

# Understanding Collaborative Practices and Tools of Professional UX Practitioners in Software Organizations

K. J. Kevin Feng  
University of Washington  
Seattle, USA  
kjfeng@uw.edu

Tony W. Li  
University of Washington  
Seattle, USA  
tonywli@uw.edu

Amy X. Zhang  
University of Washington  
Seattle, USA  
axz@cs.uw.edu

## ABSTRACT

User experience (UX) has undergone a revolution in collaborative practices, due to tools that enable quick feedback and continuous collaboration with a varied team across a design’s lifecycle. However, it is unclear how this shift in collaboration has been received in professional UX practice, and whether new pain points have arisen. To this end, we conducted a survey ( $N = 114$ ) with UX practitioners at software organizations based in the U.S. to better understand their collaborative practices and tools used throughout the design process. We found that while an increase in collaborative activity enhanced many aspects of UX work, some long-standing challenges—such as handing off designs to developers—still persist. Moreover, we observed new challenges emerging from activities enabled by collaborative tools such as design system management. Based on our findings, we discuss how UX practices can improve collaboration moving forward and provide concrete design implications for collaborative UX tools.

## CCS CONCEPTS

• **Human-centered computing** → **Collaborative interaction**;  
**Empirical studies in interaction design**; **Empirical studies in collaborative and social computing**.

## KEYWORDS

user experience, collaboration, design practice

### ACM Reference Format:

K. J. Kevin Feng, Tony W. Li, and Amy X. Zhang. 2023. Understanding Collaborative Practices and Tools of Professional UX Practitioners in Software Organizations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*, April 23–28, 2023, Hamburg, Germany. ACM, New York, NY, USA, 20 pages. <https://doi.org/10.1145/3544548.3581273>

## 1 INTRODUCTION

Today, the field of user experience (UX) is viewed as a cornerstone in the human-centered design of interactive systems [44, 80, 81]. UX typically refers to principles and practices that enhance an end-user’s experience with products, services, and companies [82], and it has seen rapid growth in both the number of practitioners in

the field [80] as well as adoption within a wide range of organizations [44]. Interest in UX jobs saw a 289% increase in 2020 alone [111], while more companies are reaching a sufficiently high level of UX maturity to hire managers to lead entire UX teams [80]. Furthermore, UX practitioners (UXPs) are increasingly working with technical, non-UX stakeholders such as developers and data scientists, as the incorporation of complex technologies such as machine learning necessitates closer and lengthier UX collaboration [120].

Alongside this growth in size and stature, the field has seen a major shift towards collaborative UX tools [89]. Popular tools now boast robust features for synchronous and asynchronous collaboration as a core product offering [2, 33, 35, 77, 101] to enable teams of UX practitioners (UXPs) to work together on UX challenges of higher complexity and scale. It was not always this way. Sketch, a Mac-only interface design tool that dominated the field throughout much of the 2010s [58], was once a native app that forced UXPs to work locally and share files through email. In 2016, Figma released its browser-based interface design tool enabling real-time collaboration on a shared canvas [33], and other tools (including Sketch) soon followed [2, 100]. In addition to eliminating the versioning and synchronization issues that inevitably arose from working locally and manually sharing design files, these tools revolutionized collaboration among UXP teams by opening up new potential for collective ideation and critique, as well as tighter feedback loops. More broadly, they democratized UX for non-UXPs and even nurtured a community of UXPs to share reusable templates and tool extensions [30].

These shifts lead us to consider whether there may be new challenges and needs for UXPs engaged in collaboration in light of the major changes in collaboration practices of UXPs and changes in collaborative tools. Prior work suggests that increased collaboration with technical stakeholders may exacerbate UX challenges such as conceptual gaps and coordinative breakdowns during periods of handoff [71, 121]. While this line of work surfaces important pitfalls, those pitfalls are not discussed alongside the recent additions of collaborative capabilities in common tools, which have since become an integral part of UX practice [89]. A limited set of previous work has catalogued the changing landscape of UX tools in recent years [58, 89] and how collaborative features impact tool adoption [103] or support specific tasks such as ideation [50]. These works, however, do not provide a high-level overview of the diverse challenges UXPs may face within such tools. Additionally, besides user complaints in tool forums [1], there has been little empirical work exploring new challenges induced by collaborative tools. Literature outside of UX provides strong indication that such challenges exist—prior work outlined forms of coordinative friction that arose with the introduction of collaborative document editors



This work is licensed under a Creative Commons Attribution International 4.0 License.

CHI '23, April 23–28, 2023, Hamburg, Germany  
© 2023 Copyright held by the owner/author(s).  
ACM ISBN 978-1-4503-9421-5/23/04.  
<https://doi.org/10.1145/3544548.3581273>

[118] and code editors [40]. Yet, it would be unwarranted to generalize these findings to UX given the prevalence of domain-specific representations and processes [71, 120].

Despite the rise in both UX teams and UXP and non-UXP collaboration, there is surprisingly little empirical understanding of how these collaborations play out among different roles across stages of the design process. This understanding is vital in a post-COVID-19 era. As organizations and employees rapidly adopt hybrid work models [4], disparate roles are likely to be physically distributed and their work will increasingly be mediated by collaborative tools. Indeed, usage of Figma and Miro increased sharply throughout the COVID-19 pandemic and remain at unprecedented levels [88, 89]. New workflows may have also emerged from usage of collaborative tools that pre-pandemic literature may fail to capture. Thus, the time is ripe to obtain an updated understanding of how UXPs collaborate, the tools they use, and in particular how these two intersect and influence each other. We pose 4 main research questions with these goals in mind:

- **RQ1:** With whom do UXPs collaborate at each stage of the design process?
- **RQ2:** What tools and tool-based collaboration strategies are UXPs using throughout their workflows?
- **RQ3:** What practices and challenges arise when UXPs hand off their work to software developers and ML practitioners?
- **RQ4:** What are UXPs' current expectations and practices around reusing design components and managing design systems in collaborative UX tools?

In this paper, we answer these questions by conducting a descriptive survey with 114 UXPs employed at software organizations—organizations that actively maintain and/or build new software—based in the U.S. to understand their collaborative practices, tools, and relationships between the two. We use a survey as our research instrument of choice due to its ability to simultaneously capture high-level trends and probe into specific practices, as demonstrated by prior survey-based work [88, 89, 97, 123]. We use the term “tool” to specifically refer to software applications (e.g., Adobe Illustrator), software artifacts (e.g., code, spreadsheets), and physical objects (e.g., pen and paper) that aid and enhance UXPs' work. We qualitatively analyzed free responses from our survey to extract themes and insights, tallied categorical responses, and applied quantitative methods to identify significant differences where relevant. A summary of our most salient findings is as follows:

- (1) UXPs collaborate with a wide range of roles, but most frequently with other UXPs and product managers. However, UXPs will typically be the ones to initiate collaborations with non-UXPs.
- (2) Most UXPs primarily use a single, all-in-one collaborative UX tool such as Figma to perform functions and tasks across all stages of the design process.
- (3) UXPs use text-based strategies, such as comments and on-design annotations, while sharing their work with collaborators but often prefer verbal methods such as meetings and voiceovers. UXPs also employ more visuals and storytelling techniques when presenting their work to non-UXPs compared to fellow UXPs.

(4) Handoffs remain a tricky challenge for UXPs due not only to persistent problems around knowledge gaps and design-development divergence, but also problems unique to collaborative UX tools such as overcommunication of irrelevant design details to developers.

(5) While greater collaboration on design systems within collaborative UX tools has popularized sharing and reusing design components, new issues around design system management have arisen that may exacerbate coordinative breakdowns: ambiguity of ownership, lack of communication about changes, overly frequent changes, and more.

Through the lens of our findings, we reflect on the state of collaborative practices of UXPs in light of major shifts in tools in the last several years. We argue that today's UXPs use tools that act as a connective “hub” of activity across an entire team, where cross-functional stakeholders can collectively manage information from a single workspace. Yet, many challenges when working with technical stakeholders still persist, and new sources of friction emerge from activities made possible by this collaborative shift, such as management of scalable design systems. We conclude by recommending advancements in collaborative techniques and tools that do not merely *offer* opportunities for collaboration but also *enrich* collaborative experiences within and around UX.

## 2 BACKGROUND AND RELATED WORK

Collaboration in UX draws upon a multitude of traditions in computer-supported collaborative work and human-centered design. Here, we review relevant past works in collaboration among (non-UX) designers, UX practices, and collaborative tools to better set the stage for our work.

### 2.1 Collaboration Among Studio Designers

Researchers have been interested in how designers collaborate well before UX was popularized as a field. Works that did not consider software-based UX focused primarily on architectural and industrial design studio environments [14, 22, 27, 95, 114], as the studio was considered central to design work and education [83, 114]. As computers made their way into design studios, tradeoffs surfaced between physical and remote software-based collaboration [14]. Indeed, the lack of interactivity in early computing tools presented obstacles to digitally communicating sophisticated designs and prompted further research on how software tools can help with these obstacles [17, 22, 59, 66], including virtual asynchronous design studios [17, 59]. Years later, when virtual work practices may be forced upon many in times such as during the COVID-19 pandemic, Lee et al. [66] showed that designers considered a distributed collaboration workflow to be more engaging than a face-to-face one because they were forced to put more thought into communicating effectively. Though the digitization of studio workspaces is not met without resistance, it unlocks flexible, accessible, and efficient collaboration workflows for designers [52, 54].

While many studies of studio design collaboration were conducted just as computers were becoming mainstream office tools, their insights can be seen integrated into modern productivity tools. Multimedia access for instructional content explored by ID-Online [17] is widely available in Canvas [51], a popular class management

system used by universities worldwide. Separation of workspaces and discussion areas is available in general-purpose project management software such as Jira [7] and Asana [6]. Shared databases with asynchronous contribution tracking, such as the one implemented by Kolarevic et al. [59], are now ubiquitous in cloud-powered productivity hubs such as OneDrive [76] and Google Workspace [41].

Although modern collaboration software appears to address some past desiderata, it is unclear whether they result in new collaborative challenges of their own. Additionally, collaborative practices have been studied extensively in traditional studio settings, but those same practices may not be preserved in contemporary UX workflows for software interfaces. We tackle these questions in our study.

## 2.2 Multi-User Collaborative Authoring Tools

Tools that enable collaborative authoring, particularly ones that allow simultaneous multi-user interactions, have captured the interests of researchers in diverse domains, including software engineering [12, 38, 40, 46, 113, 119], data science [72, 90, 102, 116, 117, 123], document editing [39, 55, 57, 65, 109], entertainment [69, 98], and geospatial analysis [29], many of whom observed novel or unforeseen issues arise from the collaborative features. For example, Goldman et al. built a collaborative, web-based Java IDE called Collabode [40], but found upon evaluation that programmers' code produced runtime errors for collaborators despite it being free of compilation errors. Wang et al. [118] conducted an interview study ( $N = 30$ ) to investigate rationales for writing practices using collaborative writing tools such as Google Docs. They found that despite the rich suite of co-editing features, users were reluctant to fully embrace them due to concerns of accountability, credit of contribution, and loss of privacy. In general, dynamics of performing tasks in collaborative tools may be different than performing those same tasks in an isolated environment [28] and deserve scrutiny during evaluation.

While prior work in this area yields rich findings, there are still numerous unanswered questions when it comes to collaboration in the UX domain. Findings in non-UX domains are not guaranteed to generalize to UX work given representational and procedural differences [71, 120], while work studying collaborative UX tools are remarkably limited, possibly due to the tools' recent adoption. Indeed, when Jung et al. [55] studied small-team remote collaboration in design challenges, they used Google Docs as their tool of choice due to its overwhelming popularity at the time rather than a specialized design tool such as Figma. Furthermore, prior evaluations of collaborative tools primarily probed practitioners' experiences working with others in their own domain [40, 55, 57, 116, 117]. However, studies of UX in professional settings revealed that UXPs frequently interface with colleagues outside of UX [43, 44], as small or even single-person UX teams were common. We account for this by investigating both inter- and intra-domain collaboration in our survey.

UX tools in industry have charged forward with releasing collaborative features, despite little academic work in this area. Figma [33] is a browser-based UI design and prototyping tool that allows multiple users to work in real-time and has gained significant popularity since its launch in 2016 [89]. Its popularity led to native

tools such as MacOS's Sketch [100] and Adobe XD [2] also enabling real-time collaboration. Social affordances for co-presence, such as multiple cursors on a single canvas, are also popular in other whiteboarding and ideation tools, such as Miro [77] and Mural [108], as well as platforms to share community assets, such as Figma's Community [30]. Interestingly, community-building in design tools has been shown by previous work to be a catalyst for tool adoption among practitioners [103]. These collaborative tools have enabled now-common UX practices around design system management [24, 78], where modular design components can be reused across an entire organization to ensure consistency, and developer handoffs [13], where the designer passes their work to developers for code implementation. Consequently, design systems have been further formalized. Moore et al. [78] identified three core components of design systems—a *design philosophy* for establishing an overall vision for the system, *interaction patterns* that compose the system's core user experience, and *content format* that helps generate in-system content (e.g., images, data formats) for particular use cases. Despite this formalization, there has been no work, to our knowledge, investigating UXPs' experiences of collaboratively managing design systems.

Given previous literature and industry trends, attention to tool-based collaboration support is clearly high. The landscape of UX tools has shifted such that collaboration is now placed at the forefront of their experiences, but we do not have an updated understanding of how this shift influences the way designers work, or what new collaborative friction, if any, has arisen from new collaborative features. Unforeseen circumstances that demand high usage of them—such as a global pandemic—may also contribute to the current gap in literature that directly tackle these questions. We bridge this gap by capturing both a high-level understanding of the current status quo in UX, as well as specific UX practices where increased collaboration may be contentious. While we could have conducted an interview study, scaling up the number of participants was important for us to obtain a higher-level view of the field, so we opted for a survey. However, we still wanted participants to share specific, nuanced experiences with us, so we designated many survey questions as open-ended responses. The resulting *descriptive survey* [70] thus became our study instrument of choice.

## 2.3 Characterizing UX Work

The growth of UX as a field [80] has prompted researchers and practitioners alike to more formally characterize UX work. Below, we review some relevant works that helped inform the *lingua franca* of modern UX, which we leverage in our survey.

The British Design Council popularized the double diamond design process in 2004 and describes it using 4 distinct phases: Discover, Define, Develop, and Deliver [11]. The double diamond has been adopted as a standard UX workflow by practitioners in industry and beyond [25, 45, 94, 120]. Activities commonly associated with each phase of the process are listed in Table 1.

Although not all UX workflows may follow the double diamond, it is a widely adopted framework with distinct activities and methods across its stages; hence, we take inspiration from it to scaffold our analysis. To make the stages more concrete for participants, we mapped its stages to 5 common categories of UX activities [10] that

Phase	Description	Common Activities
Discover	Comprehending and questioning the design problem and understanding needs of potential users.	Conducting interviews, questionnaires, field observations
Define	Converging possibilities through interpreting and aligning research findings to project objectives.	Synthesizing data from interviews, finalizing the design problem, defining personas
Develop	Exploring multiple directions with the goal of revealing attributes of an effective solution.	Sketching wireframes, conducting usability tests, iterating on low-fidelity prototypes
Deliver	Presenting a single solution and preparing it for launch and/or further iteration	Creating high-fidelity prototypes, writing pitch documents, publishing user stories

**Table 1: Outline of phases in the double diamond design process.**

loosely correspond to these stages—research, synthesis, low-fidelity design, high-fidelity design, and communication. Our mapping is described in Fig. 1, where we use a gradient to show loose correspondence.

The increasing rate of UX adoption within companies [80] has led to studies of UX in organizational contexts. Gray et al. [44] noted that despite companies’ enthusiasm to embrace UX, company culture can still be hostile to the consideration of new UX perspectives, which leads to frustration from UX designers. A select number of works examine topics in UX collaboration with particular interest in integration with technical stakeholders through designer-developer handoffs [67, 71, 115], Agile software development processes [56, 61, 68, 86], and collaboration with data scientists and AI engineers [105, 120], in addition to other UX practitioners for research synthesis [60]. Frishberg and Convertino argue for the importance of UX collaboration with non-UX parts of the organization, and recognize collaborative differences in UX teams situated in corporate departments versus design agencies [36]. We employ this observation in our work by capturing information about how participants’ UX teams are situated within their larger organization, while excluding design agencies from our scope.

UX’s proliferation as a professional practice also led to an expansion of UX-related roles. As researchers, roles—often communicated through job titles—provide us with crucial information on the nature of UX work participants engage in, as well as whether they work in UX in the first place. Indeed, contemporary UX job titles can reflect ownership of different areas of the double diamond process [64, 91], but may vary based on organization [43]. That said, many UX titles are still used fluidly and are ambiguous in their roles—these include Interaction Designer, Experience Architect, UX/UI Front End Developer, among others [64]. Even non-UX workers, such as software engineers, are beginning to take on select parts of the design process themselves if it is closely tied to their work [44]. We take this discourse on job titles into consideration in our survey by allowing participants to self-select their job function from 5 higher-level categories we observed from aforementioned work and recent UX job postings rather than aggregating by participants’ job titles. Those 5 categories are research, design, engineering, writing, and other.

Despite current literature on UX, we still lack an updated understanding of UX practitioners’ collaboration practices that holistically consider organizational factors, tools, and workflows. Changes in workplace culture due to the COVID-19 pandemic mean that some factors traditionally fostering collaboration, such as sitting next to each other [53], may be more demanding to implement. Furthermore, it is not clear from previous work how UX designers collaboratively move between tasks in the double diamond design process, particularly in larger organizations where a designer may only work in a specific phase of the process. Our study serves to shed light on these ambiguities.

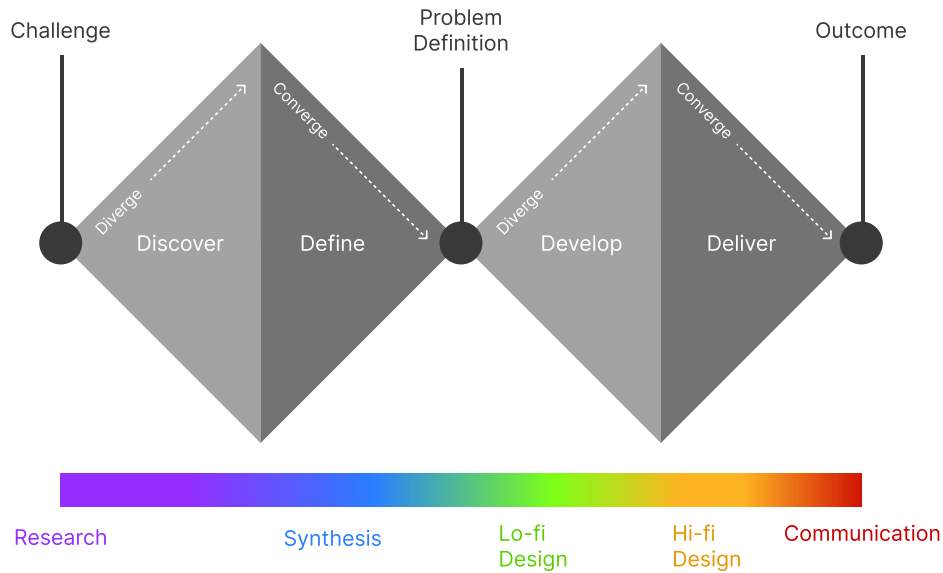
### 3 METHOD

The goal of our study is to explore collaboration patterns and tool use in UXPs working on software products. To accomplish this, we deployed a survey hosted on Qualtrics<sup>1</sup> to UXPs working in U.S.-based organizations that actively created and maintained software. We focused on U.S.-based organizations to limit variations in regional UX practices and definitions that may permeate organizations with differing “home bases” [62]. While prior work investigated collaborative behaviours between industry UXPs in specific parts of the design process [50, 60, 84], we observe collaboration patterns across the entire UX lifecycle, taking into consideration roles, team size, and common tools used.

#### 3.1 Survey Questions

Our survey questions were informed by surveys from prior work that probed practitioners’ technical practices within technology organizations [47, 116, 123], as well as the authors’ recent experiences working with UX practitioners in technology organizations. We also discussed our survey questions with two UXPs from our personal networks to refine or remove existing questions and add new ones. The survey first collected some basic demographic information such as gender, years of experience working in UX and in their organization more broadly, UX team size and structure, and the stage(s) of the design process they were involved in. Participants identified a UX role they associated with most closely in their project from our list of 5 roles—designer, researcher, engineer, writer, and other.

<sup>1</sup><https://www.qualtrics.com/>



**Figure 1: Common categories of UX activities mapped onto the double diamond design process [11].**

These UX roles were compiled by considering prior work in UX job titles [64, 91] and industry trends, such as the emergence of the UX writing role [26]. We also defined a set of non-UX roles common in software organizations based on a similar list by Zhang et al. [123]. The non-UX roles are product manager, software developer, data scientist, domain expert, and sales/marketing/finance. Participants were then invited to recall a project they previously completed. With that recollection in mind, they answered multi-select questions on who they collaborated with (both UX and non-UX roles) throughout their UX workflows, along with free-response questions about tools, strategies, and challenges.

Depending on whether participants created designs themselves (as opposed to just conducting research or writing code), they were asked a series of design-specific questions, from design system management, to developer handoff, to design environment customization. We focused on these areas as previous work indicated that these may be locations of uncertainty and coordinative tension in the UX design process [13, 24, 87]. We piloted the survey with members of our lab, which led us to modify the survey structure and clarify terminology in the questions based on feedback. Our full survey is available in our supplementary materials.

### 3.2 Survey Distribution and Participants

We sent our survey to our department and department alumni Slack channels related to HCI and design. We reached out to our personal connections in U.S.-based technology companies and asked them to pass along the survey to UXPs within their organization. We also posted the survey to design UX-related social media groups on Discord, LinkedIn, Facebook, Reddit, Blind, and Twitter (both on Twitter Communities as well as from authors' personal accounts). We began recruiting in late February 2022 and accepted our last response in early August 2022, when we reached data saturation. We leveraged snowball sampling by implementing a referral system

where participants received an additional entry to our gift card raffle for every participant they referred who also completed the survey.

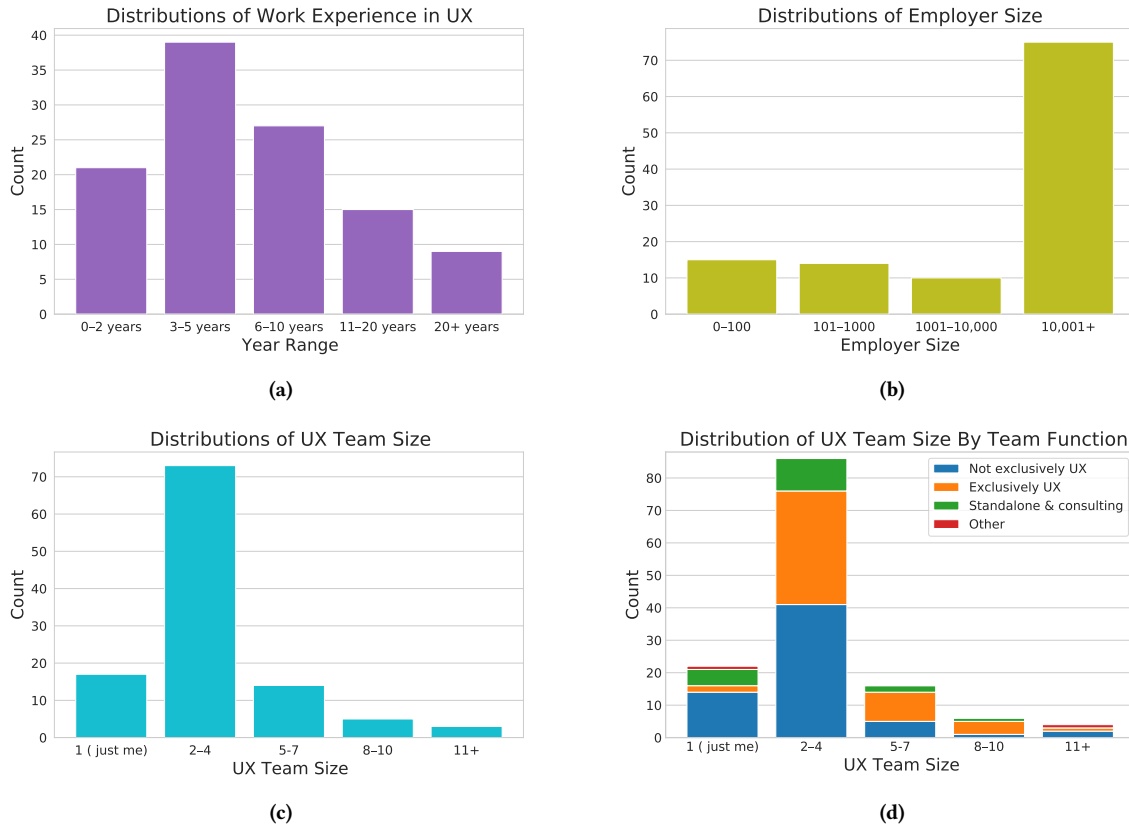
We received valid survey responses from 114 participants. Responses were considered valid if the participant held a UX-related job title and self-identified as one of five UX roles we derived from previous academic and industry literature [64, 91, 99]. All participants who contributed valid responses were entered into a raffle to win one of five \$50 USD Amazon gift cards.

Out of our 114 participants, 59% were female, 39% were male, 1% were non-binary, and 1% preferred not to disclose. Most were early- to mid-career (see Fig. 2a for a detailed breakdown of years of experience in UX). Most were employed at large organizations with over 10,000 employees (see Fig. 2b). The UX teams in which our participants worked tended to be fairly small, with most teams consisting of 2–4 people (Fig. 2c). 38% of participants' UX teams were situated within a larger team that did not exclusively do UX, 32% were within a larger UX-only team, 10% were standalone and consulted for non-UX teams, and the rest were some combination of at least two of the three. Participants were mostly UX designers (74%); besides that, 16% were UX researchers, 4% were UX writers, 3% were other (UX managerial roles), and the rest declined to answer. We had no participants who self-identified as UX Engineers.

We observed no significant correlations between participants' organization size and UX team size ( $\chi^2(12) = 9.47, p = 0.66$ ), but noticed that larger UX teams tended to be ones that focused exclusively on UX (Fig. 2d). Single-person teams were mostly situated within a non-UX team.

### 3.3 Analysis

We took both quantitative and qualitative approaches to analyze our survey data. For quantitative analysis, we performed statistical



**Figure 2: Distributions of participants' (a) work experience, (b) employer size, (c) UX team size, and (d) UX team size by team function. Note that (c) and (d) do not share the same y-axis since participants can select more than one team function in (d).**

tests (e.g., Chi-squared test of independence, Friedman test) to determine if there were statistically significant associations between participant groups and/or stages in the design process. For well-structured free response questions, such as ones where we asked participants to list tools they used, we converted the responses into lists of categorical variables for statistical testing. We also qualitatively classified such responses into higher-level categories where applicable. For less-structured free responses, such as when we asked participants to share challenges in adopting design systems, we compiled all the responses to a particular prompt into one document. One author then took two passes through the data, first performing an open coding procedure to capture meaningful statements, and then organizing the results into broader themes through axial coding.

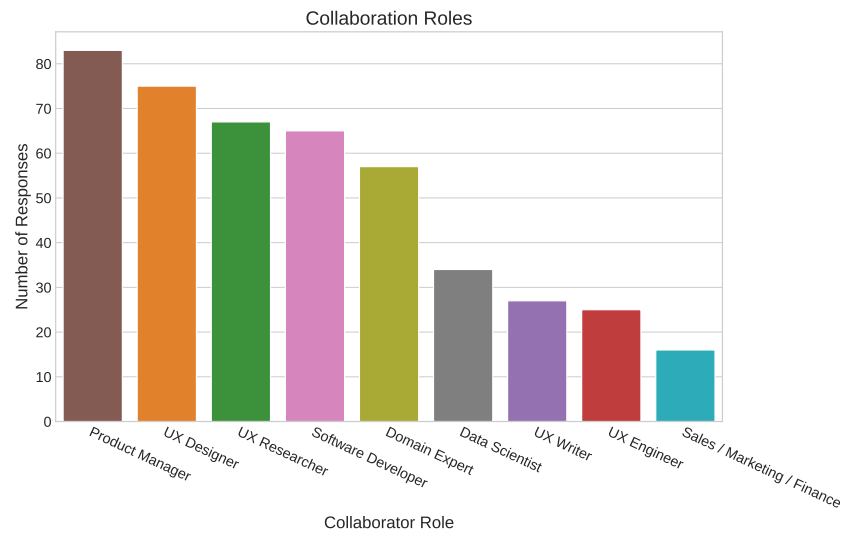
## 4 FINDINGS

With the survey data, we analyzed responses to questions relating to cross-functional collaboration across UX design stages, tool usage and collaboration, reuse with design systems, and transitioning design work to development.

### 4.1 Collaboration Across Different UX Stages

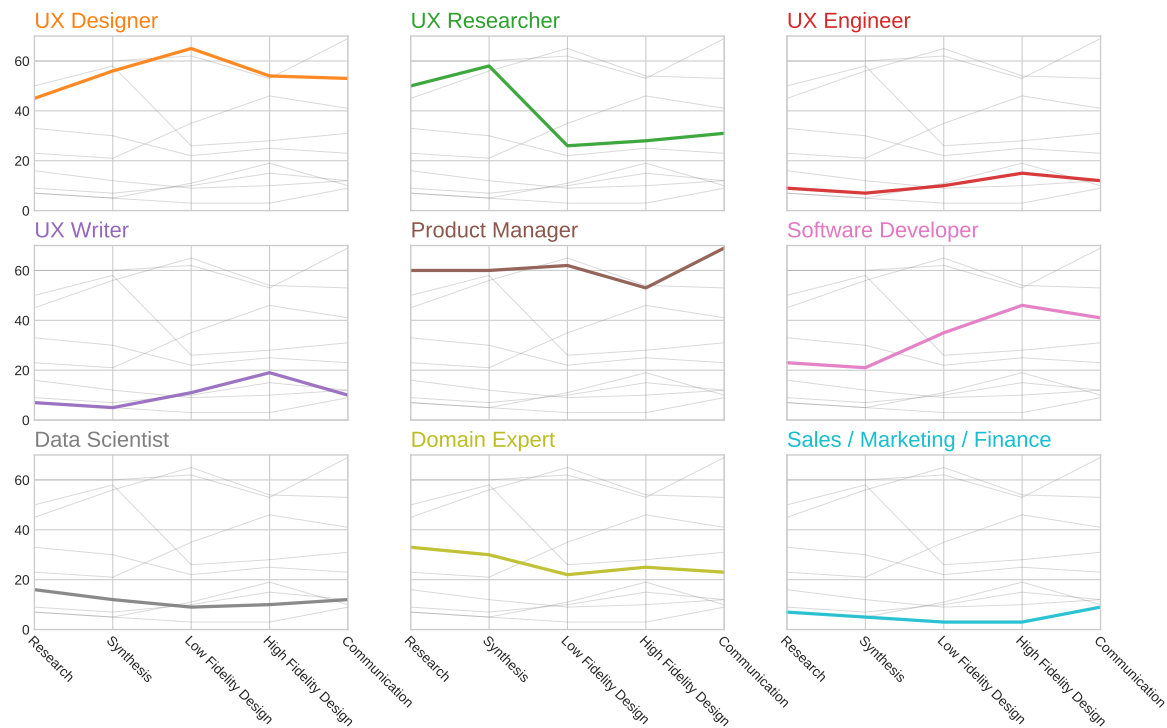
Professional UX work has necessitated cross-functional collaboration in organizational workflows. Indeed, all but 5 participants (96%) indicated that they collaborated with others in their work. With this prevalence of collaboration in mind, we asked: **with whom do UXPs collaborate in each stage of the design process?** In addition to surveying the stages of the design process (Research, Synthesis, Low-Fidelity Design, High-Fidelity Design, and Communication) in which participants were involved, we collected participants' reported UX roles, UX team sizes, and team structures.

**4.1.1 Product Managers and Designers as the Strongest Collaboration Partners.** Participants who disclosed their collaboration patterns ( $n = 112$ ) generally reported working with UX designers and product managers the most (see Fig. 3a). Those who identified as designers worked with more UX researchers in the early design process stages (research and synthesis), with other designers more in the low- and high-fidelity design stages, and with product managers and software developers in the later design process stages (low/high-fidelity design and communication). Those who identified as researchers worked with both designers and product managers more in the earlier stages of design (research and



(a) How many participants reported working with each type of collaborator role at least once through each design phase ( $n = 112$ ). Overall, UXPs collaborated most with product managers, UX designers, UX researchers, and software engineers, in that order.

Collaborations through Design Stages



(b) How many participants reported working with each type of collaborator role, through each design phase ( $n = 112$ ). Overall, collaborations with product managers and UX designers were consistently frequent throughout the design process, collaborations with UX researchers were more frequent in the early design stages than later stages, and collaborations with software developers were moderately more frequent in the later design stages than earlier stages.

Figure 3: Aggregated and disaggregated views of participants' collaboration. Roles are colour-coded between the two charts.



synthesis). Noticeably, UXP interactions with software developers were usually limited to the UX designers; UX researchers did not collaborate with software developers often (see Fig. 3b). Some collaboration patterns varied across organization and team size—for example, larger companies and teams saw more collaborations with UX researchers, especially in the earlier design process stages, due to the presence of dedicated research teams within these organizations. Standalone UX consulting teams or single-person teams collaborated more frequently with domain experts throughout the design process.

**4.1.2 UX Practitioners Commonly Initiated Collaborations.** We asked participants how their collaborations with non-UXPs were typically initiated. Out of those who responded ( $n = 108$ ), over half (63%) initiated the collaboration because they or their team wanted others' input. As one participant explained, *"I initiated because I wanted to give input. Non-UXers often forget to include UX. We include ourselves even when not invited."* A substantially lower fraction (28%) had their non-UX collaborators initiate the collaboration as the collaborators sought UX input. The rest experienced a combination of both. This imbalance in collaboration initiation is in some ways expected—previous work [44, 110] has shown that embracing UX can be challenging for organizational cultures, and UXPs may need to go to extra lengths for their work to be recognized and understood. However, given the rapid proliferation of UX [80] and establishment of UX teams in technology companies—to which many of our participants belonged—it was surprising that collaboration was still decidedly reliant on UXPs taking the initiative.

**4.1.3 Summary.** In general, UXPs appeared to have strong collaborations with product managers and UX designers throughout the entire design process, with some collaboration with UX researchers at the beginning of the process and with software developers later on. Larger teams and companies saw more collaborations with UX researchers, while smaller ones collaborated more closely with domain experts. Regardless of organization and team size, UXPs usually initiated these collaborations to gather others' input on their work.

## 4.2 Tool Use and Collaboration

In recent years, there has been increasing interest in collaborative and cloud-based UX tools [89]. This led us to ask: **what tools are UXPs using throughout their workflows, and how are UXPs collaborating in these tools?** Through free-form text responses, we asked participants to list their toolset in each stage of the design process and describe any tool-based collaboration practices they engaged in. Some participants responded with conceptual tools (e.g., journey maps) in addition to software or physical tools; we ignored those as they were outside the scope of our "tool" definition in this paper. One author then turned the responses into tool lists for tools that were mentioned by more than one participant. Tools were tallied based on the number of participants who mentioned them (as an estimate of popularity) and iteratively organized into 10 high-level categories based on their functionality (see Table 2). The author also open coded the collaboration practices described by participants. Below, we discuss our findings in tooling and tool-based collaboration in further detail.

**4.2.1 Figma as an All-In-One Tool.** We examined differences in tool use across UX stages by using a Friedman test and found the differences to be highly significant ( $Q = 33.20, p < 0.01$ ). The most common tools used for each stage are shown in Fig. 5.

The use of Figma was dominant across all 5 stages, aligning with similar findings from a 2021 survey on design tools [89]. As one participant commented, *"Figma has been an all-in-one tool, purely from the perspective that we can design, collaborate and update all within one file."* Indeed, the Figma ecosystem (which includes FigJam) proved to be at least twice as popular as the second-place tool in all stages. This revealed the presence of an all-encompassing "Swiss Army knife" for UX workflows. Tools with similar properties also exist in workflows of other software-based domains, such as data science, but those domains do not necessarily exhibit the same concentration of tool popularity observed in UX [123].

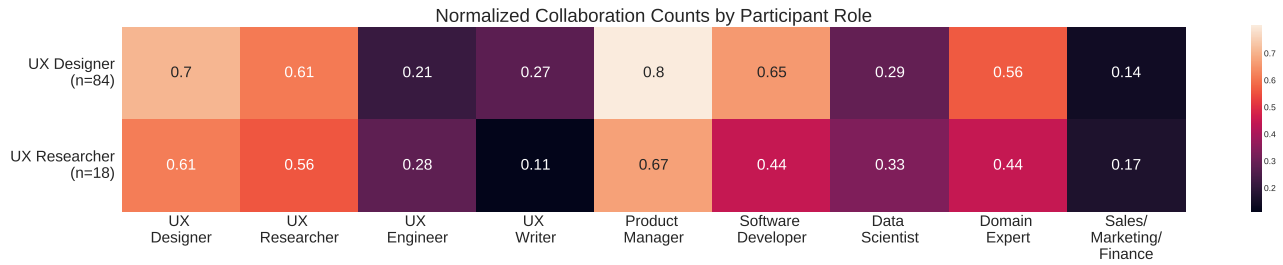
Not surprisingly, the related stages of Research and Synthesis had significant overlaps in tools: Figma, Miro, Mural, and Excel were all among the top 5 in each stage. Similarly, Figma, Sketch, and Adobe XD were all in the top 5 for both low-fidelity and high-fidelity design. All top 5 tools across the stages contained collaborative features or properties, with the exception of pen and paper during remote work. Most also contained social affordances for co-presence, such as multiple cursors and profile photos of other users if they are active in the tool.

**4.2.2 Synchronous and Asynchronous Collaboration in a Single Tool.** In addition to asking participants about the tools they used, we also asked them how they collaborated and communicated in those tools. Working in *shared, multiplayer canvases* stood out among responses—it was common for many collaborators to work and interact with each other from within one file. We observed two common roles emerging from workflows in shared canvases: author and editor. As one participant put it, *"usually one person generate[s] the initial output, then others add on and/or comment on refinement."* The author did not necessarily have to be a UXP, although that was often the case. Sometimes, software engineers and domain experts were invited to mock up their ideas in a collaborative design environment, and the UXP would refine and iterate on those ideas. This was valuable as participants collaborated in shared canvases both synchronously and asynchronously.

**Synchronous collaboration.** Some participants expressed a preference for synchronous collaboration as their collaborators tended to not review their work asynchronously under the expectation of a future presentation: *"some people go ahead and access the file on their own time but most people just wait for me to do a walkthrough presentation to review it and provide their feedback."* Indeed, when working with other UXPs, the verbal aspect of synchronous design critiques was seen as valuable: despite annotating on prototype screens and taking notes on the canvas and elsewhere, many participants still felt that much of their justification and documentation happened verbally. One participant mentioned that they recorded voiceovers walking through user flows to better explain their thought process.

**Asynchronous collaboration.** Participants often used commenting features and virtual sticky notes to leave traces of information and feedback for others when collaborating asynchronously. Comments were mentioned as effective for working with those in





**Figure 4: Normalized counts of collaborators’ roles (x-axis) broken out by participant role (y-axis). PMs, UX Designers, UX Researchers, and Software Developers are the most common roles that Designers and Researchers collaborate with. Only 12 participants identified as other roles; they are excluded from this figure.**

Tool category	Tool (# participants who mentioned this tool, >2 only)
user research	UserTesting (5), SurveyMonkey (2)
visual diagramming	FigJam (32), Miro (20), Mural (19), Google Jamboard (2)
sketching	pen and paper (18), physical whiteboard (5), Balsamiq (2), Procreate on iPad (2)
design	Figma (80), Adobe XD (11), Sketch (11), InVision (9), Adobe Illustrator (2), Axure (2), Flinto (2)
design delivery	Adobe After Effects (4), ProtoPie (3), Zeplin (2)
programming	frontend code (5), Swift Playgrounds (2)
communication software & artifacts	Microsoft Teams (15), Slack (8), Email (5), Outlook (4), videos (4), Google Meet (3), PDF (3), spreadsheets (3), calls (2), Zoom (2)
document sharing & editing	Google Docs (11), Excel (9), Word (7), Google Sheets (6), Google Suite (2), OneNote (2)
project management	GitHub (4), Notion (3), Airtable (3), Quip (2)
presentation	PowerPoint (30), Google Slides (10), Keynote (6)

**Table 2: Categories of tools used by UXPs in their workflow and the number of participants who mentioned using each tool. Note that design refers to tools that support end-to-end interface design from low-fidelity wireframes to high-fidelity prototypes, whereas design delivery tools take high-fidelity designs and prepare them for publication/testing/handoff.**

another time zone, or to non-UX stakeholders: “Most non-designers are used to text-based feedback in Quip<sup>2</sup> so I’d write documents and have them comment.” Commenting was distinct from the act of explaining design decisions, which were often documented via text-based annotations directly on or beside their designs. Participants also documented their decisions by taking advantage of the version-tracked design files to show the progression of mockups over time, overlaid with research insights and/or technical constraints from developers. These *layered artifacts*<sup>3</sup> helped participants frequently reference user or academic research, and actively share lower-fidelity work such as wireframes and user journeys.

**4.2.3 Presentation of UX Work Involved Different Techniques and Environments.** When presenting UX work, participants often left the shared canvas for presentation software, such as PowerPoint, or entered a special mode of the workspace, such as Figma’s prototype mode.<sup>4</sup> This was due to some higher-level leadership not being

familiar with Figma, as well as the ability to structure a better narrative in a slide deck or via prototype mode after the bulk of the UX work was complete.

Participants varied their presentation strategy slightly when showcasing work to non-UX collaborators compared to fellow UXPs. For one, presentations to non-UXPs tended to be more visual and higher-fidelity: instead of sharing drafts of wireframes, participants shared links to and presented clickable prototypes. Sharing an entire design file (common among UXPs) was less common; typical artifacts for communicating decisions were PDF documents, videos, slide decks, and articles. There was also a heavy emphasis on storytelling for non-UXPs: participants told success stories from clients, walked through the context and history behind the steps used to land on a certain decision, and created executive summaries, data visualizations, and other visual imagery to support their work. According to one participant, the storytelling was typically done “in layman’s terms and include[d] lots of high-level overview slides.” Some participants also mentioned that they would establish clear links between their decisions to business benefits and outcomes, with at least one participant explicitly mentioning that they discussed their decisions’ financial opportunities.

<sup>2</sup>Quip is a document and note writing tool: <https://quip.com>

<sup>3</sup>We borrow this term from Schmidt and Wagner [95], who used it to refer to annotated CAD drawings with similar collaborative properties observed in architectural practice.

<sup>4</sup>The prototype mode is a separate interactive environment that simulates how end users may interact with designs [34].

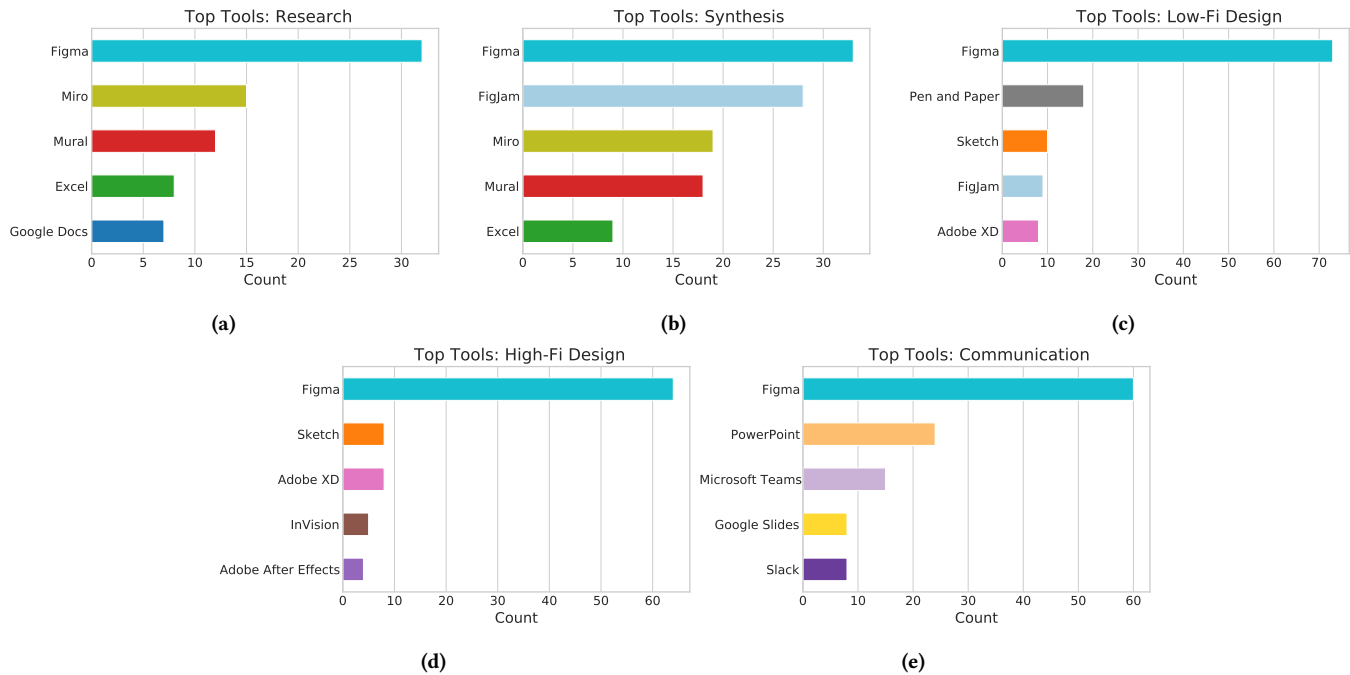


Figure 5: Top 5 commonly used tools in each of the 5 UX stages and the number of participants who mentioned them.

**4.2.4 Summary.** In Section 4.1, we saw that UXPs actively engaged in collaboration with various roles throughout the design process. Here, we show that collaborative, multiplayer tools hosting shared files to be used across multiple workflow stages were widely used among UXPs and their collaborators. UXPs employed a variety of strategies for synchronously and asynchronously collaborating in these shared environments, but the strategies can differ based on whether they were communicating with other UXPs or non-UXPs.

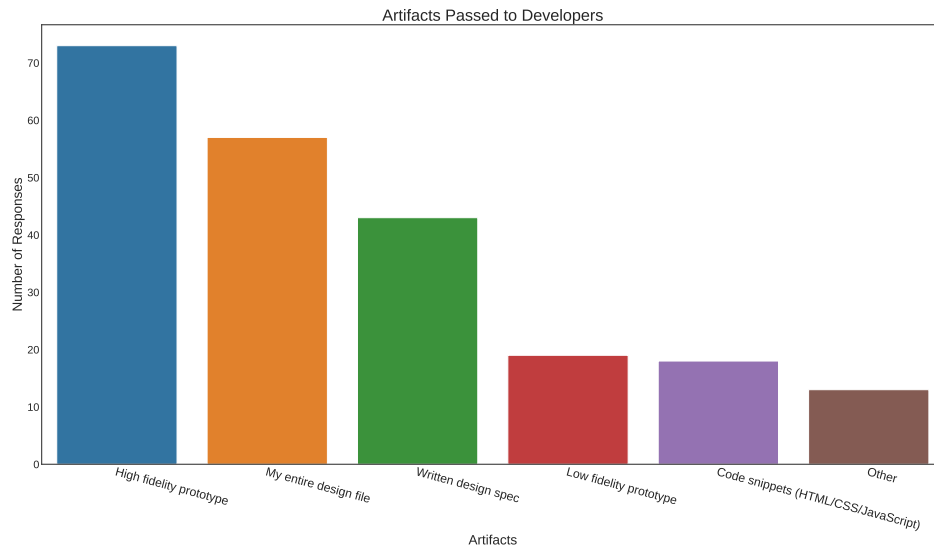
### 4.3 Collaboration During Handoff

In the words of one participant, “*the real world challenge starts when it comes to implementation.*” We asked participants about their current practices and challenges when they handed off their UX work to software developers for implementation, paying close attention to the scenarios where the design was different from implementation. We refer to this phenomenon as *design-development divergence* as the original goal of alignment between design and code is often discarded due to procedural breakdowns over time [71]. For participants who had experience in designing machine learning (ML)-powered user experiences, we asked about their collaboration practices and challenges with ML practitioners (e.g. data scientists, ML engineers). We did so given recent interest in enhancing communication and handoff between UXPs and ML practitioners (who are distinct from software developers [5]), which has been noted as uniquely difficult in prior literature [104, 105, 121]. Unlike our findings in Section 4.2, this section highlights some challenges and desiderata of *sequential* collaboration—where UXPs and developers or data scientists work one after the other in different domains—rather than *parallel* collaboration—where UXPs and developers work together within a single stage of the design process.

**4.3.1 (Over)communication During Handoff.** Out of the 96 participants who participated in handoff, the vast majority (92%) shared their work with developers for handoff. Participants used a range of artifacts to share their designs with developers, including high-fidelity prototypes, the entire design file, written design specs,<sup>5</sup> and CSS code snippets generated by design tools. High-fidelity prototypes were the most commonly shared artifact, followed by the entire design file (see Fig. 6). Most (80%) shared more than one artifact; the most common combination of artifacts was the design spec, high-fidelity prototype, and the entire design file (17%), followed by the same combination but without the design spec (11%).

Sharing the entire design file was convenient in collaborative tools, but participants pointed out pitfalls with *too much transparency*. For instance, the lack of a constrained view where one can control which specific screens collaborators can see may be troublesome: “*With Figma you can point someone to a specific frame but they can just zoom out and access every other page or frame in the entire file which often leads to confusion and frustration.*” Some participants mitigated this issue by clearly demarcating a region in the file for the final product, with watermarks and naming conventions that indicate work-in-progress designs. In general, participants relied heavily on attention-grabbing text annotations (or even paragraphs of writing) in the design file itself to explain UI behaviours. Unlike providing UX feedback (Section 4.2.2), these annotations are typically *not* written via the tools’ commenting features as developers can forget to view the file on “comment mode”: “*I will write comments in hot pink on the screen so it is obvious that it is not part of*

<sup>5</sup>Design specifications (specs) are detailed documents prepared by UXPs outlining UI design choices (colours, character styles, measurements), as well as interactions (user flows, component behaviours, and UI functionality) [9].



**Figure 6: Distribution of common artifacts shared by UXPs with developers as part of the handoff process ( $n = 88$ ).**

the UI. I do this because you have to go to comment mode in Figma to see comments, [which is] easy to miss.” In addition to annotations, participants also “redlined” parts of their design—adding red lines and text to mark up spacing and sizing of design elements.

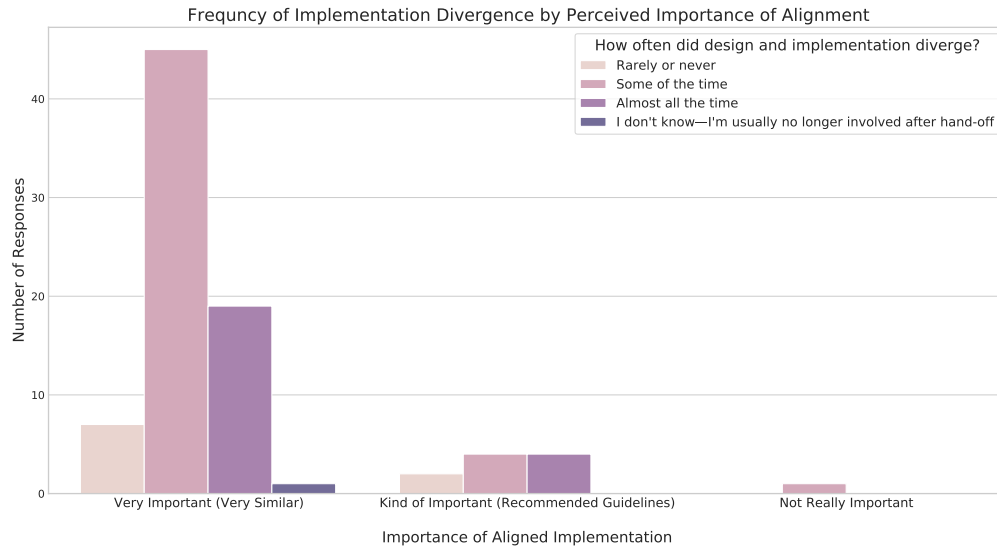
These text-based strategies alone do not solve some broader procedural challenges, such as establishing alignment while implementation happens. As one participant noted, “Design/UX work doesn’t change as much as a developer’s work, so you have to make sure you have ample time to work and still align.” Indeed, even after handoff, participants frequently made themselves available to answer questions and participated in “bug bash” sessions where they inspected the implementation with developers and logged visual “bugs” that did not align with their original design specifications. To organize and streamline the design-to-code process, one participant even broke down the screens they designed into smaller implementation tasks which they then recorded as GitHub tickets, as that allowed them to converse with developers on common ground.

**4.3.2 Encountering and Handling Design-Development Divergence.** Participants considered it to be very important that developers’ implementation closely match the look and feel of their designs (86% of  $n = 87$  respondents). A much smaller fraction (13%) expected their designs to be treated as recommended guidelines for implementation, while only one respondent thought developers could freely follow or reject their designs at will. Despite this, expectations did not align with reality: 57% sometimes encountered design-development divergence, while 26% encountered it almost all of the time (see Fig. 7). This quantifies some results of previous work that investigated why design breakdowns can easily occur at design-development boundaries [71].

More than half (59%) reported that once divergence happened, design changes were necessary to bring their work closer to implementation. This was surprising as it points to implementation being the “source of truth” instead of an intermediate representation of design and implementation, as found in prior work [67, 71]. Should

this be how it is? Our participants appeared divided on whether design should conform to implementation, vice versa, or meeting somewhere in the middle. Many (41%) “caught up” with the implementation by editing or even re-doing some of their designs. A lesser fraction (10%) was adamant about the “proper” implementation of their designs and gave feedback to developers to better align with designs. A few participants acknowledged that designs can sometimes be technically infeasible—developers may run into API or environment constraints only after starting implementation. Direct communication with developers seemed to be the most effective way to identify the problem’s root cause. One participant stated that upon deliberation, what was thought to be a technically infeasible design simply needed additional scenarios that developers encountered during implementation. Other courses of action after encountering divergence included continuing to use original, outdated designs but keeping in mind that they are different from the implementation (14%), and discarding the original designs altogether to design off of screenshots of the implementation instead (10%).

**4.3.3 Knowledge and Process Gaps for Handoffs Involving Machine Learning.** 58 participants recounted their experiences on an ML-enabled project. Commonly mentioned ML technologies include recommender systems (50%), natural language processing—e.g. personal assistants, speech-to-text—(48%), computer vision—e.g. object detection, photo tagging—(40%), and spam & malware filtering (9%). Unlike handoffs in non-ML projects, where UX work preceded technical implementation, we observed that ML handoffs often followed a *reversed* workflow where the ML model is implemented prior to the start of any UX work. Consequently, it was essential for UXPs to familiarize themselves with model capabilities to understand the feasible design space. Most (74%) learned about what models can and cannot do indirectly through frequent meetings with ML practitioners, watching them present their work about the models, and asking for demos, agreeing with prior work [105, 120]. A smaller



**Figure 7: Frequency of design/implementation divergence by perceived importance of alignment. Most respondents considered similarity important but saw more divergences than not.**

fraction of participants (16%) interacted directly with the model to learn more about it, feeding it test cases in sandbox testing environments as well as viewing common metrics for model evaluation (e.g., F1 score, ROC AUC). The remaining 10% were not aware of model capabilities at all.

We saw two categories of challenges when UXPs collaborate with ML practitioners: *knowledge* challenges and *process* challenges. Knowledge challenges included technical jargon, which was often not adequately explained, and the inability to understand technical constraints of the model. However, it was more frequent for participants to mention *process* challenges, an example of which is tricky timing:

*“A lot of times we want to test a concept with real data, but the models we’re testing with aren’t production-ready yet, or researchers have moved onto a different problem, so there can be kind of a lag for getting data, or getting feedback on how a model would theoretically work before we start usability.”*

Participants also implied that their collaborations also posed a one-way burden on ML practitioners, needing to “bug” them to ask them questions and having them “*explain [concepts] to me like I’m five.*” To mitigate this burden, participants adopted the strategy of orchestrating collaborations “*in a way that it takes minimal effort for [collaborators]*” to share knowledge. Despite this, some noted that ML practitioners were quite excited about design implementations and were very interested in having a UX perspective, but pointed out the common pitfall of “*seeing the design discipline as a service, not a partner*” when expecting handoff artifacts from UXPs. To ameliorate these challenges, participants wished for a way to “translate” technical requirements to more understandable UX contexts, such as user scenarios. Another popular suggestion was an interactive environment in which they can train their own variations of models to enhance their behind-the-scenes understanding. For instance,

one participant wanted a “*plugin where you could enter in fake training data to simulate a prediction. Example: For an NLP [...] I might have 20-30 canned phrases and each would have a preordained result.*”

We were surprised that a few participants (7%) did not believe designing ML interfaces was noticeably different than non-ML ones, a sentiment not obvious in prior work. As one participant noted, “*ML / AI [is] not some special, wild wild west.*” A common reasoning for this is that UX work primarily resides in the front-end interface layer, while AI is an abstracted back-end. This is perhaps best summarized by the following quote, the sentiments of which have been echoed by others:

*“For me, ML/AI are back-end tech, almost black-box in that I just need to know user inputs, behaviours, and outputs; how those are generated aren’t essential.”*

Divergent perspectives on whether designing with ML was uniquely challenging can perhaps be attributed to variance in the nature of projects, as one participant hinted: “*in this particular effort the particular details of our ML model did not impact design decisions that much.*” ML projects may indeed fall into various levels of AI design complexity, as taxonomized by Yang et al. [121]—ML systems with fixed capabilities and few possible outputs may require fewer specialized processes and tools than ones with constantly evolving capabilities and seemingly infinite outputs.

**4.3.4 Summary.** Despite collaborative features and handoff guides being offered in UX tools [13], UXPs still encountered persistent challenges similar to those outlined in previous work when handing designs off to developers. Handoffs to ML practitioners were often procedurally distinct, almost reversed, compared to handoffs to non-ML developers. Still, when designing for ML-enabled interfaces, many encountered knowledge and procedural challenges, expressing interest in new tools and shifts in collaboration styles.

## 4.4 Design Systems and Reuse

In UX work, design systems are used to standardize and document design work for reuse, especially across larger organizations [24]. UX tools have made design system management more accessible through specialized features for storing, organizing, and updating contents of the system [3, 18, 31]. We wanted to better understand the effectiveness of collaboration and reuse through design systems, so we explored our research question: **what are UXPs' current expectations and practices around reusing design components and managing design systems in collaborative UX tools?**

**4.4.1 Expectations Around Reusing Designs.** Most participants reused designs from other UXPs in their company (71% of  $n = 95$ ), as opposed to designs from outside their company. This was especially true for large companies (>10,000 people), possibly due to confidentiality and higher maturity in the internal design system. Smaller companies had higher proportions of UXPs who did not use designs from others (36% for companies of fewer than 100 people, 40% for companies of 101–1000 people). For solo teams, 50% of respondents did not use designs from others, a much higher proportion than for any other team size. Additionally, most UXPs expected their work to be reused (81.5%), across all company and team sizes. This expectation aligns with the high response rate we saw for UXPs using assets from design systems.

**4.4.2 Challenges in Design System Management.** To better understand obstacles to working design systems in collaborative tools, one author qualitatively categorized open responses detailing challenges in using and adopting design systems with collaborators. The most common responses were due to the design system not being able to scale to diverse collaborative teams (22% of responses), followed by lack of communication and ambiguity in ownership (20%). Refer to Table 3 for all categories of challenges and associated number of responses.

Based on these results, maintenance and update of design systems to serve the diverse needs of multiple user teams seemed to be the most difficult challenge for most UXPs. As one participant stated: *"It's the classic challenge of trying to design something that works for everyone while still allowing the freedom of a given project to make a design optimized for their given needs."* Communication—in this case, between design system creators and users—appeared to be the next most common challenge. With no clear ownership or guidance on who should update the design system and how to enforce standardization, UXPs struggled to find design systems useful. One participant mentioned: *"We all use [a design system] if it's there, but since nobody has ownership, it may fail to have new assets or stylings."* Another pointed out that it can be ambiguous when *"deciding what elements need to go into [the design system], and what don't,"* particularly for new designers who are just becoming familiar with the system. Finally, participants expressed frustration at the lack of support to transfer design systems between tools. One complained: *"design systems are often arbitrarily linked to one program like Figma, which prohibits wide use,"* while another explained that every time their team upgraded their tools, they needed to recreate their design system, which prevented the system from scaling to desired levels. Furthermore, this opened up opportunities for versioning issues in design systems, which one

participant described when their team used slider components: *"we ended up using 3 different styles of sliders—ones from the 'old' design system, ones from the 'new' design system, or custom made ones."*

**4.4.3 Summary.** Component reuse in collaborative tools was prevalent, particularly in larger organizations; however, new challenges emerged in managing design systems for structured reuse. Although the rigidity of design systems may be a reason for their adoption in the first place, it was also what makes them difficult to adapt to diverse use cases and accommodate more contributors. Some current practices and inter-tool support for collaborative design system management also appeared hazy and underdeveloped.

## 5 DISCUSSION

Throughout our survey, we see that UXPs leveraged collaborative practices and tools throughout the design process to enhance their and others' work. We took a deeper dive into specific challenges outlined by previous work—namely designing with ML [104, 120, 121], developer handoff [67, 71, 115], and design system management [24, 78]—and found those challenges to persist in collaborative environments. Furthermore, UXPs faced new challenges such as exposing extraneous information that confuses non-UX collaborators and writing attention-grabbing annotations to explain their work. We map our findings onto relevant stage(s) of the double diamond given its importance in design practice (see Fig. 8). Interestingly, findings that do not span the entire process lie only in the second diamond. This indicates that more collaborative challenges arise after the formulation of the design problem rather than before, but we note this may also be a bias due to our sample containing considerably more designers (who typically work around the second diamond) than researchers (who typically work around the first). We unpack our findings and corresponding implications below.

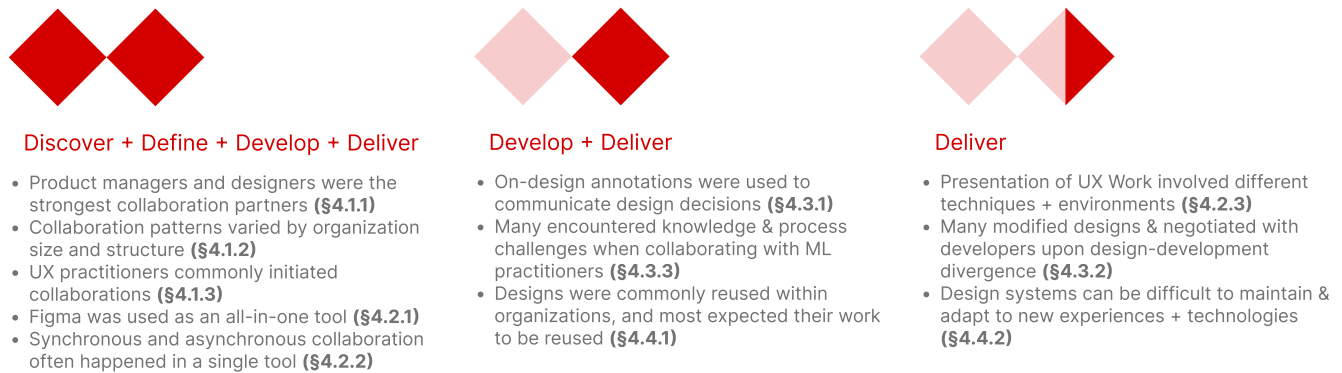
### 5.1 UX Tools Act as a Hub, But Not Necessarily Practitioners

In RQ1, we asked, "with whom do UXPs collaborate at each stage of the design process?" As delineated in Section 4.2.2, UXPs collaborate extensively in their tools throughout different stages of the design process. In particular, UXPs tended to invite collaborators (e.g., product managers, software developers) into tools typically considered to be for UX (e.g., Figma). Collaborators then "jammed" together—interactively brainstormed, created, and edited artifacts—synchronously and asynchronously in structured ways through assigned roles of authorship and editorship. This tendency sheds light on RQ2: "what tools and tool-based strategies are UXPs using throughout their workflows?" Our findings here suggest that UX tools are a *hub* of collaborative activity for the entire team, not just among UXPs. Here, we use "hub" to refer to a common space where stakeholders from multiple domains can (and are invited to) collectively contribute, deliberate, and glean pertinent information, as opposed to space accessible to those from a single domain (e.g., UX). Similarly, we use "hub tools" to refer to tools that offer a hub workspace. For example, product managers can enter hub tools to receive progress updates and synthesize brainstormed ideas into higher-level product roadmaps, and developers can frequently reference designs in the tools as a guide during implementation and request design changes based on technical feasibility.



Reason	Number of Responses
Collaborative team discrepancies with nonscalable design system	13
Lack of knowledge communication with ownership ambiguity	12
Design system changing too often	8
Mismatch between design and code libraries/models	7
Lack of documentation or awareness of systems	6
Insufficient resource budgeting and efficiency	6
None or N/A	7

**Table 3: Categories of responses to the question "What are some challenges you observed in getting design systems adopted widely by your collaborators, if any?", with the most cited reason being design systems failing to scale to collaborative teams.**



**Figure 8: Our main findings mapped onto the double diamond design process. Most of our findings span more than one stage.**

That said, we also observed in Section 4.1.2 an imbalance in how collaborations are initiated. Despite heavy cross-functional collaborative activity in UX tools, most collaborations were initiated by UXPs. Additionally, we see in Section 4.1 that the two roles with which UXPs collaborate the closest throughout the entire design process are UX designers and product managers. The lack of sustained collaboration with other roles is an indication that UXPs are typically not situated in a dominant role within a collaborative network (a “hub” role [123]) where they form extensive mutual collaborations with a wide range of non-UX roles [20]. Instead, product managers appear to be the most promising candidates for hub roles, which is unsurprising given the cross-functional nature of the job [21], but we cannot verify this hypothesis with our UXP-specific data. The combination of these two insights suggests an interesting distinction between the hub-like nature of *UX tools* and *UXPs*—the tools may act as a collaboration hub in the absence of the practitioners doing the same. In prior work on remote software teams, researchers found that workers have become familiar with collaborative technologies [15] and thus have few issues with collaborative technology readiness, even as they continue to struggle with establishing common ground and bi-directional collaboration readiness [85]. Thus, challenges with collaborative practices, such as miscommunication between UXPs and developers, may be harder to eradicate even as collaborative UX tools proliferate among non-UXPs.

UX tools are not the only example of hub tools used among software teams. Task management and communication platforms

such as Jira, GitHub, and Slack also display similar characteristics [22], and we see increasing interest in *inter-hub compatibility* where hubs offer and/or accept specialized integrations with other hubs [6, 7, 32]. This is especially important when teams in different domains are already deeply accustomed to domain-specific representations and tools [112]. In the context of multiple interconnected hubs, the aforementioned distinction suggests that hub tools should offer specialized features for hub roles to move horizontally between hubs and coordinate work efficiently. Strategies for using hub tools may also shift with inter-hub integrations—Hu et al. showed that tools and practices that benefit the inter-team collaboration may in fact harm intra-team collaboration, and vice versa [48]. Whether and this phenomenon extends to UX or UX-adjacent hubs is a question ripe for investigation in future work.

## 5.2 Design Systems Require Governance and Flexibility

The emergence of design systems has supported the scaling up of UX practices across diverse software products and services [8, 16, 42, 49, 75]. With this in mind, we asked RQ4: “what are UXPs’ current expectations and practices around reusing design components and managing design systems in collaborative UX tools?” Design systems act on the promise of reducing effort, scaffolding learning, and increasing cross-functional collaboration [24], but we found that many current design systems fail to effectively nurture collaborative behaviours and consequently undermine this promise. Specifically, there is little guidance on who is responsible

for maintaining and enforcing the design system, which erodes its potential to be trusted and adopted. The lack of flexibility in the design system to account for new technical capabilities and interactions also limits its ability to scale.

**5.2.1 Governance Handbook.** Besides the 3 core components of design systems defined by Moore et al. [78] (see Section 2.2), we argue that a fourth component—a *governance handbook*—should be included to eliminate ambiguity over who should maintain the design system and how they can do so. This handbook may include a system for assigning UXPs to oversee parts of the design system they work with as moderators, procedures for submitting and modifying components, guidelines for asking moderators questions about general usage, as well as guidelines for moderators to resolve design and usage inconsistencies that arise. Just like how a system of governance is essential for the continuity of a scalable, open source codebase [63], a governance handbook can likewise help manage a scalable design system.

**5.2.2 Variation Patterns.** In addition to a governance handbook, we recommend that *interaction patterns* [78] specify not only reusable components and their interactions, but also *variation patterns* that expose flexibility in their specifications. That is, not only does a component define its appearance and behaviour, but also how that appearance and behaviour may be adapted for new scenarios. For example, a variation pattern may include an extended font family to account for scenarios that demand sophisticated text formatting, and a guide on extending a button style to any UI featuring colour blocks, such as a progress bar. Variation patterns can help scale design systems into unforeseen territory by allowing practitioners to more concretely envision how existing patterns can be adapted and extended.

### 5.3 Implications for UX Tools

**5.3.1 One Tool, Multiple Modes.** One of our prominent findings is that Figma is strikingly popular with UXPs throughout all stages of the design process. We observed this across organization and UX team sizes. However, we also witnessed challenges arise from inviting collaborators into Figma’s all-in-one canvas environment—developers found themselves wading through a trove of irrelevant screens in a design file they were not familiar with, leading to confusion. While using an open canvas that showcases evolution of work enhances process transparency, that same transparency can be overwhelming and obstructive when unnecessary.

Many design tools already support a “presentation mode”<sup>6</sup> in which UXPs can present a polished, interactive prototype. This mode takes place in an environment separate from the rest of the design file and typically consists of an interface that mimics an end-user’s interaction with the prototype. Our findings suggest that in addition to a presentation mode, design tools can introduce other modes to benefit collaboration. Ideally, these modes will offer two capabilities: customization by users who initiate their use (e.g., selecting specific artifacts to show to specific collaborators) and supporting the attachment of mode-specific tools. For example, users can launch a “developer mode” in which only certain screens they select are shown to developers, accompanied by a

web inspector to help extract HTML/CSS snippets. When working with UX Writers, a mode featuring an interface for commenting on and critiquing text may be more helpful than a standard design environment. Wireframes of example modes are shown in Fig. 9. By shifting to a multimodal approach in which only certain portions of the overall work environment are exposed at once, all-in-one tools can still offer the convenience and benefits of supporting work across design stages while improving organization during multi-role collaboration.

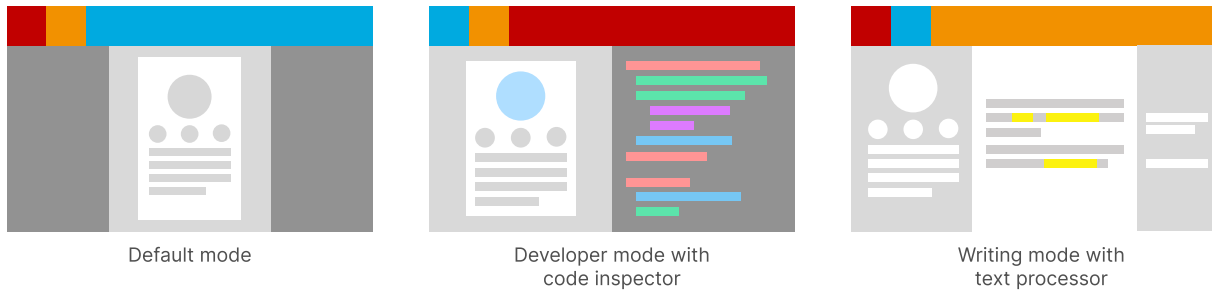
**5.3.2 Supporting Linkage of Design and Code Components.** In RQ3, we asked, “What practices and challenges arise when UXPs hand off their work to software developers and ML practitioners?” We see from Section 4.3 that despite efforts to establish clear channels of communication between UXPs and software developers, design-development divergences are still common. Previous work [67, 71, 115] have called upon co-creation practices to allow designers and developers to collaborate throughout the entire design process to minimize risk of divergence, but those practices do not seem to be actively used among UXPs in our population. Indeed, while 88 participants claimed they *shared* their work with developers for handoff, at most 46 *actively collaborated* with developers at any given stage.

How can tools help UXPs think of handoffs as a co-creation effort rather than simply a transfer of work? One approach is to keep a design and its associated implementation mutually updated. That is, we propose *linked components* where designers and developers can 1) design and prototype components through graphical manipulation, 2) actualize graphical components’ interactive behaviours using code, and 3) collaboratively link the two and test for performance and edge cases (see Fig. 10). These links should be robust but flexible—changes to the component, whether it’s the design or the code, can be indicated in the links and reviewed by others if necessary. An example of this link in action is as follows. A designer mocks up a UI component that displays the results of a rider-driver match for a ride-hailing app. The developer adds frontend code to the component that implements its specified behavior. After some user testing, the designer realizes the empty state (in which there are no available drivers for the rider) is not handled well, and modifies the design accordingly. The change is signalled to the developer, who then updates the code for the empty state. We see value in component-based links due to the rise of component-based design systems in UX [24], a shift towards frontend programming frameworks emphasizing reusable components [74, 106, 122], and a long-standing emphasis on unit testing in software engineering [96]. With our proposed approach of flexibly linking design and code components, we *anticipate* design-development divergence and *prepare practitioners for deliberation* rather than embarking on the much more arduous journey of eliminating all possibilities of divergence at all times.

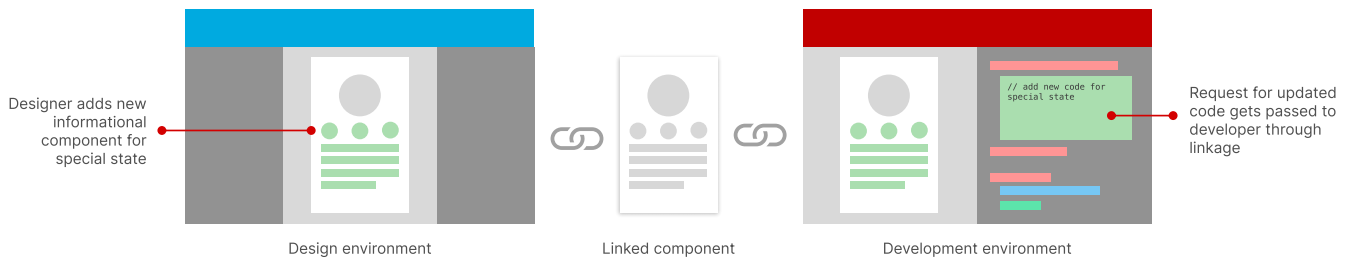
**5.3.3 Writing assistance for on-design annotations.** Despite robust commenting capabilities in collaborative tools, we found that UXPs still heavily rely on writing text directly on or beside their designs to document design decisions, citing superior visibility over comments as a reason for doing so. Writing these annotations is often an open-ended—and in the case of larger projects, daunting and time-consuming—task, especially if there are few guidelines on

<sup>6</sup>An example of such a mode is Figma’s prototyping mode [34].





**Figure 9: Design tools can offer multiple modes for collaborating with different roles. For example, a designer may switch to “developer mode” and allow developers to experiment with code snippets alongside the UI, or “writing mode” to allow writers to focus on writing and editing UI text in a separate interface.**



**Figure 10: A designer can manipulate a linked component via graphical editing in their design environment, while a developer can edit via code in their development environment. The link may also facilitate decision-making deliberations between the two parties.**

how to discuss design decisions, how much detail to write, and whether a common structure or template should be used for different annotations. Conveniently, there has been a surge in interest in AI-assisted writing with large language models for tasks such as creative writing [19, 23, 107], scientific writing [37, 73], and probing the model’s capabilities as a writing collaborator [65]. These writing tasks share the same open-ended nature (and in the case of scientific writing, the logical nature as well) of writing annotations. In addition to models working purely with text, multimodal transformers such as CLIP [92] can learn representations that link textual data with visual data [93]. Multimodal transformers with AI-assisted writing present a powerful combination for authoring on-design annotations. References to visual components can be (optionally) translated to text for writing inspiration, and writing of the annotation itself can be further enhanced by suggestions from language models. This approach can draw upon work from the accessibility community on screen readers, which already translate visual web components to descriptive text [79]. The layer hierarchy tree in design tools can also be used by the model to learn relationships between components. The learned UI structure can then be supplemented user-provided context to automatically generate semantically rich, context-aware annotations. As an example (Fig. 11), when a user starts typing an annotation, an intelligent agent takes context from that snippet and, leveraging information about UI relationships given by the layer hierarchy tree in the left panel, suggests an autocomplete for the annotation.

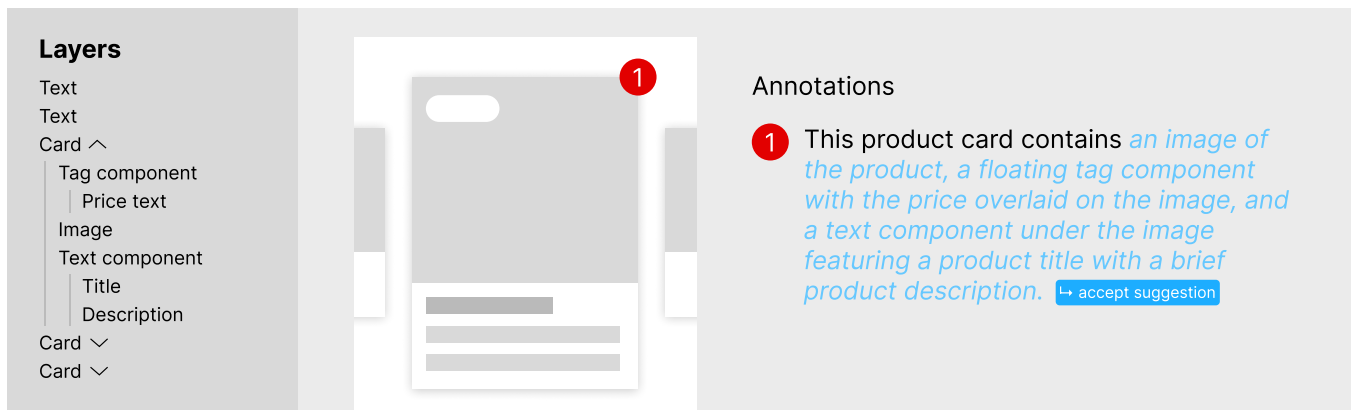
Note that writing assistance for annotations may not necessarily be generalizable to comments. Annotations are *explanatory* while

comments are more *prescriptive*—commentors typically communicate a message about an artifact or concept rather than explain the artifact or concept itself. We defer the further disambiguation of annotations and comments to future work.

## 6 LIMITATIONS AND FUTURE WORK

Our study consisted of only UXPs, which meant we were only able to capture one side of the narrative when it comes to cross-functional collaboration. For example, for developer handoff, perspectives of developers who work with UXPs in implementing designs would aptly supplement our study. Similarly, perspectives from product managers can help us verify our hypothesis that product management is a hub role within software teams. Surveying non-UXPs who work with UXPs may also allow reciprocity in collaboration [123] (or lack thereof) to be surfaced. As such, surveying non-UXPs about their practices and tools when collaborating with UXPs is a promising avenue for future work.

We also observed a sample bias towards large organizations of 10,000+ employees. Large organizations may have UX practices and tools that differ from smaller organizations [44], and although we observed hints of those differences in Section 4.1, they may be dampened by the relatively low number of participants not employed by large organizations. To tease apart differences in practices across organization sizes, future work can recruit UXPs who worked at organizations of various sizes for an interview study about their collaboration experiences. Additionally, to protect confidentiality around specialized procedures that may be in use at particular



**Figure 11:** A user starts typing an annotation. The AI-assisted writing tool then uses context from the user’s snippet and combines it with information in the layer hierarchy tree in the left panel to provide writing suggestions for the rest of the annotation. The user can then press a key to accept the suggestion, or ignore it.

organizations, we did not collect names of employers from our participants and therefore could not identify employer-related biases in our responses.

Although our survey was generally effective in surfacing both high-level practices across the design process as well as more granular practices within each stage, it has its limitations. The survey abstracted the UX process to a linear workflow to align with the ubiquitous double diamond model, but in practice, iterative loops can exist between any and all of the stages. Future versions of our survey can focus on those loops to investigate collaboration challenges when iterating on UX work. Moreover, the balance of breadth and depth in our survey meant that some of our inquiries generated rich responses that we would have liked to further investigate, while others were not as relevant. For example, we felt design systems management was an area that could use more collaborative support given participants’ experiences and asking more questions oriented around this topic could have been more valuable, but our section on design environment customization did not reveal any pertinent findings and we did not end up reporting on it. We can use the lessons learned in this survey to inform the design of future surveys, as well as follow-up interview and co-design studies, to yield more salient insights.

Finally, this study focused on software organizations based in the U.S. to ensure consistent organizational understanding of UX [62]. Future work can compare and contrast UX collaboration practices across countries and continents to strengthen our understanding of UX as a global field.

## 7 CONCLUSION

We deployed a survey to professional UXPs in software organizations to better characterize their collaborative practices and use of tools. We found that UXPs actively collaborated with team members in shared, multiplayer environments such as Figma across all stages of the design process, usually initiating the collaborations themselves. These environments offer new ways for UXPs to communicate and receive feedback on their work; however, they also present novel possibilities for collaborative friction. We posit that

tool multimodality, component linkage between design and code, and assistance for authoring on-design text annotations can mitigate this friction and improve collaborative potential of UX tooling. Tool enhancement alone, however, is not a panacea. New collaborative artifacts and practices, such as a governance handbook for design systems, need to be considered alongside tools for a holistic approach to collaboration enrichment.

UX today is experiencing pivotal growth in terms of the number of practitioners, adoption by organizations, and accessibility to non-UXPs. In the midst of this growth, our work illuminates potential paths forward for advancing collaboration within and around UX to streamline the design of human-centered, interactive systems in practice.

## ACKNOWLEDGMENTS

We extend a warm thanks to all of our participants and reviewers. We also thank Cristen Torrey and Joy Kim for their conversations and early feedback on this project.

## REFERENCES

- [1] Adobe. 2022. Adobe Support Community—Adobe XD. <https://community.adobe.com/t5/adobe-xd/ct-p/ct-xd>.
- [2] Adobe. 2022. Fast & Powerful UI/UX Design & Collaboration Tool. <https://www.adobe.com/products/xd.html>.
- [3] Adobe. 2022. Plugins. Automate tasks. Connect tools. Make XD even more powerful. <https://www.adobe.com/products/xd/features/plugins.html>.
- [4] Andrea Alexander, Aaron De Smet, Meredith Langstaff, and Dan Ravid. 2021. What employees are saying about the future of remote work. *McKinsey & Company* (2021).
- [5] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software Engineering for Machine Learning: A Case Study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. 291–300. <https://doi.org/10.1109/ICSE-SEIP.2019.00042>
- [6] Asana, Inc. 2022. Work on big ideas, without the busywork. <https://asana.com/>.
- [7] Atlassian. 2022. The #1 software development tool used by agile teams. <https://www.atlassian.com/software/jira>.
- [8] Atlassian. 2022. Atlassian Design System. <https://atlassian.design/>.
- [9] Nick Babich. 2017. Design Specifications — Speeding Up the Design to Development Workflow and Improving Productivity. <https://blog.adobe.com/en/publish/2017/11/21/design-specifications-speeding-design-development-workflow-improving-productivity>.

- [10] Nick Babich. 2020. The UX Design Process: Everything You Need to Know. <https://xd.adobe.com/ideas/guides/ux-design-process-steps/>.
- [11] Jonathan Ball. 2005. The Double Diamond: A universally accepted depiction of the design process. <https://www.designcouncil.org.uk/news-opinion/double-diamond-universally-accepted-depiction-design-process>.
- [12] Hani Bani-Salameh, Clinton Jeffery, Ziad Al-Sharif, and Iyad Abu Doush. 2008. Integrating collaborative program development and debugging within a virtual environment. In *International Workshop of Groupware*. Springer, 107–120.
- [13] Joey Banks. 2022. Keeping files organized for your team. <https://www.figma.com/best-practices/guide-to-developer-handoff/file-organization/>.
- [14] Victoria Bellotti and Sara Bly. 1996. Walking Away from the Desktop Computer: Distributed Collaboration and Mobility in a Product Design Team. In *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work* (Boston, Massachusetts, USA) (CSCW '96). Association for Computing Machinery, New York, NY, USA, 209–218. <https://doi.org/10.1145/240080.240256>
- [15] Pernille Bjørn, Morten Esbensen, Rasmus Eskild Jensen, and Stina Matthiesen. 2014. Does Distance Still Matter? Revisiting the CSCW Fundamentals on Distributed Collaboration. *ACM Trans. Comput.-Hum. Interact.* 21, 5, Article 27 (nov 2014), 26 pages. <https://doi.org/10.1145/2670534>
- [16] Uber Blog. 2019. Introducing Base Web, Uber's New Design System for Building Websites in React. <https://www.uber.com/blog/introducing-base-web/>.
- [17] Jim Budd, Surya Vanka, and Andy Runton. 1999. The ID-Online asynchronous learning network: a 'Virtual Studio' for interdisciplinary design collaboration. *Digital Creativity* 10, 4 (1999), 205–214.
- [18] Sketch B.V. 2022. Plugins for Sketch. <https://www.sketch.com/extensions/plugins/>.
- [19] Alex Calderwood, Vivian Qiu, K. Gero, and Lydia B. Chilton. 2020. How Novelists Use Generative Language Models: An Exploratory User Study. In *Proceedings of the Workshops on Human-AI Co-Creation with Generative Models and User-Aware Conversational Agents co-located with 25th International Conference on Intelligent User Interfaces*.
- [20] Fabio Calefato and Filippo Lanubile. 2016. A Hub-and-Spoke Model for Tool Integration in Distributed Development. In *2016 IEEE 11th International Conference on Global Software Engineering (ICGSE)*. 129–133. <https://doi.org/10.1109/ICGSE.2016.12>
- [21] Ellen Chisa. 2014. Evolution of the product manager. *Commun. ACM* 57, 11 (2014), 48–52.
- [22] Mao-Lin Chiu. 2002. An organizational view of design communication in design collaboration. *Design studies* 23, 2 (2002), 187–210.
- [23] John Joon Young Chung, Woosok Kim, Kang Min Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Sketching Stories with Generative Pretrained Language Models. 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 209, 19 pages. <https://doi.org/10.1145/3491102.3501819>
- [24] Elizabeth F Churchill. 2019. Scaling UX with design systems. *Interactions* 26, 5 (2019), 22–23.
- [25] Stephen J Clune and Simon Lockrey. 2014. Developing environmental sustainability strategies, the Double Diamond method of LCA and design thinking: a case study from aged care. *Journal of Cleaner Production* 85 (2014), 67–82.
- [26] Coursera. 2022. What Is a UX Writer? Writing for the User. <https://www.coursera.org/articles/whats-a-ux-writer>.
- [27] Kimberly D Elsbach and Francis J Flynn. 2013. Creative collaboration and the self-concept: A study of toy designers. *Journal of Management Studies* 50, 4 (2013), 515–544.
- [28] Thomas Erickson and Wendy A. Kellogg. 2000. Social Translucence: An Approach to Designing Systems That Support Social Processes. 7, 1 (mar 2000), 59–83. <https://doi.org/10.1145/344949.345004>
- [29] Thore Fiehn, Dennis Wilhelm, and Christian Kray. 2015. Ethermap: real-time collaborative map editing. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 3583–3592.
- [30] Figma. 2022. Add a little magic to your files. <https://www.figma.com/community/explore>.
- [31] Figma. 2022. Add a little power to your files. <https://www.figma.com/community/plugins>.
- [32] Figma. 2022. Create a seamless design workflow. <https://www.figma.com/product-integrations/>.
- [33] Figma. 2022. Figma: the collaborative interface design tool. <https://www.figma.com/>.
- [34] Figma. 2022. Present designs and prototypes. <https://help.figma.com/hc/en-us/articles/360040318013-Present-designs-and-prototypes>.
- [35] Framr B.V. 2022. Design stunning interactive products with your team. <https://www.framer.com/fp/>.
- [36] Nancy Frishberg and Gregorio Convertino. 2020. Barriers and Supports for UX/Design Leaders: From Tactical to Strategic Collaboration. *Design Management Review* 31, 4 (2020), 18–25.
- [37] Katy Ilonka Gero, Vivian Liu, and Lydia Chilton. 2022. Sparks: Inspiration for Science Writing Using Language Models. In *Designing Interactive Systems Conference* (Virtual Event, Australia) (DIS '22). Association for Computing Machinery, New York, NY, USA, 1002–1019. <https://doi.org/10.1145/3532106.3533533>
- [38] Soroush Ghorashi and Carlos Jensen. 2016. Jimbo: A Collaborative IDE with Live Preview. In *Proceedings of the 9th International Workshop on Cooperative and Human Aspects of Software Engineering* (Austin, Texas) (CHASE '16). Association for Computing Machinery, New York, NY, USA, 104–107. <https://doi.org/10.1145/2897586.2897613>
- [39] Athula Ginige, Luca Paolino, Monica Sebillio, Genoveffa Tortora, Marco Romano, and Giuliana Vitiello. 2010. A collaborative environment for spreadsheet-based activities. In *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE, 273–274.
- [40] Max Goldman, Greg Little, and Robert C Miller. 2011. Real-time collaborative coding in a web IDE. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 155–164.
- [41] Google. 2022. Easy and secure access to all of your content. <https://www.google.com/drive/>.
- [42] Google. 2022. Material Design. <https://material.io/>.
- [43] Colin M Gray. 2016. "It's More of a Mindset Than a Method" UX Practitioners' Conception of Design Methods. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 4044–4055.
- [44] Colin M Gray, Austin L Toombs, and Shad Gross. 2015. Flow of competence in UX design practice. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3285–3294.
- [45] Erika Hall and Kio Stark. 2013. *Just enough research*. A book apart New York.
- [46] Lile Hattori and Michele Lanza. 2010. Syde: A tool for collaborative software development. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering—Volume 2*. 235–238.
- [47] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé, Miro Dudik, and Hanna Wallach. 2019. Improving Fairness in Machine Learning Systems: What Do Industry Practitioners Need? (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–16. <https://doi.org/10.1145/3290605.3300830>
- [48] Xinlan Emily Hu, Rebecca Hinds, Melissa Valentine, and Michael S. Bernstein. 2022. A "Distance Matters" Paradox: Facilitating Intra-Team Collaboration Can Harm Inter-Team Collaboration. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW1, Article 48 (apr 2022), 36 pages. <https://doi.org/10.1145/3512895>
- [49] IBM. 2022. Carbon Design System. <https://carbondesignsystem.com/>.
- [50] Nanna Inie and Peter Dalsgaard. 2020. How interaction designers use tools to manage ideas. *ACM Transactions on Computer-Human Interaction (TOCHI)* 27, 2 (2020), 1–26.
- [51] Instructure, Inc. 2022. You. To the Power of the Canvas by Instructure. <https://www.instructure.com/canvas>.
- [52] Aminreza Iranmanesh and Zeynep Onur. 2021. Mandatory virtual design studio for all: Exploring the transformations of architectural education amidst the global pandemic. *International Journal of Art & Design Education* 40, 1 (2021), 251–267.
- [53] Alexander Jones and Volker Thoma. 2019. Determinants for successful agile collaboration between UX designers and software developers in a complex organisation. *International Journal of Human-Computer Interaction* 35, 20 (2019), 1914–1935.
- [54] Derek Jones, Nicole Lotz, and Georgy Holden. 2021. A longitudinal study of virtual design studio (VDS) use in STEM distance design education. *International Journal of Technology and Design Education* 31, 4 (2021), 839–865.
- [55] Young-Wook Jung, Youn-kyung Lim, and Myung-suk Kim. 2017. Possibilities and Limitations of Online Document Tools for Design Collaboration: The Case of Google Docs. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (Portland, Oregon, USA) (CSCW '17). Association for Computing Machinery, New York, NY, USA, 1096–1108. <https://doi.org/10.1145/2998181.2998297>
- [56] Gabriela Jurca, Theodore D Hellmann, and Frank Maurer. 2014. Integrating agile and user-centered design: A systematic mapping and review of evaluation and validation studies of agile-UX. In *2014 Agile conference*. IEEE, 24–32.
- [57] Tae Soo Kim, Seungsu Kim, Yoonseo Choi, and Juho Kim. 2021. Winder: Linking Speech and Visual Objects to Support Communication in Asynchronous Collaboration. 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 453, 17 pages. <https://doi.org/10.1145/3411764.3445686>
- [58] John King and Roger Magoulas. 2017. 2017 Design Salary and Tools Survey. <https://www.oreilly.com/content/2017-design-salary-survey/>.
- [59] Branko Kolarevic, Gerhard Schmitt, Urs Hirschberg, David Kurmann, and Brian Johnson. 2000. An experiment in design collaboration. *Automation in Construction* 9, 1 (2000), 73–81.
- [60] Emily Kuang, Xiaofu Jin, and Mingming Fan. 2022. "Merging Results Is No Easy Task": An International Survey Study of Collaborative Data Analysis Practices Among UX Practitioners. 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 318, 16 pages. <https://doi.org/10.1145/3491102.3517647>
- [61] Kati Kuusinen and Kaisa Väänänen-Vainio-Mattila. 2012. How to make agile UX work more efficient: management and sales perspectives. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*. 139–148.
- [62] Carine Lallemand, Guillaume Gronier, and Vincent Koenig. 2015. User experience: A concept without consensus? Exploring practitioners' perspectives through an international survey. *Computers in Human Behavior* 43 (2015), 35–48.
- [63] C. Lattemann and S. Stieglitz. 2005. Framework for Governance in Open Source Communities. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. 192a–192a. <https://doi.org/10.1109/HICSS.2005.278>
- [64] Claire Lauer and Eva Brumberger. 2014. Architect, Developer, Designer: The Anatomy of UX in Industry Job Postings. In *Proceedings of the 32nd ACM International Conference on The Design of Communication CD-ROM* (Colorado Springs, CO, USA) (*SIGDOC '14*). Association for Computing Machinery, New York, NY, USA, Article 19, 2 pages. <https://doi.org/10.1145/2666216.2666234>
- [65] Mina Lee, Percy Liang, and Qian Yang. 2022. CoAuthor: Designing a Human-AI Collaborative Writing Dataset for Exploring Language Model Capabilities. 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 388, 19 pages. <https://doi.org/10.1145/3491102.3502030>
- [66] Seunghyun Lee, Neta Ezer, Jon Sanford, and Ellen Yi-Luem Do. 2009. Designing together while apart: The role of computer-mediated communication and collaborative virtual environments on design collaboration. In *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 3593–3598.
- [67] Germán Leiva, Nolwenn Maudet, Wendy Mackay, and Michel Beaudouin-Lafon. 2019. Enact: Reducing Designer–Developer Breakdowns When Prototyping Custom Interactions. 26, 3, Article 19 (may 2019), 48 pages. <https://doi.org/10.1145/3310276>
- [68] Lassi A Liikkanen, Harri Kilpiö, Lauri Svan, and Miko Hiltunen. 2014. Lean UX: the next generation of user-centered agile development?. In *Proceedings of the 8th nordic conference on human-computer interaction: Fun, fast, foundational*. 1095–1100.
- [69] Zhicong Lu, Rubaiat Habib Kazi, Li-Yi Wei, Mira Dontcheva, and Karrie Karahalios. 2021. StreamSketch: Exploring Multi-Modal Interactions in Creative Live Streams. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (2021), 1–26.
- [70] Sarah BJ Macfarlane. 1996. Conducting a descriptive survey: 1. planning the survey. *Tropical doctor* 26, 4 (1996), 161–164.
- [71] Nolwenn Maudet, Germán Leiva, Michel Beaudouin-Lafon, and Wendy Mackay. 2017. Design Breakdowns: Designer-Developer Gaps in Representing and Interpreting Interactive Systems. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (Portland, Oregon, USA) (*CSCW '17*). Association for Computing Machinery, New York, NY, USA, 630–641. <https://doi.org/10.1145/2998181.2998190>
- [72] Kevin M Mendez, Leighton Pritchard, Stacey N Reinke, and David I Broadhurst. 2019. Toward collaborative open data science in metabolomics using Jupyter Notebooks and cloud computing. *Metabolomics* 15, 10 (2019), 1–16.
- [73] Albert Meroño-Peñuela, Dayana Spagnuolo, et al. 2020. Can a Transformer Assist in Scientific Writing? Generating Semantic Web Paper Snippets with GPT-2. In *European Semantic Web Conference*. Springer, 158–163.
- [74] Meta Platforms, Inc. 2022. React—A JavaScript Library for Building User Interfaces. <https://reactjs.org/>.
- [75] Microsoft. 2022. Fluent Design System. <https://www.microsoft.com/design/fluent/>.
- [76] Microsoft. 2022. OneDrive Personal Cloud Storage. <https://onedrive.live.com/>.
- [77] Miro. 2022. Miro: An Online Whiteboard & Visual Collaboration Platform. <https://www.miro.com/>.
- [78] Robert J. Moore, Eric Young Liu, Saurabh Mishra, and Guang-Jie Ren. 2020. Design Systems for Conversational UX. In *Proceedings of the 2nd Conference on Conversational User Interfaces* (Bilbao, Spain) (*CUI '20*). Association for Computing Machinery, New York, NY, USA, Article 45, 4 pages. <https://doi.org/10.1145/3405755.3406150>
- [79] Meredith Ringel Morris, Jazette Johnson, Cynthia L. Bennett, and Edward Cutrell. 2018. Rich Representations of Visual Content for Screen Reader Users. 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/10.1145/3173574.3173633>
- [80] Jakob Nielson. 2017. A 100-Year View of User Experience. <https://www.nngroup.com/articles/100-years-ux/>.
- [81] Don Norman. 2016. Don Norman on the term "UX". <https://www.nngroup.com/videos/don-norman-term-ux/>.
- [82] Don Norman and Jakob Nielsen. 2022. The Definition of User Experience (UX). <https://www.nngroup.com/articles/definition-user-experience/>.
- [83] Jeffrey Karl Ochsner. 2000. Behind the mask: a psychoanalytic perspective on interaction in the design studio. *Journal of Architectural Education* 53, 4 (2000), 194–206.
- [84] Jasper O'Leary, Holger Winnemöller, Wilmot Li, Mira Dontcheva, and Morgan Dixon. 2018. Charrette: Supporting In-Person Discussions around Iterations in User Interface Design. 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/10.1145/3173574.3174109>
- [85] Gary M Olson and Judith S Olson. 2000. Distance matters. *Human-computer interaction* 15, 2-3 (2000), 139–178.
- [86] Tina Øvad and Lars Bo Larsen. 2015. The prevalence of UX design in agile development processes in industry. In *2015 Agile Conference*. IEEE, 40–49.
- [87] João Pacheco, Stoyan Garbatov, and Miguel Goulão. 2021. Improving Collaboration Efficiency Between UX/UI Designers and Developers in a Low-Code Platform. In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, 138–147.
- [88] Taylor Palmer and Jordan Bowman. 2019. 2019 Design Tools Survey. <https://uxtools.co/survey-2019/>.
- [89] Taylor Palmer and Jordan Bowman. 2021. 2021 Design Tools Survey. <https://uxtools.co/survey-2021/>.
- [90] Evan Patterson, Robert McBurney, Holly Schmidt, Ioana Baldini, Aleksandra Mojsilović, and Kush R Varshney. 2017. Dataflow representation of data analyses: Toward a platform for collaborative data science. *IBM Journal of Research and Development* 61, 6 (2017), 9–1.
- [91] Cynthia Putnam and Beth Kolko. 2012. HCI Professions: Differences & Definitions. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems* (Austin, Texas, USA) (*CHI EA '12*). Association for Computing Machinery, New York, NY, USA, 2021–2026. <https://doi.org/10.1145/2212776.2223746>
- [92] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 8748–8763.
- [93] Alec Radford, Ilya Sutskever, Jong Wook Kim, Gretchen Krueger, and Sandhini Agarwal. 2021. CLIP: Connecting Text and Images. <https://openai.com/blog/clip/>.
- [94] Niko Reunanen, Zeynep Falay von Flitner, Virpi Roto, and Kirsikka Vaajakallio. 2020. Combining machine learning and Service Design to improve customer experience. In *Service Design and Service Innovation Conference*.
- [95] Kjeld Schmidt and Ina Wagner. 2002. Coordinative artifacts in architectural practice. In *COOP*. Citeseer, 257–274.
- [96] Koushik Sen, Darko Marinov, and Gul Agha. 2005. CUTE: A Concolic Unit Testing Engine for C. In *Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (Lisbon, Portugal) (*ESEC/FSE-13*). Association for Computing Machinery, New York, NY, USA, 263–272. <https://doi.org/10.1145/1081706.1081750>
- [97] Alex Serban, Koen van der Blom, Holger Hoos, and Joost Visser. 2020. Adoption and Effects of Software Engineering Best Practices in Machine Learning (*ESEM '20*). Association for Computing Machinery, New York, NY, USA, Article 3, 12 pages. <https://doi.org/10.1145/3382494.3410681>
- [98] Hanieh Shakeri, Carman Neustaedter, and Steve DiPaola. 2021. SAGA: Collaborative Storytelling with GPT-3. In *Companion Publication of the 2021 Conference on Computer Supported Cooperative Work and Social Computing* (Virtual Event, USA) (*CSCW '21*). Association for Computing Machinery, New York, NY, USA, 163–166. <https://doi.org/10.1145/3462204.3481771>
- [99] Teo Yu Siang. 2021. The Ultimate Guide to Understanding UX Roles and Which One You Should Go For. <https://www.interaction-design.org/literature/article/the-ultimate-guide-to-understanding-ux-roles-and-which-one-you-should-go-for>.
- [100] Sketch B.V. 2022. Everything you need to design. <https://www.sketch.com/>.
- [101] Sketch B.V. 2022. Work together. Go far. <https://www.sketch.com/collaborate/>.
- [102] Micah J Smith, Roy Wedge, and Kalyan Veeramachaneni. 2017. FeatureHub: Towards collaborative data science. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 590–600.
- [103] Erik Stolterman and James Pierce. 2012. Design Tools in Practice: Studying the Designer-Tool Relationship in Interaction Design (*DIS '12*). Association for Computing Machinery, New York, NY, USA, 25–28. <https://doi.org/10.1145/2317956.2317961>
- [104] Hariharan Subramonyam, Colleen Seifert, and Eytan Adar. 2021. ProtoAI: Model-Informed Prototyping for AI-Powered Interfaces. In *26th International Conference on Intelligent User Interfaces* (College Station, TX, USA) (*IUI '21*). Association for Computing Machinery, New York, NY, USA, 48–58. <https://doi.org/10.1145/3397481.3450640>
- [105] Hariharan Subramonyam, Colleen Seifert, and Eytan Adar. 2021. *Towards A Process Model for Co-Creating AI Experiences*. Association for Computing Machinery, New York, NY, USA, 1529–1543. <https://doi.org/10.1145/3461778.3462012>
- [106] Svelte. 2022. Svelte · Cybernetically Enhanced Web Apps. <https://svelte.dev/>.

- [107] Ben Swanson, Kory Mathewson, Ben Pietrzak, Sherol Chen, and Monica Dinulescu. 2021. Story Centaur: Large Language Model Few Shot Learning as a Creative Writing Tool. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. 244–256.
- [108] Tactivos, Inc. 2022. MURAL is a collaborative intelligence company. <https://www.mural.co/>.
- [109] Jaime Teevan, Shamsi T. Iqbal, and Curtis von Veh. 2016. Supporting Collaborative Writing with Microtasks. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (*CHI '16*). Association for Computing Machinery, New York, NY, USA, 2657–2668. <https://doi.org/10.1145/2858036.2858108>
- [110] Carola Fellenz Thompson, Richard I. Anderson, Irene Au, Cordell Ratzlaff, and Nida Zada. 2010. Managing User Experience: Managing Change. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (Atlanta, Georgia, USA) (*CHI EA '10*). Association for Computing Machinery, New York, NY, USA, 3143–3146. <https://doi.org/10.1145/1753846.1753938>
- [111] UX Collective. 2022. The State of UX in 2022. <https://trends.uxdesign.cc/>.
- [112] Nick van den Berg. 2022. Tweets from Nick van den Berg. <https://perma.cc/5W7G-E6AQ>.
- [113] Visual Studio Code. 2022. Collaborate with Live Share. <https://code.visualstudio.com/learn/collaboration/live-share>.
- [114] Dhaval Vyas, Gerrit van der Veer, and Anton Nijholt. 2013. Creative practices in the design studio culture: collaboration and communication. *Cognition, Technology & Work* 15, 4 (2013), 415–443.
- [115] Jagoda Walny, Christian Frisson, Mieka West, Doris Kosminsky, Søren Knudsen, Sheelagh Carpendale, and Wesley Willett. 2020. Data Changes Everything: Challenges and Opportunities in Data Visualization Design Handoff. *IEEE Transactions on Visualization and Computer Graphics* 26, 1 (2020), 12–22. <https://doi.org/10.1109/TVCG.2019.2934538>
- [116] April Yi Wang, Anant Mittal, Christopher Brooks, and Steve Oney. 2019. How data scientists use computational notebooks for real-time collaboration. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–30.
- [117] April Yi Wang, Zihan Wu, Christopher Brooks, and Steve Oney. 2020. Callisto: Capturing the "Why" by Connecting Conversations with Computational Narratives. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.
- [118] Dakuo Wang, Haodan Tan, and Tun Lu. 2017. Why Users Do Not Want to Write Together When They Are Writing Together: Users' Rationales for Today's Collaborative Writing Practices. 1, CSCW, Article 107 (dec 2017), 18 pages. <https://doi.org/10.1145/3134742>
- [119] Jeremy Warner and Philip J. Guo. 2017. CodePilot: Scaffolding End-to-End Collaborative Software Development for Novice Programmers (*CHI '17*). Association for Computing Machinery, New York, NY, USA, 1136–1141. <https://doi.org/10.1145/3025453.3025876>
- [120] Qian Yang, Alex Scuito, John Zimmerman, Jodi Forlizzi, and Aaron Steinfeld. 2018. Investigating How Experienced UX Designers Effectively Work with Machine Learning. In *Proceedings of the 2018 Designing Interactive Systems Conference (DIS '18)*. Association for Computing Machinery, New York, NY, USA, 585–596. <https://doi.org/10.1145/3196709.3196730>
- [121] Qian Yang, Aaron Steinfeld, Carolyn Rosé, and John Zimmerman. 2020. Re-Examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. 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, 1–13. <https://doi.org/10.1145/3313831.3376301>
- [122] Evan You. 2022. Vue.js - The Progressive JavaScript Framework. <https://vuejs.org/>.
- [123] Amy X. Zhang, Michael Muller, and Dakuo Wang. 2020. How Do Data Science Workers Collaborate? Roles, Workflows, and Tools. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW1, Article 22 (may 2020), 23 pages. <https://doi.org/10.1145/3392826>