistance for functionality understanding," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2010, pp. 1515–1524, doi: 10.1145/1753326.1753552.

[10] C. Kelleher and R. Pausch, "Stencils-based Tutorials: design and evaluation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2005, pp. 541–550, doi: 10.1145/1054972.1055047.

[11] N. Kong, T. Grossman, B. Hartmann, M. Agrawala, and G. Fitzmaurice, "Delta: a tool for representing and comparing workflows," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2012, pp. 1027–1036, doi: 10.1145/2207676.2208549.

[12] B. Lafreniere, A. Bunt, and M. Terry, "Task-centric interfaces for feature-rich software," in *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design*, New York, NY, USA, 2014, pp. 49–58, doi: 10.1145/2686612.2686620.

[13] S. Pongnumkul *et al.*, "Pause-and-play: automatically linking screencast video tutorials with applications," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2011, pp. 135–144, doi: 10.1145/2047196.2047213.

[14] K. Kiani, G. Cui, A. Bunt, J. McGrenere, and P. K. Chilana, "Beyond 'One-Size-Fits-All': understanding the diversity in how software newcomers discover and make use of help resources," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2019, pp. 340:1–340:14, doi: 10.1145/3290605.3300570.

[15] M. Rettig, "Nobody reads documentation," *Commun. ACM*, vol. 34, no. 7, pp. 19–24, Jul. 1991, doi: 10.1145/105783.105788.

[16] J. Rieman, "A field study of exploratory learning strategies," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 3, no. 3, pp. 189–218, 1996.

[17] M. S. Ackerman and T. W. Malone, "Answer Garden: a tool for growing organizational memory," in *Proceedings of the ACM SIGOIS and IEEE CS TC-OA Conference on Office Information Systems*, New York, NY, USA, 1990, pp. 31–39, doi: 10.1145/91474.91485.

[18] M. B. Twidale, "Over the Shoulder Learning: supporting brief informal learning," *Comput Supported Coop Work*, vol. 14, no. 6, pp. 505–547, Dec. 2005, doi: 10.1007/s10606-005-9007-7.

[19] O. D. Andrade, N. Bean, and D. G. Novick, "The Macro-structure of use of help," in *Proceedings of the 27th ACM International Conference on Design of Communication*, New York, NY, USA, 2009, pp. 143–150, doi: 10.1145/1621995.1622022.

[20] T. Grossman, G. Fitzmaurice, and R. Attar, "A survey of software learnability: metrics, methodologies and guidelines," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2009, pp. 649–658, doi: 10.1145/1518701.1518803.

[21] N. Hudson, B. Lafreniere, P. K. Chilana, and T. Grossman, "Investigating how online help and learning resources support children's use of 3D design software," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2018, pp. 257:1–257:14, doi: 10.1145/3173574.3173831.

[22] D. G. Novick, O. D. Andrade, and N. Bean, "The micro-structure of use of help," in *Proceedings of the 27th ACM International Conference on Design of Communication*, New York, NY, USA, 2009, pp. 97–104, doi: 10.1145/1621995.1622014.

[23] D. G. Novick, O. D. Andrade, N. Bean, and E. Elizalde, "Help-based tutorials," in *Proceedings of the 26th Annual ACM International Conference on Design of Communication*, New York, NY, USA, 2008, pp. 1–8, doi: 10.1145/1456536.1456538.

[24] S. Greenberg, *The Computer User As Toolsmith: The Use, Reuse, and Organization of Computer-based Tools*. New York, NY, USA: Cambridge University Press, 1993.

[25] B. Lafreniere, A. Bunt, J. S. Whissell, C. L. A. Clarke, and M. Terry, "Characterizing large-scale use of a direct manipulation application in the wild," in *Proceedings of Graphics Interface 2010*, Toronto, Ont., Canada, Canada, 2010, pp. 11–18.

[26] M. Lount and A. Bunt, "Characterizing web-based tutorials: exploring quality, community, and showcasing strategies," in *Proceedings of the 32Nd ACM International Conference on The Design of Communication CD-ROM*, New York, NY, USA, 2014, pp. 6:1–6:10, doi: 10.1145/2666216.2666221.

[27] "Wikipedia Statistics - Tables - English." [Online]. Available: https://stats.wikimedia.org/EN/TablesWikipediaEN.htm. [Accessed: 20-Sep-2018].

[28] J. M. Carroll, *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. Cambridge, MA, USA: MIT Press, 1990.

[29] J. M. Carroll and M. B. Rosson, "Interfacing Thought: cognitive aspects of human-computer interaction," J. M. Carroll, Ed. Cambridge, MA, USA: MIT Press, 1987, pp. 80–111.

[30] D. G. Novick and K. Ward, "Why don't people read the manual?," in *Proceedings of the 24th Annual ACM International Conference on Design of Communication*, New York, NY, USA, 2006, pp. 11–18, doi: 10.1145/1166324.1166329.

[31] X. Wang, B. Lafreniere, and T. Grossman, "Leveraging community-generated videos and command logs to classify and recommend software workflows," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2018, pp. 285:1–285:13, doi: 10.1145/3173574.3173859.

[32] A. Bunt, P. Dubois, B. Lafreniere, M. A. Terry, and D. T. Cormack, "TaggedComments: promoting and integrating user comments in online application tutorials," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2014, pp. 4037–4046, doi: 10.1145/2556288.2557118.

[33] P. K. Chilana, A. J. Ko, and J. O. Wobbrock, "LemonAid: selection-based crowdsourced contextual help for web applications," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2012, pp. 1549–1558, doi: 10.1145/2207676.2208620.

[34] V. Dziubak, P. Dubois, A. Bunt, and M. Terry, "Switter: supporting exploration of software learning materials on social media," in *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, New York, NY, USA, 2016, pp. 1209–1220, doi: 10.1145/2901790.2901827.

[35] B. Lafreniere, T. Grossman, and G. Fitzmaurice, "Community enhanced tutorials: improving tutorials with multiple demonstrations," in

Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, 2013, pp. 1779–1788.

[36] W. Li, T. Grossman, J. Matejka, and G. Fitzmaurice, "TwitApp: in-product micro-blogging for design sharing," presented at the Proceedings of the 24th annual ACM symposium on User interface software and technology, 2011, pp. 185–194, doi: 10.1145/2047196.2047219.

[37] J. Matejka, T. Grossman, and G. Fitzmaurice, "IP-QAT: In-product questions, answers, & tips," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2011, pp. 175–184, doi: 10.1145/2047196.2047218.

[38] L. Vermette, S. Dembla, A. Y. Wang, J. McGrenere, and P. K. Chilana, "Social CheatSheet: An interactive community-curated information overlay for web applications," *Proc. ACM Hum.-Comput. Interact.*, vol. 1, no. CSCW, pp. 102:1–102:19, Dec. 2017, doi: 10.1145/3134737.

[39] M. S. Ackerman, J. Dachtera, V. Pipek, and V. Wulf, "Sharing Knowledge and Expertise: the CSCW view of knowledge management," *Comput Supported Coop Work*, vol. 22, no. 4, pp. 531–573, Aug. 2013, doi: 10.1007/s10606-013-9192-8.

[40] M. S. Ackerman, V. Wulf, and V. Pipek, *Sharing Expertise: Beyond Knowledge Management*. Cambridge, MA, USA: MIT Press, 2002.

[41] Denise L. Peroune, "Tacit knowledge in the workplace: the facilitating role of peer relationships," *Journal of European Industrial Training*.

[42] S. Dennerlein, D. Kowald, E. Lex, D. Theiler, E. Lacic, and T. Ley, "The social semantic server: a flexible framework to support informal learning at the workplace," in *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business*, New York, NY, USA, 2015, pp. 26:1–26:8, doi: 10.1145/2809563.2809614.

[43] N. Hara, *Communities of Practice: Fostering Peer-to-Peer Learning and Informal Knowledge Sharing in the Work Place*. Springer Science & Business Media, 2008.

[44] J. Kooken, T. Ley, and R. de Hoog, "How do people learn at the workplace? Investigating four workplace learning assumptions," in *Creating New Learning Experiences on a Global Scale*, 2007, pp. 158–171.

[45] M. S. Schlager, C. Poirier, and B. M. Means, "Mentors in the classroom: bringing the world outside in," *Situated learning perspectives*, pp. 243–261, 1996.

[46] E. M. Trauth and E. Cole, "The organizational interface: a method for supporting end users of packaged software," *MIS Quarterly*, vol. 16, no. 1, pp. 35–53, 1992, doi: 10.2307/249700.

[47] D. Leonard and C. Myers, "Transferring knowledge between projects at NASA JPL (A)," Sep. 2016.

[48] A. Clement, "Cooperative support for computer work: a social perspective on the empowering of end users," in *Proceedings of the 1990 ACM Conference on Computer-supported Cooperative Work*, New York, NY, USA, 1990, pp. 223–236, doi: 10.1145/99332.99357.

[49] J. Rieman, "The Diary Study: a workplace-oriented research tool to guide laboratory efforts," in *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, New York, NY, USA, 1993, pp. 321–326, doi: 10.1145/169059.169255.

[50] E. Murphy-Hill, D. Y. Lee, G. C. Murphy, and J. Mcgrenere, "How do users discover new tools in software development and beyond?," *Comput. Supported Coop. Work*, vol. 24, no. 5, pp. 389–422, Oct. 2015.

[51] M. S. Ackerman, "Augmenting the organizational memory: a field study of Answer Garden," in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, New York, NY, USA, 1994, pp. 243–252, doi: 10.1145/192844.193019.

[52] M. S. Ackerman, "Definitional and contextual issues in organizational and group memories," *Information Technology & People*, Mar. 1996.

[53] V. Pipek and V. Wulf, "Pruning the Answer Garden: knowledge sharing in maintenance engineering," in *ECSCW 2003*, 2003, pp. 1–20.

[54] F. LintonEd.D., D. Joy, H.-P. Schaefer, and A. Charron, "OWL: a recommender system for organization-wide learning," *Journal of Educational Technology & Society*, vol. 3, no. 1, pp. 62–76, 2000.

[55] D. Constant, S. Kiesler, and L. Sproull, "What's mine is ours, or is it? a study of attitudes about information sharing," *Information Systems Research*, vol. 5, no. 4, pp. 400–421, Dec. 1994.

[56] E. Bradner, W. A. Kellogg, and T. Erickson, "The adoption and use of 'BABBLE': a field study of chat in the workplace," in *ECSCW '99: Proceedings of the Sixth European Conference on Computer Supported Cooperative Work 12–16 September 1999, Copenhagen, Denmark*, S. Bødker, M. Kyng, and K. Schmidt, Eds. Dordrecht: Springer Netherlands, 1999, pp. 139–158.

[57] W. Easley, "Understanding the experiences of youth in collaborative technical workplaces," in *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing*, New York, NY, USA, 2018, pp. 113–116, doi: 10.1145/3272973.3272987.

[58] W. Easley *et al.*, "Understanding how youth employees use Slack," in *Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing*, New York, NY, USA, 2018, pp. 221–224, doi: 10.1145/3272973.3274060.

[59] P. Bjørn, M. Esbensen, R. E. Jensen, and S. Matthiesen, "Does distance still matter? Revisiting the CSCW Fundamentals on Distributed Collaboration," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 21, no. 5, p. 27, Nov. 2014, doi: 10.1145/2670534.

[60] G. M. Olson and J. S. Olson, "Distance matters," *Hum.-Comput. Interact.*, vol. 15, no. 2, pp. 139–178, Sep. 2000.

[61] G. M. Olson, A. Zimmerman, and N. Bos, Eds., *Scientific collaboration on the Internet*. Cambridge, Mass: MIT Press, 2008.

[62] J. S. Olson and G. M. Olson, "How to make distance work work," *Interactions*, vol. 21, no. 2, pp. 28–35, Mar. 2014, doi: 10.1145/2567788.

[63] K. Eason, *Information Technology and Organisational Change*. Bristol, PA, USA: Taylor & Francis/Hemisphere, 1988.

[64] K. Ehrlich and N. S. Shami, "Searching for expertise," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2008, pp. 1093–1096, doi: 10.1145/1357054.1357224.

[65] D. W. McDonald and M. S. Ackerman, "Just talk to me: a field study of expertise location," in *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, New York, NY, USA, 1998, pp. 315–324, doi: 10.1145/289444.289506.

[66] J. Zhang, M. S. Ackerman, L. Adamic, and K. K. Nam, "QuME: a mechanism to support expertise finding in online help-seeking communities," in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 2007, pp. 111–114, doi: 10.1145/1294211.1294230.

[67] T. Reichling and V. Wulf, "Expert recommender systems in practice: evaluating semi-automatic profile generation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2009, pp. 59–68, doi: 10.1145/1518701.1518712.

[68] A. Strauss and J. M. Corbin, *Basics of qualitative research: Grounded theory procedures and techniques*. Sage Publications, Inc, 1990.

[69] A. Sellen and A. Nicol, "Building user-centered on-line help," in *Readings in Human–Computer Interaction*, R. M. Baecker, J. Grudin, W. A. S. Buxton, and S. Greenberg, Eds. Morgan Kaufmann, 1995, pp. 718–723.

[70] E. A. Smith, "The role of tacit and explicit knowledge in the workplace," *Journal of Knowledge Management*, Dec. 2001.

[71] P. K. Chilana, A. J. Ko, J. O. Wobbrock, and T. Grossman, "A multi-site field study of crowdsourced contextual help: usage and perspectives of end users and software teams," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2013, pp. 217–226, doi: 10.1145/2470654.2470685.

[72] W. Jones, *Keeping Found Things Found: The Study and Practice of Personal Information Management: The Study and Practice of Personal Information Management*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2008.

[73] I. Becerra-Fernandez, "Searching for experts on the web: A review of contemporary expertise locator systems," *ACM Trans. Internet Technol.*, vol. 6, no. 4, pp. 333–355, Nov. 2006.

[74] S. A. Paul, "Find an expert: designing expert selection interfaces for formal help-giving," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2016, pp. 3038–3048, doi: 10.1145/2858036.2858131.