"A Lot of Moving Parts": A Case Study of Open-Source Hardware Design Collaboration in the Thingiverse Community

KATHY CHENG, University of Toronto, Canada

SHURUI ZHOU, University of Toronto, Canada

ALISON OLECHOWSKI, University of Toronto, Canada

Open-source is a decentralized and collaborative method of development that encourages open contribution from an extensive and undefined network of individuals. Although commonly associated with software development (OSS), the open-source model extends to hardware development, forming the basis of open-source hardware development (OSH). Compared to OSS, OSH is relatively nascent, lacking adequate tooling support from existing platforms and best practices for efficient collaboration. Taking a necessary step towards improving OSH collaboration, we conduct a detailed case study of *DrawBot*, a successful OSH project that remarkably fostered a long-term collaboration on Thingiverse – a platform not explicitly intended for complex collaborative design. Through analyzing comment threads and design changes over the course of the project, we found how collaboration occurred, the challenges faced, and how the DrawBot community managed to overcome these obstacles. Beyond offering a detailed account of collaboration practices and challenges, our work contributes best practices, design implications, and practical implications for OSH project maintainers, platform builders, and researchers, respectively. With these insights and our publicly available dataset [20], we aim to foster more effective and efficient collaborative design in OSH projects.

CCS Concepts: • **Human-centered computing** → **Empirical studies in collaborative and social computing**; • **Applied computing** → *Computer-aided design*.

Additional Key Words and Phrases: Collaborative Design; Open-source; 3D Printing; Remixing

1 Introduction

From smartphones to automobiles, coffee makers to pacemakers, physical products are indispensable in our daily lives. Designing and developing these products successfully requires effective collaboration among multiple individuals [8, 80]. Complex hardware development has been traditionally restricted to professional engineering firms, performed by design teams operating within formal organizational structures and commercial incentives [100]. However, the open-source model has recently emerged as a promising alternative means of hardware development.

Open-source is a decentralized and collaborative method of development that encourages open contribution from an extensive and undefined network of individuals [40, 74]. Contributors are

Authors' Contact Information: Kathy Cheng, University of Toronto, Toronto, Canada, kathy.cheng@mail.utoronto.ca; Shurui Zhou, University of Toronto, Toronto, Canada; Alison Olechowski, University of Toronto, Toronto, Canada.

distributed physically in time and space, membership is transient (which allows contributors to come and go as they please), and agents are semi-autonomous in their work [69]. Open-source communities have the freedom to follow non-profit motivations [12], which may lead to design innovations for under-served or low-power populations.

The open-source model is often associated with software development (OSS), and this success [76] has inspired product designers to seek a similar approach to hardware development (i.e., OSH) [47]. Although the goals of open-source are shared among hardware and software, the artifacts involved in the design processes are distinct; while OSS mainly relies on text-based source code, OSH projects involve 3D CAD (computer-aided design) models that represent geometric and topological data [19, 93]. Due to differences in the artifacts, OSH requires considerations for design representation, materials, and manufacturing [5]. Therefore, while lessons on collaboration effectiveness and efficiency from OSS are relevant to OSH, the tools and practices cannot be directly applied.

Many hardware designers are enthusiastic about OSH [13]; however, the potential for complex collaborative design is currently limited by the lack of widely-accepted, dedicated groupware for OSH [15, 64]. As a result, some hardware designers have resorted to using social coding platforms like GitHub [13, 26], despite not being intended for hardware artifacts like 3D CAD models. Predictably, a software-focused platform is ill-suited for hardware development and introduces pain points and collaboration inefficiencies, such as cumbersome 3D model manipulation [15], poor granularity for branch/merge [18], and lack of synchronous/asynchronous communication channels [64]. As OSH expands, it is imperative to devise dedicated strategies to facilitate collaboration within OSH projects, rather than inheriting existing systems from OSS.

Platforms dedicated to hardware do exist, in the form of *collective design* communities (e.g., Thingiverse,[1] Thangs3D[2]), as opposed to *collaborative design* communities. The primary aim of these platforms is for sharing designs for 3D printed artifacts, rather than supporting a complex collaborative design process [25]. However, anecdotal evidence suggests that makers are indeed trying to engage in collaborative design activities through reusing existing designs – stretching the collective design platform's capabilities to enable coordinated action – despite the limited collaboration features (e.g., poor version tracking, lack of multi-user CAD) [55, 73, 82].

In light of the growing interest in collaborative OSH and anecdotes of successful projects in spite of the insufficient support from existing platforms, there is a crucial need to comprehend how collaboration unfolds in OSH projects. Current empirical research on OSH has focused on general patterns of design in collective design communities, and pain points faced by individual makers [3, 50]. Although these studies are useful for characterizing high-level practices, they lack in-depth exploration of the dynamics, processes, or inefficiencies specific to collaboration. Without this understanding, it remains unclear how to develop best practices or dedicated tooling support for efficient and effective OSH collaboration.

Our goal is to close this knowledge gap so that we can better understand and improve the way that makers collaborate on OSH designs. We first need to provide a descriptive analysis of the current practices that enable successful collaborative design, and then we can identify the challenges faced in the community. The following two research questions guided our work:

RQ1: How do designers collaborate during a successful open-source hardware (OSH) project?
 RQ1.1: What tasks involved collaboration?
 RQ1.2: How did designers collaborate to achieve those tasks?
RQ2: How is OSH collaboration hindered, and what are the opportunities for improvement?

---

[1]https://thingiverse.com/

[2]https://thangs.com/

To address these questions, we conducted a detailed case study of *"DrawBot"*, an OSH project hosted on Thingiverse, the largest online 3D printing community [96]. Although Thingiverse is primarily a collective design platform – and not explicitly intended for complex OSH collaboration – the case we present serves as a rare instance of thriving collaborative design. The case study approach enables a retrospective analysis of a collaborative, long-lived, and well-maintained OSH project, unveiling the aspects contributing to the project's success, and areas for improvement in future projects. Focusing on a project-level analysis allows for a deeper investigation of the collaborative interactions among participants, providing longitudinal insights into how both the design and community evolved over the project's lifetime. While previous research has applied the case study method to examine open-source collaboration in software communities [41, 56, 65], the exploration of OSH collaboration practices is relatively nascent. To our knowledge, our study is the first to investigate collaborative design within a hardware-focused community. Therefore, we make the following contributions to CSCW:

(1) An in-depth case study of a unique collaborative OSH design project that successfully overcame tooling limitations to flourish on a collective design platform.

(2) An investigation of how makers collaborated on both design-related and non-design-related tasks, offering best practices for the maintainers and participants of OSH projects to collaborate efficiently.

(3) An identification of the tooling support limitations which harm collaborative awareness and impede the full potential of OSH collaboration. As a first step towards addressing these challenges, our work contributes several design implications for OSH platform builders.

(4) A novel approach for analyzing collaborative design projects on collective design platforms. To facilitate further exploration of OSH collaboration, we provide practical insights for researchers conducting similar case studies, as well as our publicly available dataset (found here [20]).

2 Background & Related Work

2.1 Open Source Hardware Development (OSH)

Open-source hardware development (OSH) allows free and open participation from a distributed, asynchronous, and large network of users [107]. Products developed through OSH efforts are physical artifacts that are free for the public to: (1) *study* (i.e., access sufficient information to understand a product's use and design), (2) *modify* (i.e., create product derivatives), (3) *make* (i.e., physically fabricate and manufacture the product), and (4) *distribute* (i.e., share or sell the physical product or associated documentation) [75].

There is no single, widely-accepted platform for OSH, comparable to GitHub's widespread use in OSS [29]. OSH occurs on various platforms, such as Thingiverse [9, 17, 73], GitHub [13, 26], and Instructables [99]. OSH projects typically leverage an assortment of tools for collaboration, communication, or version control [14]. Mies et al. evaluated five popular platforms used for OSH based on their support of communication, coordination, knowledge integration, and information logistics [64].[3] GitHub narrowly outperformed the other OSH-specific options, despite not being designed to support collaboration on 3D CAD models. In fact, none of the five platforms offered collaborative modelling capabilities [64].

2.1.1 Collective Design vs. Collaborative Design. OSH practices have been classified in the literature as following a variety of working styles, such as, open design [7, 98], collective design [48, 69, 79, 107], peer-to-peer design [13], social product design [1], and collaborative design [26, 93]. While

---

[3]The five platforms analyzed by Mies et al. were GitHub, MediaWiki, GrabCAD, Wevolver, and Wikifactory.

there is ambiguity in the terminology of characterizing types of OSH projects, scholars generally agree that OSH projects can be broadly categorized into two non-mutually exclusive types, which we call *collective design* and *collaborative design*.

Collective design occurs when large numbers of people engage in design activity to accomplish their individual goals [69]. Designers may only know each other through their designs, and communicate through loosely formed and informal design networks [69]. On collective design platforms (e.g., Thingiverse, Thangs3D), users can freely post designs, contribute to other users' designs, and build upon each others' designs in a sequential and derivative manner [107]. While collaboration occurs in the sense that the design process involves multiple people, these processes do not involve coordinated action that evolves the design towards a shared goal [13].

In contrast, collaborative design involves the joint efforts of many designers to address design goals, requiring coordination and cooperation [79, 98]. Products designed in this manner are typically complex – in both product architecture and required effort [51], mimicking products developed in industrial product design teams [14], thus necessitating collaborative design. Unlike in formal design organizations, participants of open-source collaborative design are distributed in time and space, self-organized, and transient in their level and duration of involvement. Nonetheless, a common motivation, communication channels, and shared representation (i.e., design artifacts) facilitate the ongoing collaborative design process [62].

Previous research has investigated the evolutionary design patterns within collective design platforms (as we will review in Section 2.2), however, there are few empirical accounts of collaborative design in OSH. Müller-Seitz and Reger documented a case study of the OScar project, an open-source effort to design and develop a functional car through the Internet [67]. The study demonstrated that tooling support, in the form of discussion boards, was of utmost importance to the coordination of the project. However, both the project and the investigation predated many ICT (information and communication technology) and CMC (computer-mediated communication) tools that are available today, such as wiki software, video-sharing platforms, or blogging platforms [67]. An updated case study is needed to describe current OSH collaboration practices and challenges.

More recently, Mellis and Buechley explored OSH collaboration on the Arduino microcontroller development platform, making comparisons to OSS [63]. The Arduino community resembled an ecosystem of many small-scale collaborations among groups of contributors. This decentralized nature was reflected in the design process, where contributors created many design alternatives, rather than improving one single, central option. Making alternative designs was considered a positive contribution to the community, in contrast to OSS, where divergent repositories, or forks, are often seen as resulting from disagreement or conflict among developers [63, 104], which can fragment the community [106].

2.1.2 Collaborative Design Challenges in OSH. There is a desire for collaborative OSH, but there is not yet a suitable platform to seamlessly accommodate collaborative design. Some past studies have pointed out the challenges of OSH, which we summarize below.

OSH projects inherently involve diverse artifacts, encompassing both physical and digital components, which distinguish them from OSS projects. While OSS users can compile code at virtually zero cost to produce the final product, OSH users face challenges related to the cost of materials and manufacturing for physical components [5, 45, 49]. These considerations not only apply to the end product, but also at various prototyping stages, which may discourage makers from continuously testing new design iterations [13]. Furthermore, the physical nature of hardware products prevents distributed collaborators from being able to test each other's design changes [63].

Collaboration is also challenging with the digital artifacts in OSH projects, such as 3D CAD models. The lack of in-browser integration of 3D models is a frequently cited barrier to OSH

collaboration [15]. Current platforms offer little opportunity for viewing and manipulating CAD models (e.g., rotation, annotation, exploded view), interoperability (compatibility with various 3D file formats), and online concurrent editing [15].

Beyond artifact-related challenges, OSH faces many process-related challenges that hinder collaborative awareness [28]. One major challenge is the lack of version control systems dedicated to OSH, resulting in the adoption of tools intended for software development. These tools simply store subsequent versions of a file without offering mechanisms to summarize changes between versions [63]. The inadequate tracking of design history versions makes it difficult to reuse models [30] – which is particularly harmful to OSH projects, where the creation of derivatives from existing designs is a fundamental practice [63, 107].

Documentation is of utmost importance to open-source collaboration, allowing for a simple way to make knowledge accessible to a large community [45]. Compared to OSS, OSH employs a diverse range of documentation types, encompassing images, CAD files, and videos [61]. This diversity adds complexity to collaboration since processes with physical artifacts (e.g., assembly) are not inherently captured in a digital format [99].

Finally, the absence of one universally accepted OSH platform results in collaboration occurring simultaneously on different platforms, such as social media, forums, blogs, GitHub, or Thingiverse [26]. With the scattered nature of the project, OSH participants must navigate all of these different sites, which can harm collaboration efficiency.

Overall, existing research has characterized OSH practices, offering two conceptions: collective design and collaborative design. OSH has primarily fostered collective design, where individuals are accomplishing their own goals on the platform, whereas less attention has been given to collaborative design, which faces many coordination and cooperation challenges. To address this problem, our case study contributes a detailed investigation of a collaborative design project, which, despite being conducted on a collective design platform, has successfully fostered a long-term, complex collaboration. Through our study, we not only document the existing practices that contributed to the project's success, but also propose improvements to further strengthen OSH design processes.

2.2 Thingiverse

In this work, we focus on open-source hardware development on the Thingiverse platform. Thingiverse self-describes as a "thriving design community for discovering, making, and sharing 3D-printable things," and is the largest online maker community for 3D CAD models [3, 54, 96]. As of November 2023, Thingiverse contains over 2.5 million published designs [96], with steady growth since its inception in January 2009 (Figure 1). Due to this popularity, Thingiverse has long been the platform of interest for researchers seeking to better understand OSH practices. In this section, we first briefly introduce several concepts of the platform, followed by a review of related Thingiverse literature.

2.2.1 Background. Thingiverse's primary purpose is to provide an open online repository of 3D design models, known as *"Things"* [32]. Each Thing includes a variety of metadata: name, author's username, publication date, description (analogous to a README file), and uploaded files (e.g., CAD files, images, documentation) – see Figure 3 for a snapshot of a Thing page. The author must provide all of this metadata to publish the design. Users can interact with Things through likes, downloads, comments, makes, and remixes – a *"make"* refers to a user-uploaded photo or description showcasing their successful creation of a Thing, and a *"remix"* is a user-generated Thing derived from one or more existing Things.
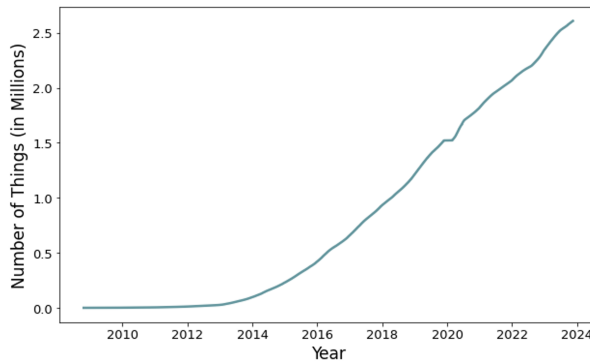
Fig. 1. Cumulative number of Things published on Thingiverse from January 2009 to November 2023.

Remixing resembles "forking" in software repositories, which allows makers to copy and modify a design to fit their needs, while formally establishing traceability between related designs, since the platform tracks remixing relationships. The majority of designs on Thingiverse are shared under open licenses (e.g., Creative Commons) that explicitly allow remixing adaptations [35].

2.2.2 Related Work. This section provides an overview of relevant studies on design practices within the Thingiverse platform, covering how makers engage with Things, the creation of derivative Things (remixes), and the overall experiences of Thingiverse makers. Prior work has focused on scraping metadata of Things (e.g., likes, views) to understand the types of designs on Thingiverse [107], characteristics of popular designs [77, 78], and user engagement dynamics [71, 72]. Novak et al. conducted a longitudinal study on Thingiverse user interactions on Thingiverse, revealing that commenting is associated with experienced makers [72], since commenting requires a deeper understanding of the design and motivation to actively engage with it [95]. Remixes were high-effort actions, requiring 3D CAD skills, creativity, and a considerable time investment [72].

Remixing. Remixing plays a critical role in Thingiverse; in fact, more than half of Things are derived from prior Things [31, 42, 107]. This significant practice has been of particular interest to HCI and CSCW researchers [91]. Oehlberg et al. conducted an in-depth analysis of remixing behaviour in Thingiverse, to explore how remixing activity was impacted by *Customizer*[4] – a Thingiverse-integrated built-in web app that allows simple modifications of 3D designs using pre-defined parameters [73]. Things created with Customizer did not elicit subsequent user activity, and the authors of these Things did not participate in experimental, creative, or collaborative projects [73]. This finding is consistent with Flath et al.'s investigation of "shallow" (low degree of change) and "deep" (high degree of change) remixes [31, 103]. Although these studies are enlightening, there are limitations with their coarse method of classifying "deep" or "shallow" remixes, and manual coding of remix relationships was deemed necessary for future work [31].

In addition to analyzing patterns of remixes, researchers have also explored the relationships between remixing and innovation [6, 94, 95]. When investigating the types of added value in remixed designs, Voigt found that remixes do not necessarily always contribute new design elements (referred to as "feature-driven innovations"), but they can also streamline the design reuse process through linking related designs [2], or providing different file types of the same design (called "production process-driven innovations") [101].

---

[4]http://thingiverse.com/apps/customizer

While researchers have explored remixing patterns and innovation, they have predominantly focused on the incremental changes between remixed designs. What remains unexplored is how remixing occurs in a collaborative project, where contributions made in remixes can contribute to the overall evolution of a design project. Our study aims to address this gap and shed light on the collaborative contributions of remixing.

Makers. While many researchers have studied the designs published on Thingiverse, others have focused on the designers themselves, understanding practices, such as design discovery [57], license choices [66], motivations for participation [34, 46, 87, 89], and how motivation impacts engagement [89, 107]. Makers can be active contributors to the design ecosystem, or more passive "lurkers" who observe and read other members' content without contributing their own [87]. Seeking to comprehend the diverse user roles on Thingiverse, Li et al. studied user activity in a Thingiverse *group.*[5] They found that the most active members contribute most to the group (in posts and likes), but it is unclear whether roles would manifest differently in collaborative projects where members work towards shared design goals [55].

Regardless of motivation, type, or activity, all makers will face challenges in the design and development process. Two pain points identified in existing work include tracking design versions and navigating the dependencies between related designs, which can pose obstacles to productive remixing practices [73]. Alcock et al. identified pain points with understanding product functionality and customizability, proposing corresponding design recommendations for platform builders to develop beginner-friendly CAD tools [3]. While these studies provide insights into the pain points experienced by individual makers, we still lack an understanding of the challenges that impede productive collaborative design efforts.

Previous research has examined various aspects of open-source hardware on Thingiverse, such as published designs, remixing practices, and the experiences of makers. These studies provide large-scale overviews of metadata, thus offering insights into Thingiverse as a collective design platform. However, there is a lack of investigations of collaborative design on Thingiverse, which would require coordinated action and may face different challenges compared to collective design (see Section 2.1). As such, an exploratory study is needed to investigate collaborative design practices in a successful and well-maintained OSH project.

3 Methods

Our goal is to understand how makers collaborate in an OSH project within a collective design community. To do this, we conducted a detailed case study of designs hosted on Thingiverse. We chose a case study approach because it is the most suitable method for answering the "how?" question in a contemporary phenomenon [105], such as OSH collaboration. Furthermore, case studies are the preferred method for exploring a phenomenon that is closely tied to its real-world context [105] – in our case, a collaborative project situated within Thingiverse. Separating the design process from the context of Thingiverse is neither feasible nor desirable, as our aim is to comprehend how collaboration unfolds on a traditionally collective design platform.

Our project-level analysis allows us to attain a comprehensive understanding of the design and development process from the project's inception. By doing so, we can glean insights into exactly how collaboration occurred – answering questions such as who was involved, the duration of their involvement, the challenges encountered, and the contributions that resulted from the collaborative efforts. Focusing on one self-contained project allows us to grasp the nuances of collaborative

---

[5]*Groups* are self-governed sub-communities within Thingiverse where makers gather based on shared interests. Group members can discuss in forum posts, publish designs, and connect with other members of the group.

processes, thus addressing our research goals. Thus, a case study of a single project is the ideal way to explore OSH collaboration practices.

Before delving further into our methodology, it is important to explain our definition of a "project". OSS projects on GitHub are rather straightforward – generally, a repository represents one project [16]. However, on Thingiverse, projects are not well-defined. Our case centres around one particular Thing posted on Thingiverse, but we expand the scope of the project to encompass the individual Thing as well as all related Things in ancestors and remixes. This decision was prompted by the platform's constraints, wherein only the original author of a Thing can modify it. Unlike in OSS, on Thingiverse, collaborators cannot contribute to the project through pull requests. However, Thingiverse's remix feature allows us to capture forking behaviour, to an extent. Thus, while we are interested in one particular Thing, we considered all of the ancestor Things, as well as all remixed Things, as part of the complete project. To the best of our knowledge, our study is the first to consider this network of designs as a project on Thingiverse.

In this section, we first provide a rationale for our project selection, followed by an overview of our data collection, processing, and analysis methods. Figure 2 summarizes our research method.
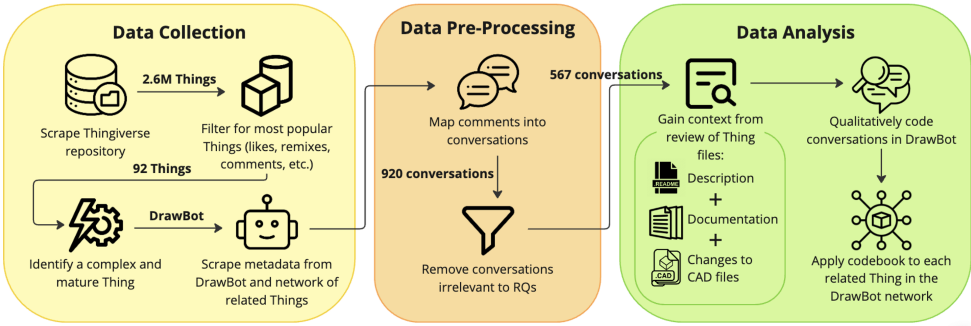

Fig. 2. Overview of research methods, from data collection to analysis.

3.1 Project Selection

In this study, the project we focus on is the *"Drawing Robot - Arduino Uno + CNC Shield + GRBL"*.[6] The Drawing Robot – or DrawBot – is a pen plotter device designed to create detailed 2-dimensional illustrations from pre-programmed sequences of movements provided by the user. Our selection of the DrawBot for this case study followed the steps outlined in Figure 2 and is summarized below.

First, we aimed to select an active and popular project, with high engagement among Thingiverse users. To do this, we filtered Things to obtain the 99th percentile in engagement metrics (i.e., likes, views, and downloads) and metrics that signify high-quality designs (i.e., number of comments, threads, makes, files, remixes, and ancestors); in total, 92 Things were in the 99th percentile across all nine metrics. We explain how to interpret these metrics below:

- *Likes, views, and downloads:* Higher numbers signify more activity without known tradeoffs in innovation or complexity.
- *Comments and makes:* Higher numbers signify more activity, but comments suggest experienced makers [3, 72] – so, we prioritize a high comment and thread count.
- *Remixes:* A higher count indicates a more popular Thing, but we avoid heavily remixed Things (100+ remixes) that might signify mass customization with low complexity [11, 101].

---

[6]https://thingiverse.com/thing:2349232

- *Ancestors:* Due to the inverted U-shaped relationship between the number of ancestors and innovativeness [6, 95], we avoid Things with no ancestors, and Things with many ancestors.

Overall, these descriptive statistics help us to maximize a Thing's activity and innovativeness while avoiding potential pitfalls associated with projects scoring highest on these metrics. After obtaining 92 things that met our filtering criteria, we manually inspected them for suitability based on the following considerations: product complexity and project maturity.

The chosen product needed to be complex to offer us a rich and comprehensive investigation of design collaboration. Consequently, complexity comprised three criteria: (1) the product must serve a practical purpose and function to avoid trivial items (e.g., figurines); (2) the product must be an assembly – products consisting of multiple parts that interface and interact [70]; and (3) the product should be electro-mechanical (i.e., incorporate both hardware and software components). In recent years, electronic toolkits (e.g., Arduinos) have become a staple in many open hardware projects [27, 63, 68, 90], allowing makers to create software-controlled devices (e.g., robots) [23]. These electro-mechanical components require makers to design and configure both hardware and software, and are associated with more complex products [51].

Lastly, the project must have been published for at least several years (we arbitrarily selected 4), since open-source hardware projects often require *"several years [to] indicate success"* [4], with long-term sustained active involvement signalling a well-maintained project. Simultaneously, the project should still be active (i.e., growing in comments and remixes), reflecting ongoing refinement. However, a slowing growth in activity signals that the project has entered a mature stage, with the primary development work likely completed. By selecting a project that has been active over several years, we can observe a more in-depth understanding of the collaborative design process.

In summary, our selection process ensured the Drawbot's high user engagement, sufficient complexity, and long-term sustained activity.

3.2 Data Collection and Pre-Processing

We used Thingiverse's public API to extract metadata from all published Things, totalling 2,608,691 as of November 2023. Scraping all Things was necessary to filter projects for our selection process.

3.2.1 Collecting metadata for each Thing. Once the DrawBot project was selected, we used the Thingiverse API to scrape the relevant metadata presented in the DrawBot Thing page as shown in Figure 3. We also collected all metadata for each Thing in the project (DrawBot's ancestors and children). Figure 4 provides a network of the 74 Things in the DrawBot project, where nodes are Things and edges are remixing relationships (pointing from parent Thing to child Thing); four Things are direct ancestors of the DrawBot, 31 Things are direct remixes from the DrawBot, and the remaining 38 Things are either ancestors or children of the directly related Things. Although we collected all of this metadata, our analysis focused on the comments, descriptions, and posted files, which typically included 3D CAD models and documentation. We have made our full dataset publicly available for use by the research community [20].

Of the 74 related Things in the DrawBot project, 31 Things had comments, totalling 2,483 comments. The average length of a comment was 48 words (SD = 58). Across 74 Things, there were 50 unique authors and 684 unique commenters. On average, each author made 19 comments (SD = 92), and non-author commenters made 2.3 comments (SD = 3.0). The average length of a description was 168 words (SD = 392); the DrawBot description itself was the longest, at 3,179 words. The average file count was 5 (SD = 5).

3.2.2 Collecting and preprocessing comments. In order to understand discussions among DrawBot contributors, we collected comments from each Thing (component D in Figure 3). Note that the
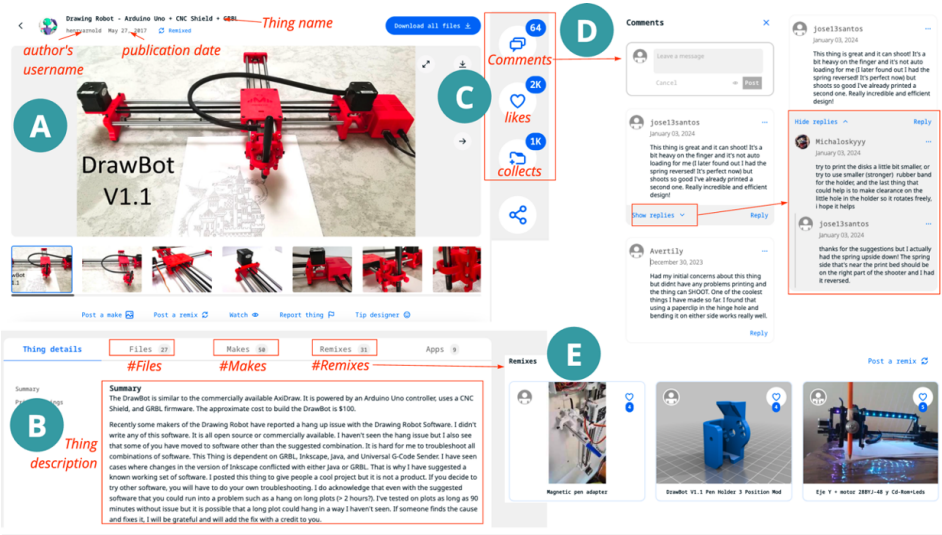
Fig. 3. Drawing Robot "Thing" studied in this paper. The metadata includes: (A) Thing name, author's username, and publication date; (B) Thing description, number of files, makes, and remixes; (C) user activity (e.g., likes, comments, collects); (D) the comment panel, with commenter usernames, posted dates, and threaded replies; (E) the remix panel, including remix activity with page links and author usernames.
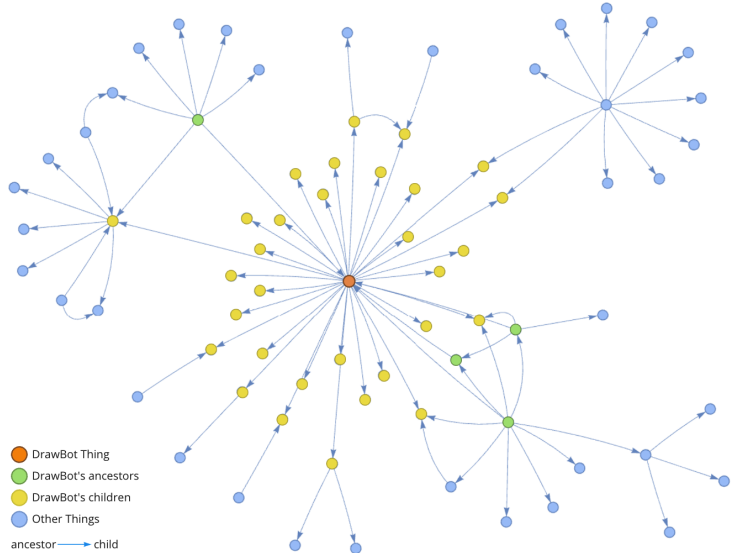
Fig. 4. Directed network of Things within the DrawBot project. The direction of the arrows indicates a remix relationship, pointing from ancestor Thing to child Thing. The orange node represents the DrawBot Thing; yellow nodes represent DrawBot remixes; green nodes represent DrawBot ancestors; and blue nodes represent other Things within the network indirectly connected to the DrawBot Thing (i.e., 2 or more nodes away).

comment panel lists comments linearly in chronological order. Users could reply to a comment to have a cohesive conversation. Thus, we iteratively collect all the comments and nested replies.

Constructing discussion threads. After collecting all comments, we read through each comment chronologically to map conversations together. Although Thingiverse has a threaded reply feature, it was not used consistently, necessitating an initial read-through. We define a conversation as an exchange between multiple people who are discussing the same topic; for example, in Figure 3, two makers are discussing a troubleshooting topic and replying directly to each other. In total, there were 920 conversations. Throughout this process, we also tagged conversations for relevance to our research questions; conversations solely expressing praise or appreciation, such as *"I love your machine and your artwork"* or *"Thank you so much for your time,"* were deemed irrelevant and discarded; of the 920 conversations, 353 were discarded. In the remaining 567 relevant discussions, there were 402 unique commenters.

3.3 Qualitative Data Analysis

Following data processing, we began the qualitative content analysis [53] in order to determine how designers collaborated in the project. One coder conducted the analysis.

After an initial examination of the DrawBot Thing description and documentation to understand the design and function of the device, the first author independently coded the conversations of the DrawBot Thing (567 conversations, comprising 1,540 comments) using a bottom-up coding approach [22, 84], because there is no widely accepted categorization of collaborative contributions in OSH, and the exploratory nature of our research questions. For each conversation, they coded: (1) the component of the DrawBot to which the conversation pertained, and (2) the collaborative contribution made by the commenter in the conversation.

To guide the coding of the component, they referred to a comprehensive parts list provided by the DrawBot author. This document outlined all the necessary components for constructing and operating the DrawBot, categorizing them into hardware, electronics, or software. Our codebook for relevant components was structured based on this detailed documentation (see Table 2 in Appendix A).

In addition to the project component, the first author also coded the task accomplished in the conversation. They focused on discussions resulting in productive outputs or the generation of new knowledge. Many codes emerged from commenters suggesting ways to improve various DrawBot components; others provided troubleshooting help for makers struggling with printing or building their DrawBot; and some codes were developed because makers identified problems with the design or fit of parts.

After the first round of open coding, a second round was conducted to ensure consistency between comments analyzed at the beginning and end of the process. A third round involved axial coding to group similar codes, identifying higher-level themes. This process required iteration between the comment data and related literature to refine the theme categorization, which is characteristic of qualitative research analysis [21]. The themes categorized tasks into *design-related* and *non-design-related* contributions.

Once the codebook had been established, a fourth and final round of coding applied the established codebook to other related Things in the DrawBot project (200 conversations, comprising 380 comments). Similar to our process of coding comments of the DrawBot Thing, we also read the Things' descriptions, analyzed their attached files, and used CAD software to compare versions of the same CAD models, to understand the context of these comments.

4 Results

In this section, we present the findings of the DrawBot case study. First, we provide an overview of the collaborative design activity in the project. Next, we elaborate on how makers collaborated on both design-related and non-design-related tasks. Lastly, we present the collaboration challenges

faced and how the community overcame them. As per ethical guidelines for internet research [33], we have paraphrased all quotes used in this paper to protect the anonymity and searchability of Thingiverse users.

4.1 Collaboration Practices (RQ1)

In this section, we provide a detailed account of collaboration practices in the DrawBot project, shedding light on the types of tasks requiring collaboration (RQ1.1) and how makers collaborated to complete them (RQ1.2). Through analyzing comments and remixes, we found that makers collaborated to fulfil: *design-related* and *non-design-related* contributions (summarized in Table 1). In the subsequent sections, we describe collaboration scenarios that occurred in the project.

Table 1. Types of collaborative contributions found in the comments. Contributions to open-source hardware collaboration can be *design-related* or *non-design-related*. We also report the percentage of conversations that discussed the contributions. Note that the percentages do not add to 100% since one conversation could discuss many topics.

Category	Sub-category	Description	Example(s) from Dataset	%
Design-Related	New Feature	Designing and developing new functionality for the design.	*"I am adapting DrawBot to draw with 12 colors."*	10.8
	Error Fix	Modifying components to correct issues or errors in the design.	*"You need to modify the corrupted line in the MI GRBL Extension."*	4.3
	Performance Enhancement	Optimizing the design to improve efficiency (e.g., power, speed).	*"You could lower the mass so that the drawing robot can move faster."*	2.8
	Component Replacement	Substituting elements with alternative components.	*"Could a Raspberry Pi be used in place of an Arduino?"*	6.2
Non-Design-Related	Issue Tracking & Management	Reporting, updating, and requesting support for issues.	*"New error as of Dec. 2019: Users are getting Python errors."*	10.6
	Documentation	Writing and improving documentation.	*"I will change the instructions to point to the better MI extension."*	5.3
	Version Tracking	Reporting and managing changes to the design.	*"I posted an updated top clamshell. Make sure you have the latest."*	1.3
	Community Support	Participating in discussions, providing troubleshooting support, and helping others.	*"How can I help you? How are the parts breaking, or do you mean the whole machine does not work?"*	66.0

4.1.1 Overview of the DrawBot Project. Over the course of the DrawBot project, collaborations between designers resulted in numerous contributions that evolved the DrawBot design. Through qualitative coding, we were able to identify which of the 402 commenters actually made design changes to the DrawBot. Furthermore, we are able to obtain a detailed understanding of the topic, duration, and outcome of the collaboration.

   The network graph in Figure 5 summarizes the collaboration activity among all 48 makers who made design-related contributions to the Drawbot; we call them "contributors". This network shows which contributors collaborated, the extent of the collaboration, the specific components of the DrawBot they discussed, and the contribution made (whether hardware, electronics, software or a combination). In particular, there are four highlighted clusters in the network, which each represent an instance of collaboration that we will describe in Section 4.1.2.

4.1.2 Design-Related Contributions. Design-related contributions pertain to the actual DrawBot device. The examples presented in this section describe how makers collaborated to modify or
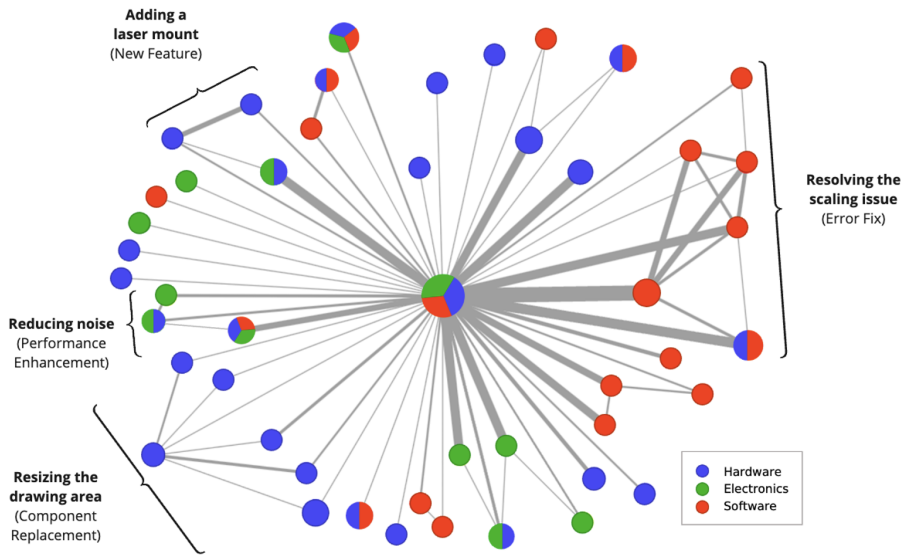
Fig. 5. Contributors to the DrawBot project. Each node represents a maker who made a design-related contribution, with the size of the node representing the number of contributions made (note: for the sake of figure legibility, we scaled down the central node since this was the DrawBot author, who made numerous contributions). The edges represent conversations (i.e., threads) shared between makers, with the edge weights showing the frequency of collaboration (i.e., the number of conversations). Blue, green, and red nodes represent exclusively hardware, electronics, and software contributions, respectively. Multi-coloured nodes represent interdisciplinary contributors.

improve one or more of the components of the DrawBot, which may include changes to the CAD models, code files, configuration of the electronics, or assembly. To give a better idea of when and how often these contributions were made to the project, Figure 6 shows the number of design-related contributions per year of the project's active development.

New Feature. The most common design-related contributions were *New Features*, encompassing additional functions, parts, or enhanced features of existing parts within the DrawBot. As shown in Figure 6a, the number of hardware and software contributions peaked in 2020, and there were few new electronics features overall. Of the relevant conversations, 10.8% discussed adding a new feature to the design.

Many makers were interested in adding parts to the pen holder component to hold items beyond pens, such as pencils, vinyl cutters, and paintbrushes. In particular, three makers collaborated to add a laser mount, enabling wood engraving. The author initially proposed this idea in June 2017, stating, *"I intend to experiment with a laser mount."* At first, uptake on this task was slow, until newcomers to the project were also interested in the laser functionality. The author responded to each new collaborator with the information they had, such as, *"Check out this extension, which provides guidance on connecting the laser controls to the Arduino board. I haven't tested it but it looks promising."* As the collaborators made progress in designing, prototyping, and testing the new design, they reported updates in the comments, like, *"I needed to replace the rods to support the weight of the laser module".*

Eventually, in December 2019, one of the collaborators responded with a final solution, stating, *"I used the Neje 3500 laser, which is compatible with the existing firmware. My first attempt at a custom*

469:14

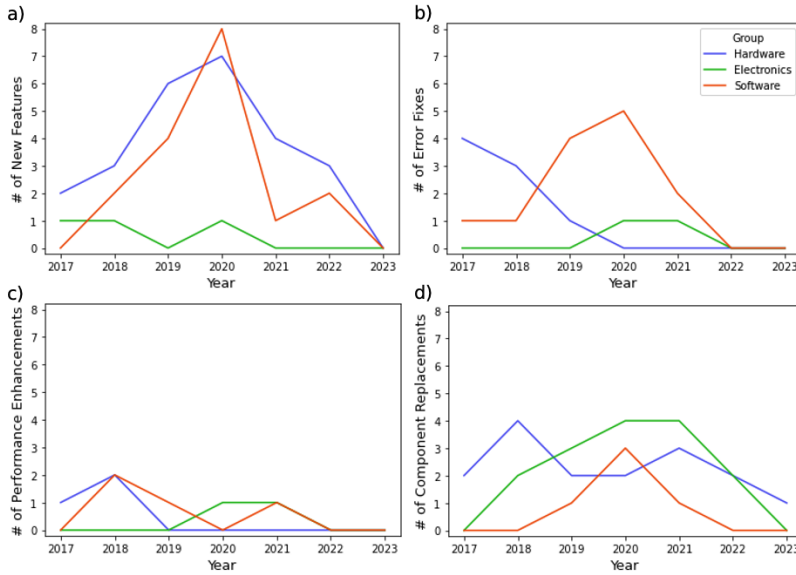Kathy Cheng, Shurui Zhou, and Alison Olechowski

Fig. 6. Number of design-related contributions made to the DrawBot's hardware, electronics, and software components over time, based on contribution type: (a) new features; (b) error fixes; (c) performance enhancements; and (d) component replacements.

mount is working well so far." To which the author responded, *"Great info! I'm planning to do this too. Please share your STL files as a remix."* Unfortunately, the profile of the maker who contributed to the new laser mount has since been deleted, so we were not able to confirm if the files were posted. Nonetheless, they contributed new knowledge as to how to incorporate a laser into the DrawBot.

Error Fix. Error fix contributions were necessary when there was a problem with a component that either rendered the DrawBot unusable or negatively impacted the DrawBot's manufacture or functionality. Error fixes were mentioned in 4.3% of conversations.

The first design-related contribution to the DrawBot, aside from the initial post by the author, was a hardware error fix. This fix involved redesigning the clamshell parts after another maker faced difficulties with printing and assembling, mentioning, *"I could force fit the bearings into the clamshell, but the two clamshell halves are still misaligned, causing severe bending of the rods"*. After the author uploaded an updated clamshell model, a new error was introduced, this time in the 3D printing of the part, with another maker reporting, *"still struggling with the clamshell print, it keeps lifting off the base. Could you check if the clamshell is properly flush with the print base?"*; to which the author replied, *I apologize, you were right! The top clamshell had a slight 0.2-degree angle, which I overlooked. Please see Revision 3 of the file."*

As shown in Figure 6b, most hardware error fixes occurred early in the DrawBot Thing's publication, gradually decreasing until 2020, when no further error fixes occurred. This trend could be the case because hardware errors prevent the maker from being able to successfully print and assemble the device – which is a required prerequisite to wiring electronics and programming software – so, problems with the 3D models were quickly identified and resolved. Furthermore, since a maker would typically only make a DrawBot once, and physical parts remain unchanged (unless they require replacement or upgrades), a more mature project sees fewer hardware errors to fix, as DrawBots are already up and running smoothly.

Although hardware error fixes decreased over time, software errors exhibited the opposite trend, steadily increasing until 2020. In January 2020, the DrawBot community faced a significant software issue with a compatibility bug between Inkscape and the MI Extension, resulting in scaling problems with drawings. One maker identified the error, *"Hello everyone, has anyone encountered the same problem I'm facing with Inkscape's scaling of G-code output?"*, followed by many other makers reporting the same problem.

Initially, the author proposed a fix suggesting adjustments to the MI extension's Python code for G-code creation, stating *"Try this workaround: adjust the drawing scale to 3.5433"*, which worked for some types of illustrations (e.g., text writing) but not all. After two months of employing this workaround, a maker decided that it was time for a more permanent and holistic solution, stating, *"I wanted to address the scaling issue that to this point has remained unresolved. I've been looking into this, and here's what I found"*, accompanied by an extensive list of links to online resources.

This launched a trial-and-error process involving six makers who posted their ideas, shared online resources, and tested each other's solutions on their DrawBots. After several discussions until August 2020, a fix was finally found for the original MI Extension software, but the six makers simultaneously became interested in replacing the extension with a newer, more reliable version. This group of six makers again dedicated themselves to resource sharing and testing, leading to a collaborator presenting a fix in a remix in November 2020. As described by the collaborator, *"I created an updated Inkscape Extension that fixes problems with the original, and is also compatible with the latest version of Inkscape."*

Performance Enhancement. Over the course of the DrawBot project, performance enhancement contributions were made the least, mentioned in 2.8% of conversations. One priority was to increase DrawBot's drawing speed. One collaborator experimented with modifying hardware components: *"I've replaced the stainless steel rod with 8 mm aluminium tubes to reduce mass, enabling the drawing robot to move faster."* Once they posted a successful remix of this design change, this inspired another maker to follow suit, responding to the collaborator who posted the remix, *"I found the rods in a nearby shop! They're cheaper and easier to cut."*

In the context of this 3D printing community, makers may find it relatively straightforward to enhance the performance of hardware components (e.g., designing lighter parts). However, when it came to electronic components, which are typically standardized and purchased off-the-shelf, makers rarely suggested performance enhancements. In a scarce number of instances where collaborators did discuss improving the performance of electronic components (e.g., CNC Shield), it was most often to suggest a replacement, rather than to offer new approaches to wire or solder the part. Over the month of February 2021, four makers collaborated in selecting replacements by providing links to various stepper drivers and discussing the advantages and disadvantages of each. After identifying the optimal one and successfully testing it, one of the collaborators suggested, *"for a significant noise reduction, switch from A4988 to TMC2209 stepper drivers. Be aware that the TMC2209 drivers come with a larger heatsink that is not compatible with the provided housing in the Thing Files."* Not only did they offer a replacement that would improve the noise reduction, but they also detailed necessary hardware adjustments to accommodate the new drivers, highlighting a holistic approach to performance enhancement.

As shown in Figure 6c, performance enhancements were scarce, which aligns with existing literature that suggests OSH communities often emphasize creating alternative designs rather than continuously refining a single central design [63]. This trend may be attributed to a lack of engineering expertise in design optimization or a lack of incentive to pursue the ideal design, given the maker-oriented nature of Thingiverse.

Component Replacement. Although design-related contributions are typically associated with the DrawBot's enhanced performance, or the introduction of a new feature, we found that some contributions do not necessarily improve the design. Rather, 6.2% of conversations discussed replacing a component in the design. Design changes as a result of *Component Replacement* happened most often because the maker already had a particular component and wanted to make use of it. For example, one commenter asked, *"Can the CNC shield v4 be operated with an Arduino NANO? I have a spare one."* Another common reason was the unavailability of specific materials in the maker's geographic area. For example, one maker expressed, *"I can't find Nema 17 0.4 amp motors in my country. Would it be fine to use a different Nema 17 motor and modify the driver current?"*

Beyond the lack of component availability, some makers also cited material tradeoffs as a reason for *Component Replacement*. Even though the open-source nature of the project implies technical freedom in design, the reality of creating a physical product requires the end user to procure the hardware materials. Thus, there exists an inherent tradeoff in the selection of materials, often influenced by considerations such as component costs. For instance, in June 2018, one collaborator posted a remix, saying, *"Here's a budget-friendly version. I cut costs by swapping the rods with hollow, shorter ones and the metal bearings with 3D printed ones."*

After this collaborator successfully downsized their DrawBot's drawing area by replacing the rods, other makers followed suit, referring to the original maker who posted the change. In total, five separate makers consulted with this maker to achieve the component replacement task. This cluster of makers is shown in Figure 5. Since the discussion on rod replacement involved fewer iterations and less problem-solving, the five nodes are not interconnected but rather directly connected to the first maker who successfully achieved the dimensions.

In each of these examples, the primary motivation behind the design change was working around practical obstacles. Obstacles related to component availability, cost, or ease of assembly are key considerations in the design of physical products. Figure 6d shows that the occurrence of component replacements remained relatively constant over the project's duration. This could be attributed to the nature of component replacements being driven by individual makers' needs (e.g., the materials available to them), and this lack of access is not affected by the evolution of the design.

4.1.3 Non-Design-Related Contributions. Although non-design-related contributions do not involve changing the actual DrawBot, they are essential for driving the development work. These non-design-related contributions are essentially "articulation work," or *"work to make the cooperative work work"* [86]. In this section, we highlight how non-design-related contributions facilitated streamlined collaboration.

Issue Tracking & Management. Throughout the project, makers used comments to raise and discuss issues in the comments, encompassing both problems and tasks for the community to address. 10.6% of the conversations were about tracking and managing issues.

The issue tracking process typically followed three stages: (1) reporting (e.g., *"FYI, the GRBL parameters displayed in the images and in the PDF instructions are different"*); (2) acknowledging or undertaking the task (e.g., *"Yes, I'm aware of some discrepancies in the documentation. I plan to fix these in the next revision"*); and finally, (3) resolving or updating (e.g., *"I corrected links in the documentation and made minor edits."*).

Issues were not always in response to problems; sometimes, an issue signified a task or specific area in which the community should focus their development efforts. For instance, when aiming to expand the software capabilities for different types of illustrations, the author issued an open call for future directions, stating, *"If you're open to enhancing the design, consider upgrading the software to a newer version of GRBL for added features. Additionally, explore a controller that integrates Arduino*

and CNC circuitry, which can also support a laser. Understandably, if it's too much, the current design works well, but improvements could be exciting."

Documentation. Contributions relating to documentation involved the creation of new documentation or the improvement of existing documentation, which included user manuals, images, and assembly instructions. Documentation contributions represented 5.3% of conversations.

In the initial publishing of the DrawBot, the author provided a parts list, CAD files necessary to print the DrawBot, and links to software downloads with screenshots of correct parameters. However, there was no user guide with full manufacturing and assembly instructions, until one enthusiastic maker created and shared an 84-page guide on Google Slides: *"Here's the link to an instruction manual I made."* Appreciative for the effort, the author responded, *"Wow, impressive manual! Thank you! Can I share it in the Thing files? Otherwise, it might be overlooked in the comments. Feel free to include it in a remix of this drawing robot."* As changes were made to the design, both the author and the creator of the documentation updated this user guide on Google Slides, and exported it as a PDF to include in the Thing files.

Version Tracking. In the absence of sophisticated version control tools, the DrawBot community used the comments section to facilitate the tracking of design versions, notifying others about the availability of new design files. Only 1.3% of conversations contributed to version tracking.

One maker, aiming to enhance the pen holder design, created a remix and shared the updated design in the comments, stating, *"Here is a link to a significantly improved pen holder. I'm creating one for my DrawBot."* When the new holder was successfully printed and integrated into the DrawBot assembly, the original author then "merged" the new version by adding the CAD files from the remix into the original DrawBot repository.

Not all design changes made in remixes were merged; only those considered significant by the author were included on the DrawBot page. Factors such as the severity and prevalence of the problem addressed by the new design influenced the decision. For instance, an improved version of a widely-used component, like the pen holder, was more likely to be merged, while the addition of a wall mount was somewhat of a diversion from the main design. These designs could still be found in remixes, but were not added to the Thing files.

Community Support. Community support involved answering questions and providing troubleshooting support. Notably, 66.0% of the comments in the DrawBot community were related to community support. This contribution involved answering various questions regarding: (1) 3D printing (e.g., *"can I print these using PLA instead of ABS?"*); (2) the cost and procurement of materials (e.g., *"filament aside, how much are the materials?"*); and (3) assembly (e.g., *"Can someone show the wiring of the servo to the CNC Shield?"*).

Providing troubleshooting help was another popular contribution. Makers seeking troubleshooting help often shared images and videos for hardware or electronics issues and provided their code and error messages for software-related problems. The DrawBot author promptly responded to all troubleshooting requests, offering supportive messages, such as, *"Simply shift the servo up by one spline, and you should be fine. Don't be too hard on yourself. This project has a lot of moving parts, quite literally!"*

As the project progressed, and more makers successfully printed and operated the DrawBot, they actively engaged in troubleshooting tasks to assist the original author. In one example, a maker was having issues fitting the belt into the clamshell, which prompted a contributor (who was not the author) to offer methods to tighten the belt. In acknowledgement, the author replied, *"First check out the valuable suggestions provided by [Maker X] in the comment below. If that doesn't work, I have a few other ideas."*

4.2 Collaboration Challenges & Workarounds (RQ2)

Our objective with this study was to explore how collaboration unfolds in a successful OSH project. Although the DrawBot project was a success, there were challenges that impeded the collaboration's full potential. In this section, we present three collaboration challenges articulated by makers in the comment threads, alongside a discussion of how the DrawBot community overcame these challenges.

4.2.1 Poor Version Control Support. On Thingiverse, the Thing page is essentially a snapshot of the repository, showing only the most recent version, rather than an overview of the entire design process [60]. Since there is no version control support, some makers unintentionally used outdated components. In the clamshell scenario described in Section 4.1.2, despite a new CAD file being uploaded, four different makers were unaware of the update and downloaded and printed the outdated version, leading to issues highlighted in the comments, like *"Is there an issue with the top clamshell? I've had difficulties printing it and problems with adherence."* These situations not only frustrate makers but also result in wasted time and print material.

To address the lack of version control support and work around this challenge, DrawBot contributors had two strategies: remixing, and tracking changes in the description.

Remixing. Remixing allows makers to copy and upload new versions of Things. In order to post a remix, it must contain a CAD file, which was straightforward for the majority of the DrawBot remixes which contained upgraded or new 3D models. However, this can be inconvenient for revisions to a software component. For instance, one maker replaced the GRBL firmware with a newer, more versatile system, and was asked by the author, *"Please share a remix of your solution. You just need to upload one STL file along with information directing to your GitHub link."* Employing this workaround, the maker copied a random CAD file from the DrawBot into their remix, but this is not an optimal solution. Future development is needed to support remixing of electro-mechanical designs in OSH.

Updating the Thing Description. On a Thing page, the description section is intended to summarize a Thing's function and instructions for 3D printing and assembly. In addition to providing this information, we found that DrawBot's author used the description to track design changes and document version history. After each new revision, the author added a short summarization of the change with the revision date, similar to *commit messages* in collaborative software development [97]. One such example is, *"Update Dec 1, 2020: I've uploaded upgraded parts for the pen slider, pen holder, cable support anchor point, and Arduino enclosure mount, all of which are backward compatible with the originals. These enhancements result in a neater and sturdier DrawBot."* While this process is manual and tedious, it currently serves as the only means to track and display revision updates on the Thing page.

4.2.2 Scattered Collaboration. Collaboration on Thingiverse can be scattered, which makes it difficult for contributors to maintain an awareness of the people, design artifacts, and documentation of the project.

Remixing is considered a way to maintain links between designs, but some Thingiverse makers suggest that remixing can actually harm collaboration through community fragmentation. When one maker requested an updated design for the motor mounts, they decided against creating a remix, and instead asked the author, *"Could you please fix these clamshell holes? It would be a shame to do a remix and split the community".*

Maintaining an awareness of the contributions made within remixes is also challenging, especially when there are many remixes that span multiple pages that the user has to click through. Thus, it is

common that even when someone takes the proper steps to create a remix, it can be overlooked by other makers. In the example where makers wanted to resize the drawing area, one maker pointed another to an existing solution, stating, *"Other users have successfully enlarged the design and shared their builds. Check out their remix"*, accompanied by a link to the remix.

Documentation can also be scattered, since many makers link to external image- or video-sharing platforms, like Imgur, YouTube, or Google Drive. For example, makers created YouTube videos to demonstrate assembly instructions. As stated by one maker, *"I successfully configured and optimized my DrawBot. Check out the results in this video. Feel free to leave comments here or on YouTube if you have any questions."* These uploads can attract makers not originally from Thingiverse to become involved in the project. However, the lack of integration between these platforms and Thingiverse further perpetuates the fragmentation of the community.

4.2.3 Lost Contributions. Although makers were encouraged to make remixes to share their design contributions, not all were captured in this method; many were provided in the comments instead. Currently, individuals other than the original author cannot directly modify the Thing, its metadata, or related files. Consequently, valuable fixes provided in the comments may be hidden amidst numerous conversations. To avoid this challenge, the author suggested, *"Remix the DrawBot with your code as a separate file to prevent it from getting buried in the comments. With your permission, I may eventually incorporate your fix into the documentation, but for now, a remix would be greatly appreciated. I'll link to your remix from my Thing description."* This example highlights two approaches to prevent lost contributions: creating a remix, and updating the Thing description (Section 4.2.2).

However, due to the subjective nature of what makers consider as derivatives and the voluntary nature of attributing credit in remixes, makers were not always disciplined in linking remix relationships. In the comments, one maker might direct another to a related Thing page, but that Thing is in no way linked to the original DrawBot page. As a result, makers would be unaware of these contributions unless they read the comments in detail.

While these challenges result from a lack of awareness of derivatives (i.e., downstream Things), contributions in sources (i.e., upstream Things) can similarly be lost. When makers were not properly given credit where credit was due, tensions arose in the community, as illustrated by a comment in one of the DrawBot network Things: *"I created the original design. They simply 'borrowed' it, modified the base, and failed to mention that they aren't the author of this plotter"*.

Crediting Contributors. To avoid tensions, crediting fellow makers for their contributions is a simple, yet effective approach. The DrawBot author diligently followed this practice, and acknowledged collaborators in both the comments and Thing description, with quotes like *"[Maker Y] supplied assembly documentation. Thank you very much. I value your efforts."*, or *"New error as of Dec. 2019. Appreciation to [Maker Z] for documenting this issue."* Again, it is important to note that these attributions are done manually, relying entirely on the Thing author to uphold this best practice.

5 Discussion

This section discusses the main implications of our work, which are targeted towards the participants, platform designers, and researchers of OSH projects.

5.1 Best Practices for Future Open Hardware Projects

Among existing accounts of OSH, DrawBot stands out as a successful and sophisticated example, in terms of fostering a long-lived community and continuously-evolving design. As such, we believe there are several lessons to be learned from this case that can be applied to future OSH projects to

improve collaborative design. In the following paragraphs, we discuss three unique facets of the DrawBot project that, in our view, played a crucial role in its success.

The first facet is the use of high-quality documentation, which has been highlighted as key to successful OSH projects [4]. Despite the lack of formal documentation support features, the DrawBot author managed to create this documentation through iterative additions to the Thing description. Notably, the DrawBot's description was an order of magnitude longer than that for other Things, reflecting the depth and richness of available documentation. Given DrawBot's complexity, which involves designing, troubleshooting, and integrating both physical and digital artifacts, the comprehensive documentation outlining assembly instructions and addressing known challenges was exceptionally valuable. Strong documentation is key to retaining newcomers to OSS communities [36, 102], and this may have contributed to the sustained engagement of newcomers in this project, contrasting with previous studies which found that makers tend not to continuously engage with a specific Thing [59].

The maintainer is significant in any open-source context, responsible for overseeing project development, reviewing contributions, and ensuring the overall sustainability of the project [44]. The significance of the maintainer is particularly evident in the case of the DrawBot. Since there are no existing best practices for collaborative design on collective design platforms like Thingiverse, the maintainer – or author in this case – must assume a leadership role to establish norms, suggest tasks, and provide general guidance to the community. The DrawBot author conversed with every single DrawBot contributor (Figure 5), and this responsiveness likely fostered confidence in others about the quality and sustainability of the project. Importantly, responding to each newcomer is a laborious task [38], and future work should explore workload management.

Alongside responsiveness, another factor potentially contributing to DrawBot's success was the author's interdisciplinary expertise spanning hardware, electronics, and software components. While this multidisciplinary expertise by the author was evident in their design contributions to the DrawBot, our study lacks the necessary data to confirm if a multidisciplinary project maintainer is the key to successful OSH collaboration. A promising area for future exploration is the impact of multidisciplinarity on OSH collaboration, which we discuss in Section 5.3.

Although the DrawBot project succeeded with clusters of collaborators contributing to the same type of component, prior work suggests an alternative best practice. Salehi et al., in a study of software development, argue that rotating individuals within clusters of design tasks can be beneficial to collaboration, inviting more diverse perspectives and cross-pollination of ideas [85]. This difference in approach may stem from inherent distinctions between hardware and software projects, such as the nature of the modularity, which is closely tied to product architecture [37].

Overall, DrawBot's success can be attributed to high-quality documentation, responsiveness, and interdisciplinarity. Future OSH maintainers could consider these best practices a valuable starting point when initiating similar collaborative design projects.

5.2 Design Implications for OSH Platforms

One of our study's contributions is identifying challenges that hindered the collaborative design process. Here, we see that even in successful OSH projects, tooling limitations hold project potential back. In an effort to minimize tooling-related challenges for future OSH projects, we discuss five design implications for the platform builders of OSH communities, aiming to improve awareness and traceability.

The first design implication is to make documentation more accessible and traceable. As presented in Section 4.2.2, contributors often struggle to maintain awareness of essential documentation scattered across various links, some of which lead to external sites such as GitHub and YouTube. While the maintainer tried to centralize documentation links in the Thing description, it is still

cumbersome to explore content from disparate links. Similar challenges occur in OSS community sites, like Stack Overflow, and researchers have made recent contributions to increase the visibility of repeatedly-shared links [58], and automatically align documentation with relevant questions in discussion posts [83]. OSH platforms may also benefit from more automated approaches for navigating documentation or summarizing documentation content. Although solutions from the software literature may assist in summarization methods for text-based content (e.g., assembly instructions), more advanced techniques are required to analyze images and CAD models [92].

Next, platform builders should address the poor navigation and traceability of the remixing graph, which makes it challenging for makers to find the necessary information. Nickerson relates this challenge to the *theory of information foraging* [69], which aims to understand the way humans search, gather and consume information, often online [81]. Generally, the longer a user spends searching for a new idea, the less likely that it will be found [81]. Consequently, if new designs are difficult to find in remixing graphs, there is a risk that contributions can go unnoticed (Section 4.2.3). Thus, it is imperative to improve the design of the remixing graph to provide users with a better overview of remixes and sources. Drawing on the types of design-related contributions we identified, there could be value in organizing remixes by: (1) component, e.g., grouping all parts related to the pen holder; (2) product family, e.g., grouping all the files that pertain to a downsized DrawBot; and (3) version, e.g., grouping parts belonging to the same revision.

Along the same lines, contributions can be buried in lengthy comment threads. A fruitful area for future work could involve developing methods to summarize the knowledge exchanges and outcomes of such conversations, inspired by OSS research [43]. Such a summarization tool could automatically identify feature requests for the next design iteration. Furthermore, it would streamline the project onboarding process by automatically directing newcomers to relevant conversation threads, alleviating the burden on the maintainer to respond to each question individually.

The next design implication motivates the development of an issue tracker to identify/flag and react to bugs/errors. In the DrawBot project, all of the tasks for tracking issues were done manually, which is tedious and error-prone. Designing a system similar to the GitHub issue tracker would streamline reporting, updating, and resolving issues [10].

Finally, OSH platforms need sophisticated version control and pull request functionalities, allowing makers to contribute directly to the Thing without resorting to workarounds, such as remixing and copying files from the remix into the original Thing as a pseudo-merge. There are many shortcomings with this method: the manual merge involves many steps; only the original author can edit the Thing; and there is no version history, further impairing contributors' awareness of the design evolution. Branching and merging for CAD files is indeed emerging as a promising tool for CAD collaboration, but there are still essential features lacking, such as selective merging [18].

Our work contributes to understanding awareness and traceability needs in OSH projects, providing specific areas for future development. Implementing these design insights will not only aid in overcoming awareness challenges within OSH projects but may also be relevant for broader open collaboration contexts, such as software development [24] or collaborative writing [39, 52].

5.3 Practical Implications for OSH Researchers

To our knowledge, this study is the first to analyze a network of connected designs on a collective design platform like Thingiverse, aiming to understand the challenges and opportunities of a collaborative design project. In this section, we aim to provide practical guidance for OSH researchers who may wish to conduct similar empirical investigations of collaboration processes. First, we present some heuristics that allowed us to recognize core contributors, followed by a discussion on navigating multidisciplinarity in OSH projects.

5.3.1 Identifying Collaborators. By qualitatively coding comment threads, we were able to gain deeper insight into the nature and context of design changes, which are not captured in remix activity alone. One notable finding is that the authors of remixes are not representative of all contributors. Since qualitative analysis can be time-consuming, making it impractical to read every comment for contributor identification, we have compiled a short list of heuristics that we believe can be valuable in future studies to more efficiently identify key contributors. Generally, we found that contributors were:

- Makers who made the most comments.
- Makers who comment on two or more related Things, indicating that they are following the evolution of the design in remixes.
- Makers whose usernames were mentioned in the Thing description, often signifying acknowledgement of their contributions by the author.
- Makers who include a Thingiverse link in their comment. This usually indicated that they referred to a related design, which may not have been captured as a remix.

Following and expanding on these heuristics can help researchers identify and analyze contributors to OSH projects more efficiently.

5.3.2 Navigating Multidisciplinarity. A product like DrawBot requires collaboration between makers with hardware expertise and makers with software expertise, but the platforms supporting them are distinct and disconnected. In the DrawBot project, we observed version tracking and evolution of the hardware components occurring on the Thing page, while software contributions were occasionally made on GitHub. It is possible that the development processes can diverge on different platforms. Therefore, OSH researchers conducting similar investigations should consider tracking design and collaboration activity on concurrent platforms, which might offer interesting insights into how multidisciplinarity affects OSH collaboration.

Below are some open questions regarding multidisciplinarity to tackle in future work:

- Are the designers collaborating on hardware platforms (e.g., Thingiverse) also involved in collaborations on software platforms (e.g., GitHub)?
- Do the platforms separate software-focused and hardware-focused designers?
- How might collaboration practices differ, considering that GitHub is a more sophisticated version control system than Thingiverse, which is a collective design platform?

With these practical implications, our goal is to assist OSH researchers interested in conducting case studies in a similar fashion. Building on our work with additional case studies will contribute to a more nuanced understanding of OSH collaboration to the CSCW community.

5.4 Limitations & Future Work

As with any case study, our goal is not to be generalizable to all OSH projects. Instead, we aimed to uncover insights potentially relevant to other projects through an in-depth examination of collaborative practices within an advanced OSH design project. Therefore, as was our intention, this project's characteristics are not representative of all the designs in collective design platforms.

Next, while our project selection allows us to achieve our research objective, it only provides insights into one online community. Future work should compare our findings with other collaborative design projects, to gain a more comprehensive understanding of OSH collaboration.

Another limitation pertains to how we define a project and what constitutes a successful one. Given the novelty of our approach, we based our inclusion criteria (e.g., remixes and ancestors) and success metrics (e.g., number of likes, comments, remixes) on insights derived from related

literature [3, 4, 6, 72, 73, 78, 101, 107]. It is in the interest of future work to validate our methods and formalize project-level boundaries on platforms like Thingiverse.

Furthermore, our study is limited by the type of activity trace data [88] available through Thingiverse's API. As a result, we cannot analyze design processes that occurred outside of the Thingiverse platform, such as in-person assembly activities. In future work, we plan to conduct qualitative interviews with the collaborators of an OSH project, to gain these additional insights.

Finally, qualitative analysis can be vulnerable to the researcher's bias. To address this concern, we analyzed multiple data sources, including comments, descriptions, and actual design changes in CAD models, to triangulate our findings, which is recommended for case study research [105].

6 Conclusion

In this work, we conducted a detailed case study of collaborative design practices in the DrawBot project on Thingiverse. Using our mixed-methods approach, we analyzed discussions in comment threads, design contributions in remixing relationships, and changes made to various documentation, to understand how collaboration occurred in a successful and complex OSH project.

Our research revealed that designers collaborated on various tasks, including designing new features, addressing errors, enhancing performance, and replacing components of the DrawBot. Collaborative efforts ranged from short-term discussions involving a few makers to address noise reduction to more extensive collaborations lasting 11 months, focusing on resolving significant software errors. Practical considerations – those related to designing a physical product, such as material availability, costs, and the use of standardized electronic components – played a crucial role in shaping these collaborations. In addition, not all collaborations resulted in a change to the DrawBot design; many discussions centred around process-related contributions, such as documentation improvements and troubleshooting assistance. Lastly, our study highlighted unique approaches exercised by the DrawBot community to overcome limitations within the Thingiverse platform, such as tracking version history.

This case study sheds light on the resilience and resourcefulness of the DrawBot community, demonstrating that a collaborative OSH design project can indeed thrive, despite inadequate support from existing platforms. However, in order to advance the scope of OSH, our study also identifies various areas for improvement. We contribute best practices for OSH project maintainers to enhance collaboration efficiency, design implications for platform builders to address current challenges, and practical implications for guiding researchers in expanding OSH research within CSCW. With each step taken towards improving design collaboration, we push the boundaries of what OSH innovation can achieve.

Acknowledgments

References

[1] Kaveh Abhari, Elizabeth J. Davidson, and Bo Xiao. 2017. Co-innovation platform affordances: Developing a conceptual model and measurement instrument. *Industrial Management & Data Systems* 117, 5 (2017), 873–895. https://doi.org/10.1108/IMDS-05-2016-0156 Num Pages: 23 Place: Wembley, United Kingdom Publisher: Emerald Group Publishing Limited.

[2] Mathieu Acher, Benoit Baudry, Olivier Barais, and Jean-Marc Jézéquel. 2014. Customization and 3D printing: a challenging playground for software product lines. In *Proceedings of the 18th International Software Product Line Conference - Volume 1*. ACM, Florence Italy, 142–146. https://doi.org/10.1145/2648511.2648526

[3] Celena Alcock, Nathaniel Hudson, and Parmit K. Chilana. 2016. Barriers to Using, Customizing, and Printing 3D Designs on Thingiverse. In *Proceedings of the 19th International Conference on Supporting Group Work (GROUP '16)*. Association for Computing Machinery, New York, NY, USA, 195–199. https://doi.org/10.1145/2957276.2957301

[4] Rafaella Antoniou, Jérémy Bonvoisin, Pen-Yuan Hsing, Elies Dekoninck, and Daniela Defazio. 2022. Defining success in open source hardware development projects: a survey of practitioners. *Design Science* 8 (2022), e8. https://doi.org/10.1017/dsj.2021.30

[5] Rafaella Antoniou, Romain Pinquié, Jean-François Boujut, Amer Ezoji, and Elies Dekoninck. 2021. IDENTIFYING THE FACTORS AFFECTING THE REPLICABILITY OF OPEN SOURCE HARDWARE DESIGNS. *Proceedings of the Design Society* 1 (Aug. 2021), 1817–1826. https://doi.org/10.1017/pds.2021.443 Publisher: Cambridge University Press.

[6] Seungyeop Baik, Sunyoung Choi, Suho Ahn, and Hangjung Zo. 2022. The role of remixing for innovation in online innovation communities. In *ICEB 2022 Proceedings*, Vol. 22. AIS, Bangkok, Thailand, 544–551. https://aisel.aisnet.org/iceb2022/50

[7] Yekta Bakırlıoğlu and Cindy Kohtala. 2019. Framing Open Design through Theoretical Concepts and Practical Applications: A Systematic Literature Review. *Human–Computer Interaction* 34, 5-6 (Sept. 2019), 389–432. https://doi.org/10.1080/07370024.2019.1574225 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/07370024.2019.1574225.

[8] 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 - CSCW '96*. ACM Press, Cambridge, MA, USA, 209–218. https://doi.org/10.1145/240080.240256

[9] Alexander Berman and Francis Quek. 2020. ThingiPano: A Large-Scale Dataset of 3D Printing Metadata, Images, and Panoramic Renderings for Exploring Design Reuse. In *2020 IEEE Sixth International Conference on Multimedia Big Data (BigMM)*. IEEE, New Delhi, India, 18–27. https://doi.org/10.1109/BigMM50055.2020.00014

[10] Tegawende F. Bissyande, David Lo, Lingxiao Jiang, Laurent Reveillere, Jacques Klein, and Yves Le Traon. 2013. Got issues? Who cares about it? A large scale investigation of issue trackers from GitHub. In *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, Pasadena, CA, USA, 188–197. https://doi.org/10.1109/issre.2013.6698918

[11] Paulo Blikstein. 2013. *Digital Fabrication and 'Making' in Education: The Democratization of Invention*. transcript Verlag, Bielefeld, Germany, 203–222. https://doi.org/10.1515/transcript.9783839423820.203

[12] Andrea Bonaccorsi and Cristina Rossi. 2003. Why Open Source software can succeed. *Research Policy* 32, 7 (July 2003), 1243–1258. https://doi.org/10.1016/s0048-7333(03)00051-9

[13] Jérémy Bonvoisin, Tom Buchert, Maurice Preidel, and Rainer G. Stark. 2019. How participative is open source hardware? Insights from online repository mining. *Design Science* 4 (Nov. 2019), e19. https://doi.org/10.1017/dsj.2018.15 Publisher: Cambridge University Press.

[14] Jérémy Bonvoisin, Robert Mies, Jean-François Boujut, and Rainer Stark. 2017. What is the "Source" of Open Source Hardware? *Journal of Open Hardware* 1, 1 (Sept. 2017), 5. https://doi.org/10.5334/joh.7

[15] Jérémy Bonvoisin, Laetitia Thomas, Robert Mies, Céline Gros, Rainer Stark, Karine Samuel, Roland Jochem, and Jean-François Boujut. 2017. Current state of practices in open source product development. In *DS 87-2 Proceedings of the 21st International Conference on Engineering Design (ICED 17)*, Vol. Vol 2: Design Processes, Design Organisation and Management. The Design Society, Vancouver, Canada, 111–120. https://www.designsociety.org/publication/39565/Current+state+of+practices+in+open+source+product+development ISBN: 9781904670902.

[16] Hudson Borges, Andre Hora, and Marco Tulio Valente. 2016. Understanding the Factors That Impact the Popularity of GitHub Repositories. In *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, Raleigh, NC, USA, 334–344. https://doi.org/10.1109/icsme.2016.31

[17] Erin Buehler, Stacy Branham, Abdullah Ali, Jeremy J. Chang, Megan Kelly Hofmann, Amy Hurst, and Shaun K. Kane. 2015. Sharing is Caring: Assistive Technology Designs on Thingiverse. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Seoul Republic of Korea, 525–534. https://doi.org/10.1145/2702123.2702525

[18] Kathy Cheng, Phil Cuvin, Alison Olechowski, and Shurui Zhou. 2023. User Perspectives on Branching in Computer-Aided Design. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (Oct. 2023), 371:1–371:30. https://doi.org/10.1145/3610220

[19] Kathy Cheng, Michal K. Davis, Xiyue Zhang, Shurui Zhou, and Alison Olechowski. 2023. In the Age of Collaboration, the Computer-Aided Design Ecosystem is Behind: An Interview Study of Distributed CAD Practice. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (April 2023), 137:1–137:29. https://doi.org/10.1145/3579613

[20] Kathy Cheng, Shurui Zhou, and Alison Olechowski. 2024. "A Lot of Moving Parts": A Case Study of Open-Source Hardware Design Collaboration in the Thingiverse Community. [dataset]. https://doi.org/10.17605/OSF.IO/7KDTY

[21] Juliet Corbin and Anselm Strauss. 2008. *Basics of Qualitative Research (3rd ed.): Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, Inc., Thousand Oaks, California. https://doi.org/10.4135/9781452230153

[22] John W Creswell. 2014. *Research design: Qualitative, quantitative, and mixed methods approaches.* SAGE, New Delhi, India.

[23] Irena Pletikosa Cvijikj and Florian Michahelles. 2011. *The Toolkit Approach for End-user Participation in the Internet of Things.* Springer Berlin Heidelberg, Berlin, Heidelberg, 65–96. https://doi.org/10.1007/978-3-642-19157-2_4

[24] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. 2012. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work - CSCW '12.* Association for Computing Machinery, Seattle, Washington, USA, 1277. https://doi.org/10.1145/2145204.2145396

[25] Jason Xinghang Dai, Jean-François Boujut, Franck Pourroy, and Philippe Marin. 2020. Issues and challenges of knowledge management in online open source hardware communities. *Design Science* 6 (Jan. 2020), e24. https://doi.org/10.1017/dsj.2020.18 Publisher: Cambridge University Press.

[26] X. Dai, J. F. Boujut, F. Pourroy, and P. Marin. 2020. Correlation between process openness and collaboration tool usage in open source hardware design: an empirical study. *International Journal on Interactive Design and Manufacturing (IJIDeM)* 14, 2 (June 2020), 481–489. https://doi.org/10.1007/s12008-019-00632-0

[27] Ruta Desai, James McCann, and Stelian Coros. 2018. Assembly-aware Design of Printable Electromechanical Devices. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology.* ACM, Berlin Germany, 457–472. https://doi.org/10.1145/3242587.3242655

[28] Paul Dourish and Victoria Bellotti. 1992. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work - CSCW '92.* Association for Computing Machinery, Toronto, Ontario, Canada, 107–114. https://doi.org/10.1145/143457.143468

[29] Emily Escamilla, Martin Klein, Talya Cooper, Vicky Rampin, Michele C. Weigle, and Michael L. Nelson. 2022. The Rise of GitHub in Scholarly Publications. In *Lecture Notes in Computer Science.* Springer International Publishing, Padua, Italy, 187–200. https://doi.org/10.1007/978-3-031-16802-4_15

[30] Amer Ezoji, Romain Pinquie, and Jean François Boujut. 2021. Towards a Better Understanding of Open Source Hardware Design Reuse in Company-Community Collaboration. In *Advances on Mechanics, Design Engineering and Manufacturing III (Lecture Notes in Mechanical Engineering),* Lionel Roucoules, Manuel Paredes, Benoit Eynard, Paz Morer Camo, and Caterina Rizzi (Eds.). Springer International Publishing, Cham, 149–154. https://doi.org/10.1007/978-3-030-70566-4_24

[31] Christoph M. Flath, Sascha Friesike, Marco Wirth, and Frédéric Thiesse. 2017. Copy, transform, combine: exploring the remix as a form of innovation. *Journal of Information Technology* 32, 4 (Dec. 2017), 306–325. https://doi.org/10.1057/s41265-017-0043-9

[32] Robbie Fordyce, Luke Heemsbergen, Thomas Apperley, Michael Arnold, Thomas Birtchnell, Michael Luo, and Bjorn Nansen. 2016. Things, tags, topics: Thingiverse's object-centred network. *Communication Research and Practice* 2, 1 (Jan. 2016), 63–78. https://doi.org/10.1080/22041451.2016.1155337

[33] Aline Shakti Franzke, Anja Bechmann, Michael Zimmer, and Charles M. Ess. 2020. *Internet Research: Ethical Guidelines 3.0.* Technical Report 2056305118763366. The Association of Internet Researchers. 83 pages. https://aoir.org/reports/ethics3.pdf

[34] Sascha Friesike, Christoph M. Flath, Marco Wirth, and Frédéric Thiesse. 2019. Creativity and productivity in product design for additive manufacturing: Mechanisms and platform outcomes of remixing. *Journal of Operations Management* 65, 8 (2019), 735–752. https://doi.org/10.1016/j.jom.2018.10.004 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1016/j.jom.2018.10.004.

[35] Sascha Friesike, Frédéric Thiesse, and George Kuk. 2021. What can the maker movement teach us about the digitization of creativity? *Commun. ACM* 64, 3 (March 2021), 42–45. https://doi.org/10.1145/3447524

[36] Felipe Fronchetti, David C. Shepherd, Igor Wiese, Christoph Treude, Marco Aurélio Gerosa, and Igor Steinmacher. 2023. Do CONTRIBUTING Files Provide Information about OSS Newcomers' Onboarding Barriers?. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2023).* Association for Computing Machinery, New York, NY, USA, 16–28. https://doi.org/10.1145/3611643.3616288

[37] Kosmas Gavras and Vasilis Kostakis. 2021. Mapping the types of modularity in open-source hardware. *Design Science* 7 (Jan. 2021), e13. https://doi.org/10.1017/dsj.2021.11 Publisher: Cambridge University Press.

[38] R. Stuart Geiger, Dorothy Howard, and Lilly Irani. 2021. The Labor of Maintaining and Scaling Free and Open-Source Software Projects. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (April 2021), 175:1–175:28. https://doi.org/10.1145/3449249

[39] R. Stuart Geiger and David Ribes. 2010. The work of sustaining order in wikipedia: the banning of a vandal. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work (CSCW '10).* Association for Computing Machinery, New York, NY, USA, 117–126. https://doi.org/10.1145/1718918.1718941

[40] Matt Germonprez, Georg J.P. Link, Kevin Lumbard, and Sean Goggins. 2018. Eight Observations and 24 Research Questions About Open Source Projects: Illuminating New Realities. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (Nov. 2018), 57:1–57:22. https://doi.org/10.1145/3274326

[41] Mohammad Gharehyazie, Daryl Posnett, Bogdan Vasilescu, and Vladimir Filkov. 2014. Developer initiation and social interactions in OSS: A case study of the Apache Software Foundation. *Empirical Software Engineering* 20, 5 (Aug. 2014), 1318–1353. https://doi.org/10.1007/s10664-014-9332-x

[42] Thomas Gillier, Sergio Toral Marín, and Rocio Martinez-Torres. 2020. Does Remixing Facilitate User Innovation? An Exploratory Study of an Online 3D Printing Platform. *Academy of Management Proceedings* 2020, 1 (Aug. 2020), 11721. https://doi.org/10.5465/AMBPP.2020.77 Publisher: Academy of Management.

[43] Saskia Gilmer, Avinash Bhat, Shuvam Shah, Kevin Cherry, Jinghui Cheng, and Jin L.C. Guo. 2023. SUMMIT: Scaffolding Open Source Software Issue Discussion Through Summarization. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (Oct. 2023), 297:1–297:27. https://doi.org/10.1145/3610088

[44] Georgios Gousios, Andy Zaidman, Margaret-Anne Storey, and Arie van Deursen. 2015. Work Practices and Challenges in Pull-Based Development: The Integrator's Perspective. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*. IEEE, Florence, Italy, 358–368. https://doi.org/10.1109/ICSE.2015.55

[45] André Hansen and Thomas J. Howard. 2013. The Current State of Open Source Hardware: The Need for an Open Source Development Platform. In *ICoRD'13 (Lecture Notes in Mechanical Engineering)*, Amaresh Chakrabarti and Raghu V. Prakash (Eds.). Springer, India, 977–988. https://doi.org/10.1007/978-81-322-1050-4_77

[46] J. Piet Hausberg and Sebastian Spaeth. 2020. Why makers make what they make: motivations to contribute to open source hardware development. *R&D Management* 50, 1 (2020), 75–95. https://doi.org/10.1111/radm.12348 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/radm.12348.

[47] Alfredo Herrera. 2020. The Promises and Challenges of Open Source Hardware. *Computer* 53, 10 (Oct. 2020), 101–104. https://doi.org/10.1109/MC.2020.3011080

[48] Megan Hofmann, Udaya Lakshmi, Kelly Mack, Rosa I. Arriaga, Scott E. Hudson, and Jennifer Mankoff. 2022. Making a Medical Maker's Playbook: An Ethnographic Study of Safety-Critical Collective Design by Makers in Response to COVID-19. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1 (March 2022), 1–26. https://doi.org/10.1145/3512948

[49] T. J. Howard, S. Achiche, A. Özkil, and T. C. McAloone. 2012. OPEN DESIGN AND CROWDSOURCING: MATURITY, METHODOLOGY AND BUSINESS MODELS. In *DS 70: Proceedings of DESIGN 2012, the 12th International Design Conference*, Vol. 1. The Design Society, Dubrovnik, Croatia, 181–190. https://www.designsociety.org/publication/31985/OPEN+DESIGN+AND+CROWDSOURCING%3A+MATURITY%2C+METHODOLOGY+AND+BUSINESS+MODELS

[50] Nathaniel Hudson, Celena Alcock, and Parmit K. Chilana. 2016. Understanding Newcomers to 3D Printing: Motivations, Workflows, and Barriers of Casual Makers. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 384–396. https://doi.org/10.1145/2858036.2858266

[51] Mark A Jacobs. 2007. Product Complexity: A Definition and Impacts on Operations. *Decision Sciences Institute* 38, 5 (Oct. 2007), 5. https://ecommons.udayton.edu/mis_fac_pub/94

[52] Brian C. Keegan. 2013. A history of newswork on Wikipedia. In *Proceedings of the 9th International Symposium on Open Collaboration (WikiSym '13)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/2491055.2491062

[53] Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications, Thousand Oaks, CA, USA. https://doi.org/10.4135/9781071878781

[54] Harris Kyriakou, Jeffrey V. Nickerson, and Gaurav Sabnis. 2017. Knowledge Reuse for Customization: Metamodels in an Open Design Community for 3D Printing. *MIS Q.* 41, 1 (mar 2017), 315–332.

[55] Man-Lin Li, Zhong-Lin Fu, Wei Guo, Lei Wang, Jian Ma, and Li-Wen Shi. 2023. Dynamic analysis of identifying user roles and evolutionary paths in collective intelligence design community. *Advanced Engineering Informatics* 57 (Aug. 2023), 102126. https://doi.org/10.1016/j.aei.2023.102126

[56] Tianshi Li, Elizabeth Louie, Laura Dabbish, and Jason I. Hong. 2021. How Developers Talk About Personal Data and What It Means for User Privacy: A Case Study of a Developer Forum on Reddit. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW3 (Jan. 2021), 1–28. https://doi.org/10.1145/3432919

[57] Chen Liang, Anhong Guo, and Jeeeun Kim. 2022. CustomizAR: Facilitating Interactive Exploration and Measurement of Adaptive 3D Designs. In *Designing Interactive Systems Conference*. ACM, Virtual Event Australia, 898–912. https://doi.org/10.1145/3532106.3533561

[58] Jiakun Liu, Haoxiang Zhang, Xin Xia, David Lo, Ying Zou, Ahmed E. Hassan, and Shanping Li. 2021. An exploratory study on the repeatedly shared external links on Stack Overflow. *Empirical Software Engineering* 27, 1 (Nov. 2021), 19. https://doi.org/10.1007/s10664-021-10028-y

[59] Thomas Ludwig, Volkmar Pipek, and Peter Tolmie. 2018. Designing for Collaborative Infrastructuring: Supporting Resonance Activities. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (Nov. 2018), 1–29. https://doi.org/10.1145/3274382

[60] Thomas Ludwig, Oliver Stickel, Alexander Boden, and Volkmar Pipek. 2014. Towards sociable technologies: an empirical study on designing appropriation infrastructures for 3D printing. In *Proceedings of the 2014 conference on Designing interactive systems (DIS '14)*. Association for Computing Machinery, New York, NY, USA, 835–844. https://doi.org/10.1145/2598510.2598528

[61] Kelly Mack, Megan Hofmann, Udaya Lakshmi, Jerry Cao, Nayha Auradkar, Rosa Arriaga, Scott Hudson, and Jen Mankoff. 2023. Rapid Convergence: The Outcomes of Making PPE During a Healthcare Crisis. *ACM Transactions on Computer-Human Interaction* 30, 1 (Feb. 2023), 1–25. https://doi.org/10.1145/3542923

[62] Mary Lou Maher, Mercedes Paulini, and Paul Murty. 2011. Scaling Up: From Individual Design to Collaborative Design to Collective Design. In *Design Computing and Cognition '10*, John S. Gero (Ed.). Springer Netherlands, Dordrecht, 581–599. https://doi.org/10.1007/978-94-007-0510-4_31

[63] David Mellis and Leah Buechley. 2012. Collaboration in open-source hardware: third-party variations on the arduino duemilanove. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM, Seattle Washington USA, 1175–1178. https://doi.org/10.1145/2145204.2145377

[64] R. Mies, J. Bonvoisin, and R. Stark. 2020. DEVELOPMENT OF OPEN SOURCE HARDWARE IN ONLINE COMMUNI-TIES: INVESTIGATING REQUIREMENTS FOR GROUPWARE. *Proceedings of the Design Society: DESIGN Conference* 1 (May 2020), 997–1006. https://doi.org/10.1017/dsd.2020.38 Publisher: Cambridge University Press.

[65] Audris Mockus, Roy T. Fielding, and James Herbsleb. 2000. A case study of open source software development: the Apache server. In *Proceedings of the 22nd international conference on Software engineering - ICSE '00 (ICSE '00)*. ACM Press, Limerick, Ireland, 263–272. https://doi.org/10.1145/337180.337209

[66] Jarkko Moilanen, Angela Daly, Ramon Lobato, and Darcy W. E. Allen. 2014. Cultures of Sharing in 3D Printing: What Can We Learn from the Licence Choices of Thingiverse Users? *Journal of Peer Production* 2015, Issue #6 Disruption and the law (Nov. 2014), 29. https://papers.ssrn.com/abstract=2440027

[67] Gordon Müller-Seitz and Guido Reger. 2010. Networking beyond the software code? an explorative examination of the development of an open source car project. *Technovation* 30, 11 (Nov. 2010), 627–634. https://doi.org/10.1016/j.technovation.2010.07.006

[68] Hari Suman Naik, Albrecht Fritzsche, and Kathrin M. Moeslein. 2021. Modularity in making: simplifying solution space for user innovation. *R&D Management* 51, 1 (2021), 57–72. https://doi.org/10.1111/radm.12427 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/radm.12427.

[69] Jeffrey V. Nickerson. 2015. Collective Design: Remixing and Visibility. In *Design Computing and Cognition '14*. Springer International Publishing, Cham, 263–276. https://doi.org/10.1007/978-3-319-14956-1_15

[70] A. Noort, G.F.M. Hoek, and W.F. Bronsvoort. 2002. Integrating part and assembly modelling. *Computer-Aided Design* 34, 12 (Oct. 2002), 899–912. https://doi.org/10.1016/s0010-4485(01)00145-2

[71] James Novak, I. and Paul Bardini. 2019. The Popular Culture of 3D Printing: When the Digital Gets Physical. In *Handbook of Research on Consumption, Media, and Popular Culture in the Global Age*. IGI Global, Hershey, PA, USA, 188–211. https://doi.org/10.4018/978-1-5225-8491-9

[72] James I. Novak. 2020. 500 days of Thingiverse: a longitudinal study of 30 popular things for 3D printing. *Rapid Prototyping Journal* 26, 10 (Oct. 2020), 1723–1731. https://doi.org/10.1108/RPJ-01-2020-0021

[73] Lora Oehlberg, Wesley Willett, and Wendy E. Mackay. 2015. Patterns of Physical Design Remixing in Online Maker Communities. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 639–648. https://doi.org/10.1145/2702123.2702175

[74] David L. Olson and Kirsten Rosacker. 2012. Crowdsourcing and open source software participation. *Service Business* 7, 4 (Nov. 2012), 499–511. https://doi.org/10.1007/s11628-012-0176-4

[75] OSHW. 2023. Definition (English). https://www.oshwa.org/definition

[76] Margit Osterloh and Sandra Rota. 2007. Open source software development—Just another case of collective invention? *Research Policy* 36, 2 (March 2007), 157–171. https://doi.org/10.1016/j.respol.2006.10.004

[77] Spiros Papadimitriou, Evangelos Papalexakis, Bin Liu, and Hui Xiong. 2015. Remix in 3D Printing: What your Sources say About You. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, Florence Italy, 367–368. https://doi.org/10.1145/2740908.2745943

[78] Spiros Papadimitriou and Evangelos E. Papalexakis. 2014. Towards laws of the 3d-printable design web. In *Proceedings of the 2014 ACM conference on Web science*. ACM, Bloomington Indiana USA, 255–256. https://doi.org/10.1145/2615569.2615660

[79] Mercedes Paulini, Paul Murty, and MaryLou Maher. 2013. Design processes in collective innovation communities: a study of communication. *CoDesign* 9, 2 (June 2013), 90–112. https://doi.org/10.1080/15710882.2012.716850

[80] Adam Pawlak. 2010. Challenges in collaborative design in engineering networks. In *eChallenges e-2010 Conference*. IEEE, Warsaw, Poland, 8 pages.

[81] Peter Pirolli and Stuart Card. 1995. Information foraging in information access environments. In *CHI '95: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. ACM Press, Denver, Colorado, USA, 51–58. https://doi.org/10.1145/223904.223911

[82] 3D Printing. 2010. r/3Dprinting. https://www.reddit.com/r/3Dprinting/ [Accessed 13-12-2023].

[83] Rohith Pudari, Shiyuan Zhou, Iftekhar Ahmed, Zhuyun Dai, and Shurui Zhou. 2023. Aligning Documentation and Q&A Forum through Constrained Decoding with Weak Supervision. In *2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, Bogotá, Colombia, 346–351. https://doi.org/10.1109/ICSME58846.2023.00043 ISSN: 2576-3148.

[84] Johnny Saldaña. 2009. *The Coding Manual for Qualitative Researchers*. SAGE Publications, London, England.

[85] Niloufar Salehi and Michael S. Bernstein. 2018. Hive: Collective Design Through Network Rotation. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (Nov. 2018), 151:1–151:26. https://doi.org/10.1145/3274420

[86] Kjeld Schmidt and Liam Bannon. 1996. Taking CSCW Seriously: Supporting Articulation Work. *Computer Supported Cooperative Work (CSCW)* 1 (1996), 7–40. https://doi.org/10.1016/B978-0-12-415040-9.50118-4

[87] Courtney Shaw. 2018. *Why Do Makers Make? Examining Designer Motivations on Thingiverse.com*. Honors Thesis. University of New Hampshire, Durham, Hew Hampshire. https://scholars.unh.edu/honors/430

[88] Wenxuan Wendy Shi, Sneha R. Krishna Kumaran, Hari Sundaram, and Brian P. Bailey. 2023. The Value of Activity Traces in Peer Evaluations: An Experimental Study. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (April 2023), 151:1–151:39. https://doi.org/10.1145/3579627

[89] Michael A. Stanko and B. J. Allen. 2022. Disentangling the collective motivations for user innovation in a 3D printing community. *Technovation* 111 (March 2022), 102387. https://doi.org/10.1016/j.technovation.2021.102387

[90] Evgeny Stemasov and Ali Askari. 2023. Design Where and When Necessary – In-Situ Remixing for Prototyping Electronics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI2023 Workshop [WS2] - Beyond Prototyping Boards: Future Paradigms for Electronics Toolkits)*. ACM, Hamburg Germany, 4.

[91] Evgeny Stemasov, Enrico Rukzio, and Jan Gugenheimer. 2021. The Road to Ubiquitous Personal Fabrication: Modeling-Free Instead of Increasingly Simple. *IEEE Pervasive Computing* 20, 01 (Jan. 2021), 19–27. https://doi.org/10.1109/MPRV.2020.3029650 Publisher: IEEE Computer Society.

[92] Evgeny Stemasov, Tobias Wagner, Jan Gugenheimer, and Enrico Rukzio. 2022. ShapeFindAR: Exploring In-Situ Spatial Search for Physical Artifact Retrieval using Mixed Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3491102.3517682

[93] Julian Stirling, Kaspar Bumke, Joel Collins, Vimal Dhokia, and Richard Bowman. 2022. HardOps: utilising the software development toolchain for hardware design. *International Journal of Computer Integrated Manufacturing* 35, 12 (Dec. 2022), 1297–1309. https://doi.org/10.1080/0951192X.2022.2028188

[94] Juan Tan, Dongqing Miao, and Qiong Tan. 2020. Empirical Study on Influencing Factors of Knowledge Product Remixing in OIC. *IEEE Access* 8 (2020), 34215–34224. https://doi.org/10.1109/ACCESS.2020.2974693 Conference Name: IEEE Access.

[95] Juan Tan, Congcong Qi, Xiaohui Gao, Jianle Lu, and Qiong Tan. 2022. Conflict or Collaboration—The Impact of Knowledge Endowment Heterogeneity on Remix in Open Collaborative Communities. *Frontiers in Psychology* 13 (2022), 11 pages. https://www.frontiersin.org/articles/10.3389/fpsyg.2022.941448

[96] Thingiverse.com. 2023. Thingiverse - Digital Designs for Physical Objects. https://www.thingiverse.com/about

[97] Yingchen Tian, Yuxia Zhang, Klaas-Jan Stol, Lin Jiang, and Hui Liu. 2022. What makes a good commit message?. In *Proceedings of the 44th International Conference on Software Engineering (ICSE '22)*. Association for Computing Machinery, New York, NY, USA, 2389–2401. https://doi.org/10.1145/3510003.3510205

[98] James Tooze, Sharon Baurley, Robert Phillips, Paul Smith, Edwin Foote, and Sarah Silve. 2014. Open Design: Contributions, Solutions, Processes and Projects. *The Design Journal* 17, 4 (Dec. 2014), 538–559. https://doi.org/10.2752/175630614X14056185480069 Publisher: Routledge _eprint: https://doi.org/10.2752/175630614X14056185480069.

[99] Tiffany Tseng and Mitchel Resnick. 2014. Product versus process: representing and appropriating DIY projects online. In *Proceedings of the 2014 conference on Designing interactive systems (DIS '14)*. Association for Computing Machinery, New York, NY, USA, 425–428. https://doi.org/10.1145/2598510.2598540

[100] Karl T Ulrich and Steven D Eppinger. 2015. *Product Design and Development*. McGraw-Hill, Columbus, OH, USA.

[101] Christian Voigt. 2018. Not Every Remix is an Innovation: A Network Perspective on the 3D-Printing Community. In *Proceedings of the 10th ACM Conference on Web Science (WebSci '18)*. Association for Computing Machinery, New York, NY, USA, 153–161. https://doi.org/10.1145/3201064.3201070

[102] Morten Warncke-Wang, Rita Ho, Marshall Miller, and Isaac Johnson. 2023. Increasing Participation in Peer Production Communities with the Newcomer Homepage. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2

(Sept. 2023), 1–26. https://doi.org/10.1145/3610071

[103] Marco Wirth, Sascha Friesike, Christoph Flath, and Frederic Thiesse. 2015. Patterns of Remixes or Where Do Innovations Come from: Evidence from 3D Printing. In *ECIS 2015 Research-in-Progress Papers*. AIS, Münster, Germany, 10.

[104] Pavlína Wurzel Gonçalves, Gül Çalikli, and Alberto Bacchelli. 2022. Interpersonal Conflicts During Code Review: Developers' Experience and Practices. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW1 (March 2022), 1–33. https://doi.org/10.1145/3512945

[105] Robert K. Yin. 2018. *Case study research and applications: design and methods* (sixth ed.). SAGE, Los Angeles.

[106] Shurui Zhou, Bogdan Vasilescu, and Christian Kästner. 2019. What the fork: a study of inefficient and efficient forking practices in social coding. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. ACM, Tallinn Estonia, 350–361. https://doi.org/10.1145/3338906.3338918

[107] Ali Gürcan Özkil. 2017. Collective design in 3D printing: A large scale empirical study of designs, designers and evolution. *Design Studies* 51 (July 2017), 66–89. https://doi.org/10.1016/j.destud.2017.04.004

A Components of the DrawBot

Table 2 summarizes and describes the components of the DrawBot.

Table 2. Components required to create and use the Drawbot, including hardware, electronics, and software.

Category	Sub-category	Description	Example(s) from Dataset
Hardware	Pen Holder	A gripping device that securely holds the drawing tool.	*"I want to explore swivel blade mount to cut materials."*
	Pen Lift	A mechanism that raises and lowers the drawing tool in the Z-direction.	*"Your pen isn't lifting enough. Try increasing the angle to servo from 35 to 45 or 60."*
	Guide Rails	Linear rods that provide stability and directional guidance for the clamshell.	*"Are the rods prone to bending when the Y-axis is fully extended?"*
	Clamshell	A component that moves along the guide rails to enable precise X- and Y-axis movement of the pen holder.	*"I've trimmed some material from the upper clamshell that was interfering with the lower clamshell posts."*
	Stepper Motor Mount	A housing for the stepper motors driving the movement of the clamshell.	*"Does putting the motors on top lead to more vibrations compared to on the bottom?"*
	Belt Guide	A belt and pulley system that transfers motion from the stepper motors to the clamshell.	*"Any suggestions on how to increase belt tension? Maybe by adding a belt tensioner attachment?"*
Electronics	Arduino Uno	A microcontroller board that interprets and translates drawing instructions for the CNC Shield.	*"A standard Uno lacks the horsepower to read files from an SD card and plot them. You would need a Mega Arduino."*
	CNC Shield	An extension board mounted on the Uno that precisely controls the motors.	*"Make sure the stepper drivers are correctly placed on the CNC board."*
Software	Inkscape & MI Extension	A vector graphics software used to create and edit vector graphics; extensions export graphics as G-code.	*"I'm having problems exporting intricate designs (like more than 5 paths); Inkscape crashes or is unresponsive."*
	UGS (Universal G-Code Sender)	A program that communicates to the Uno to send G-code, enable manual control, and reset the DrawBot.	*"Ensure that when using UGS to move the pen in either the X or Y direction, it shouldn't follow a diagonal path."*
	GRBL (G-code Real-time Boot Loader)	A firmware program that runs on the Arduino Uno that interprets G-code to control the DrawBot's movements.	*"You need to use the special version of GRBL that can manage commands for raising and lowering the pen."*
	Other Software	General software needed to design, develop, or run the DrawBot.	*"Check and make sure you're running Python version 3.X on your computer."*