

# Third-Party Developers and Tool Development For Community Management on Live Streaming Platform Twitch

Jie Cai

Pennsylvania State University  
University Park, USA  
jie.cai@psu.edu

He Zhang

Pennsylvania State University  
University Park, USA  
hpz5211@psu.edu

Ya-Fang Lin

Pennsylvania State University  
University Park, USA  
yml5563@psu.edu

John M. Carroll

Pennsylvania State University  
University Park, USA  
jmc56@psu.edu

## ABSTRACT

Community management is critical for stakeholders to collaboratively build and sustain communities with socio-technical support. However, most of the existing research has mainly focused on the community members and the platform, with little attention given to the developers who act as intermediaries between the platform and community members and develop tools to support community management. This study focuses on third-party developers (TPDs) for the live streaming platform Twitch and explores their tool development practices. Using a mixed method with in-depth qualitative analysis, we found that TPDs maintain complex relationships with different stakeholders (streamers, viewers, platform, professional developers), and the multi-layered policy restricts their agency regarding idea innovation and tool development. We argue that HCI research should shift its focus from tool users to tool developers with regard to community management. We propose designs to support closer collaboration between TPDS and the platform and professional developers and streamline TPDs' development process with unified toolkits and policy documentation.

## CCS CONCEPTS

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

## KEYWORDS

Community Management, Discord, Twitch, Live Streaming, Third-Party Developers, Moderation Tools, Extension and Bot Development, Community moderation, Platform Governance

## ACM Reference Format:

Jie Cai, Ya-Fang Lin, He Zhang, and John M. Carroll. 2024. Third-Party Developers and Tool Development For Community Management on Live Streaming Platform Twitch. In *Proceedings of the CHI Conference on Human*

*Factors in Computing Systems (CHI '24)*, May 11–16, 2024, Honolulu, HI, USA. ACM, New York, NY, USA, 18 pages. <https://doi.org/10.1145/3613904.3642787>

## 1 INTRODUCTION

The advanced technological infrastructure reduces the barrier to content creators and facilitates user-generated content (UGC) to reach users worldwide on many online platforms. The increasing number of content creators and the exponential increase of multi-modal content inevitably cause user and content management on many platforms because not all users and UGC follow the terms of services [58, 92]. On the one hand, many platforms promote good and civil behaviors to flourish in diverse online communities; on the other hand, they curb destructive behaviors and punish violators breaking community rules.

Many platforms provide technical support to empower stakeholders to perform most of the management tasks, such as automated moderation bots to support volunteer moderators for Subreddit [14, 19] and live streamers in Twitch channels [78, 96]. To meet the diverse needs of sub-communities, many platforms at the same time provide open-access application programming interfaces (APIs) for third-party developers (TPDs) to develop tools (e.g., bots, extensions, plugins) to help users manage their communities with specific needs and norms, creating a diverse and thriving market of third-party tools [62]. Additionally, existing platform-provided tools also receive critiques from users that they often lack accuracy, do not meet the needs of a specific group of users, and increase the workload of users to some extent (e.g., more time to set up, need to review cases to approve) [7, 40, 92]. Many users use third-party tools to supplement their community management [8].

Community management and moderation have been broadly discussed in HCI and CSCW (e.g., [45, 60, 97]). However, many HCI scholarships about community management tool design focus on tool users' perspectives and explore issues about how to moderate (automatically or manually) (e.g., [14, 16, 37]), when to intervene (proactive or reactive) (e.g., [9, 11, 77]), and what actions to take (account suspension or content removal) (e.g., [39, 49]). Tool users' perspectives highlight the role and importance of tools in their community management; the complaints about the platform's tools also highlight the need to improve the existing tool design. The tension between tool importance for end-users and tool complaints to platforms points to a research gap related to tool developers, especially third-party developers (TPDs), who work as mediators between

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

CHI '24, May 11–16, 2024, Honolulu, HI, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0330-0/24/05

<https://doi.org/10.1145/3613904.3642787>

end-users and platforms and develop diverse tools to support the platform's ecosystem. These TPDs are usually active community members who engage with the communities, identify issues, and react faster to crisis management than the platform does [7, 36]. TPDs' essential role in community management, in addition to the tension between users' needs and the drawbacks of existing tools, highlights the need to understand tool design from the TPD's perspective. In this study, we ask the following research question:

- What are the practices of TPDs regarding tool development for live-streaming community management?

This study aims to understand their discussion about tool development practices and identify their challenges. We chose a semi-private TPD community on Discord and crawled their discussion with topic modeling and in-depth qualitative analysis. We found that TPDs experience multi-layered policy regulation regarding tool development; they also work as mediators between the platform and the tool users to create a management tool ecosystem but face challenges from the platform regarding technical, social, and policy-related issues. In line with HCI's work to build safe video-sharing platforms [2, 67], this study contributes to existing research about online community management with a bottom-up governance structure from TPDs' perspectives. We highlight how multi-layered policy and stakeholder relationships affect TPDs' development process with implications to support their autonomy and agency.

## 2 RELATED WORK

### 2.1 Tool Development For Online Community Management

Managing communities to attract and integrate new users is essential for community growth. New users play a critical role in the development and sustainability of online communities because of their potential to introduce fresh ideas and perspectives and contribute to the diversity of content [1]. These new users can be temporal or regular help seekers or experts who can support others [82]. As existing members may leave a community for various reasons, attracting and integrating new members becomes essential for maintaining an active and vibrant community. Tools like bots and extensions play social and functional roles to augment community administrators' management.

Maintaining communities requires deliberate efforts from the community and its members. This thread focuses on, on the one hand, how to facilitate community building and engagement of community members, such as designing chatbot and CAPTCHAs to facilitate community discussion [48, 75], setting a good example for the rest of the community members to mimic normative behaviors [11, 77], providing rewards for community members' contributions (e.g., Karma point on Reddit), and showing recognition of loyal community members (e.g., Twitch badge system) [12]. In this thread, bots play important social roles, such as greeting newcomers [76], encouraging participation [47], and personalizing interactions and support [73, 100].

While many HCI scholars focus on the community norms and policy [13, 15] and algorithm design [50, 52, 61, 66, 86] to improve the governance structure and performance. A thread of research

focuses on the tool users, such as general online viewers [99], content creators [41], and community administrators [9, 40, 53]. This thread mainly focuses on bots' functional roles [83, 104], such as comment-filter tools [41], toxicity prediction tools [99], evidence-capture tools [87], and harmful content blurring tools [17].

Though scholars have investigated different mechanisms to support community management and moderation from both the platform and tool users' perspectives, we know little about how TPDs implement these mechanisms and develop efficient tools for community management.

### 2.2 Third-Party Developers and Collaboration

TPDs are actors acting on behalf of the platform owner developing applications, services, or systems to address the needs of the end-users [33]. Third-party development tends to be governed by the contractual relationship with the platform [6]. TPDs are rarely directly compensated for their development work; instead, they are offered a marketplace for their applications with a greater reach to end-users [94]. They are a necessary part of the platform's ecosystem, collaborating with the platform to exchange innovative ideas and co-create commercial values [72]. Platforms can diversify their services through a voluminous development in a wide range of application areas [5]. By doing so, platforms can remain lean in terms of resources and labor on the platform while incorporating the ideas of the developers' communities in cultivating a digital ecosystem [33].

To successfully build platform ecosystems, many platforms shift the focus from developing applications on their own to offering resources to facilitate TPDs in their development work [71], such as providing an interface between the platform and TPDs, to give these developers access to the core platform functionality to build applications. For example, the live streaming platform - Twitch provides an API for third-party developers to use part of the platform's functions to connect to tools they developed; these tools are faced by end-users such as streamers, moderators, and viewers. Facilitating third-party development is a way for the platform to transfer its design capability to end-users and, in turn, generate complementary assets in the form of applications with their potential values [89].

The existing HCI literature on developers focuses mainly on gaming developers [28–31, 55] or open-source software development, such as GitHub [38], and hardware development [64], with topics specifically focused on collaboration with support [27, 34], teamwork [31, 32] and conflict in design [22]. For example, interaction designers and software developers face challenges in communicating preexisting interactions. There is a need to design collaborative tools to facilitate the designer's ability to express and the developer's ability to implement complex interactive systems [63].

Third-party development in the community management and moderation context is unique in two aspects: First, third-party developers are not developers within the organization regarding work status and viewpoints. For example, professional and voluntary developers in open-source communities face collaborative challenges due to the different viewpoints and backgrounds of projects and programming [93]. Second, the community management and

moderation context makes open source less possible since the moderation tools often target and can be potentially utilized by violators. It is risky to share the development process details with the public. These TPDs often have a (semi-)private space to discuss and share their knowledge. Considering the professionalism and the sensitive context, it is necessary to understand their practices to support them. In line with recent work to understand the TPDs' development process [72], this study would focus on TPDs and understand their tool development practices regarding streaming management and moderation and their roles in the platform ecosystem.

### 2.3 Third-Party Developers on Discord Working for Twitch As the Research Context

Live streaming as a new medium provides mass communication in the chatroom [35]. While streamers are broadcasting with various low- and high-fidelity equipment [20], viewers can register a pseudonymous account and send messages in the chatroom. The streamer can read and respond orally to these messages. Often, the streamer would like to interact with viewers to build communities [80] and promote prosocial behaviors in the chat [75]. As communities grow, streamers appoint some volunteer moderators and deploy bots to effectively manage the channels and viewership.

We choose Twitch as the research context because 1) live streaming is a promising industry with interactive and multimodal design and has thrived in decades and evolved with novel technology, 2) it is a leading live streaming platform in the industry, its other stakeholders (e.g., streamers, viewers, volunteer moderators) has caught much attention in HCI (e.g., [10, 77, 98, 101]). Who developed tools for these stakeholders receives little attention. Twitch has an extension and streaming tool market for third-party tools with different categories, such as an extension on the go, viewer engagement, stats and trackers, loyalty and recognition, etc. It also allows other third-party tools to connect to the platform, such as BetterTTV<sup>1</sup> and Frankerfacez<sup>2</sup>; these are all commonly used extensions on Twitch [8]. There are also other personally developed bots, such as Sery\_Bot<sup>3</sup>, which is a timely and valuable bot for streamers to combat human-bot coordinated attacks in live streaming communities [7, 36]. All these tools are either partnered with Twitch or independently run by TPDs.

Discord can be considered part of Twitch's ecology. Many streamers use Discord as a semi-private space for their community management [10, 11]. Unlike other public spaces allowing users to view content without registering or joining the community, such as Reddit, GitHub, and Stack Overflow, Discord only allows users who have joined the community, namely the server, to view its content and engage. All users can easily access and join the server with a link invitation. In the server, some public channels are visible to all users; some private channels are only available to a specific group of users. Discord allows users to communicate with text-based and voice-based chats. Initially, streamers usually use Discord to have group voice chat with teammates when playing and streaming video games. Off the stream, Discord also works as a space for

streamers and their followers/fans to socialize. Nowadays, it works as a third space for people with similar interests.

Since Twitch and Discord are closely connected, many TPDs for these two platforms have formed communities on Discord. The leading author joined the informal online communities called the *TwitchDev* server. This server is run by non-staff volunteers such as moderators and administrators. TPDs will get a developer role in this Discord by the administrators of this Discord server if they prove that their program developments are building for Twitch users or using the Twitch-provided tools. The community has thousands of TPDs to share their experiences. It has more than 2000 active members daily, including Twitch official staff, TPDs, broadcasters, and viewers (as shown in Figure 1). It includes many categories and channels, such as *developer-showcase*, *looking-for-devs*, and *extension-help*. Usually, *lobby* is the primary channel where everyone sends their first messages, asks questions, and socializes.

## 3 METHODS

We focused on TPDs' tool development practices. We specifically refer to "tool" as "bot and extension" for community management, not systems or tools TPDs used to do development or programs they develop for Twitch or other platforms. We first crawled all the chat history of the Discord server. After we went through all the channels, we realized that the "lobby" contains the most extensive dataset, and all other channels are comparatively small. Then, we decided to focus on the "lobby" as the primary data source to run topic modeling to identify the bot-related topics. We use bot and extension development interchangeably based on TPDs' description. Next, we qualitatively analyze these topics to explore their practices and challenges.

### 3.1 Data Exploration and Pre-processing

We collected 45,376 data entries from the largest channel, "lobby." This data spans from October 25, 2018, to April 25, 2023. Each entry represents a message sent by a user in the channel and includes the content and timestamp of the message. Among these, we processed each message to remove English stopwords using the NLTK<sup>4</sup> library. Combining our reading and reference to prior work with similar data structure [59], we excluded messages with fewer than ten words post-stopword removal to ensure a more meaningful and contextual analysis. This preprocessing step aligns with standard practices in text analysis to filter out potentially non-informative short texts [91]. We obtained 8,219 data entries, and this dataset comprised 279,581 words (1,567,282 characters). Based on our research questions and feasibility, the data collection from the channel was a one-time and fixed process. We did not consider any changes in the channel before or after the data collection.

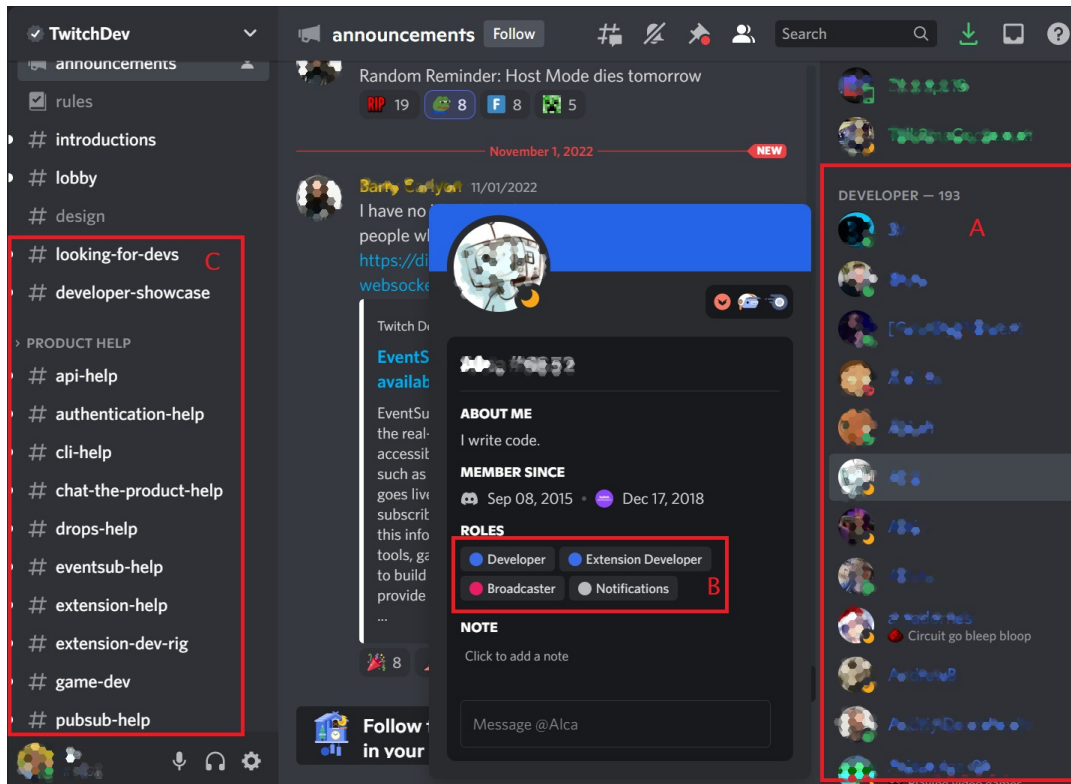
**3.1.1 Topic Modeling.** Topic modeling facilitates the discovery of hidden thematic structures instead of relying solely on keyword retrieval [3]. Discord comments are short texts presenting sparse, noisy, and non-informative words, posing challenges in generating accurate topics. Prior research has highlighted the effectiveness of topic modeling in extracting thematic structures from such short-text datasets, making it a valuable method for discerning underlying

<sup>1</sup><https://betterttv.com/>

<sup>2</sup><https://www.frankerfacez.com/>

<sup>3</sup><https://serycodes.carrd.co/>

<sup>4</sup><https://www.nltk.org/>



**Figure 1: A screenshot of the *TwitchDev* server on Discord. (A) the list of 193 active developers online; (B) a user with many different tagged roles verified by the server administrators such as developer and extension developer; (C) various channels for developers to discuss and collaborate on technical issues related to bot development.**

themes and patterns [56, 69]. We employ Latent Dirichlet Allocation (LDA) [4] for topic modeling. The LDA, alpha, and beta parameters are set to the default value of 1.0 divided by the number of topics. We performed a cluster analysis on the pre-processed data and selected one potentially more dominant cluster quantity in the results (Figure 2a), which inspired us in the number of thematic trends (8, red line). Coherence score is considered one of the most effective metrics for topic modeling [84]. Hence, the number of topics is determined according to coherence score [85] of the extracted topics given various possible topic numbers ranging from 7 to 20. Figure 2b shows that the consistency model suggests using eight topics as the optimal number. Table 1 illustrates the computation extraction of topic distribution for generated results from using LDAvis [81], where each result may be generated by multiple topics, and each topic contains several keywords.

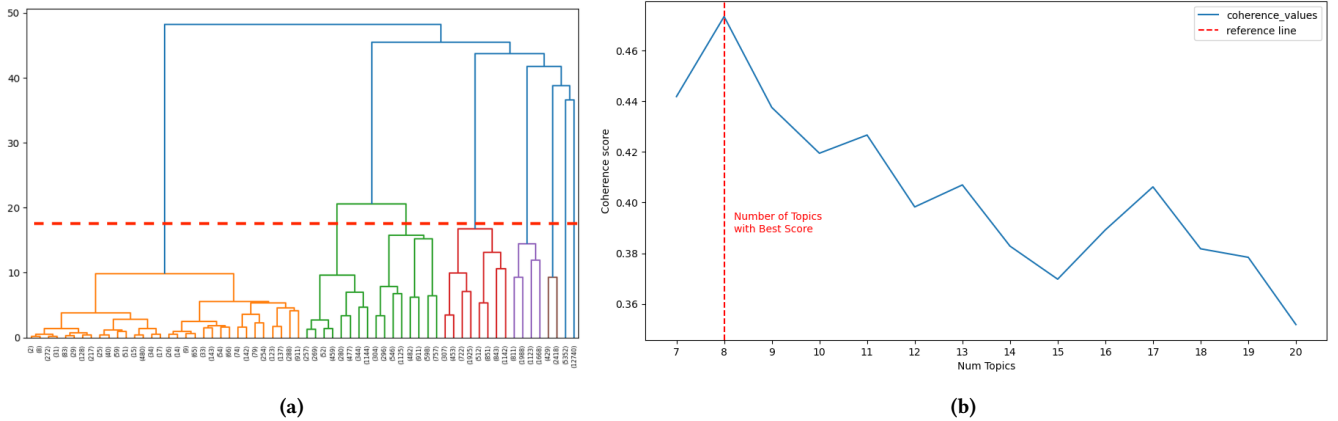
Because we aim to focus specifically on the subjects of “bot” and “extension,” we have chosen Topics 4 and 6 as the main candidates for further analysis. In the results of our topic modeling, Topic 4 prominently highlights “extension” as one of its pivotal terms, with a notable weight of 0.025, positioning it as the top term. This implies that discussions or narratives related to extensions are predominantly captured within this topic. Similarly, Topic 6 incorporates the term “bots” with a weight of 0.010, ranking it sixth, suggesting that it encapsulates discussions revolving around bots.

Further scrutiny of the LDA topic modeling outcomes on the corpus reveals that besides Topics 4 and 6, other topics don’t extensively cover contents related to “bots” and “extension.” While this doesn’t rule out the presence of related content in other topics, focusing on Topics 4 and 6 can more efficiently and comprehensively delve into discussions regarding “bots” and “extension” within this channel. We organized the data labeled as Topics 4 and 6 in the topic modeling, and ultimately, we obtained 2,179 comments, constituting the “corpus” used in this study.

We clarify that we focus on bots and extensions because they are closely related to the development we aim to discuss in this paper. We also quickly reviewed other topics and found they were about general information inquiries, specific tech support, and socialization among TPDs. These are related to the Twitch platform or their programming experience but not specifically related to third-party tool development. For example, topics with “UserVoice, Twitch, and API” include limited comments on tool development. Still, these are eventually captured by the bot and extension topics, as we mentioned in the relevant findings sections.

### 3.2 Thematic Analysis

We undertook an exhaustive analysis of 2,179 comments (hereinafter referred to as the “corpus”) pertinent to the topics of “extension” and “bot.” After completing the corpus compilation, it



**Figure 2: Topic Modeling Results:** (a) Result of Agglomerative Hierarchical Clustering Dendrogram. The x-axis represents the clustering results of sample data, with numbers indicating the quantity of samples contained within each cluster. The y-axis represents the similarity during the merging of different clusters, where greater distance signifies lower similarity. (b) Result of Coherence Model. The x-axis represents the number of topics, and the y-axis represents the coherence score.

	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6	Term 7	Term 8	Term 9	Term 10
Topic 0:	0.043**"twitch"	0.025**"help"	0.020**"discord"	0.019**"party"	0.014**"com"	0.010**"support"	0.010**"account"	0.010**"new"	0.010**"third"	0.010**"report"
Topic 1:	0.022**"channel"	0.017**"still"	0.013**"would"	0.013**"want"	0.011**"people"	0.010**"well"	0.010**"name"	0.009**"stream"	0.009**"since"	0.009**"channels"
Topic 2:	0.022**"game"	0.017**"twitch"	0.012**"time"	0.012**"see"	0.012**"use"	0.011**"example"	0.010**"work"	0.009**"thing"	0.009**"uservoice"	0.009**"guess"
Topic 3:	0.039**"thanks"	0.030**"answer"	0.023**"http"	0.020**"quest"	0.015**"got"	0.014**"answers"	0.014**"join"	0.012**"barry"	0.010**"help"	0.009**"language"
Topic 4:	<b>0.025**"extension"</b>	0.019**"chat"	0.015**"twitch"	0.015**"good"	0.012**"message"	0.011**"use"	0.009**"code"	0.009**"token"	0.009**"need"	0.009**"bits"
Topic 5:	0.072**"twitch"	0.030**"tv"	0.019**"api"	0.016**"contact"	0.015**"ask"	0.015**"dev"	0.015**"user"	0.012**"send"	0.012**"luck"	0.012**"know"
Topic 6:	0.018**"one"	0.017**"use"	0.016**"something"	0.015**"thank"	0.011**"already"	<b>0.010**"bots"</b>	0.010**"wait"	0.010**"go"	0.009**"looking"	0.009**"think"
Topic 7:	0.033**"twitch"	0.031**"support"	0.030**"link"	0.026**"us"	0.022**"found"	0.021**"looks"	0.021**"help"	0.019**"request"	0.015**"please"	0.015**"friends"

**Table 1: The results of the LDA topic modeling analysis on the corpus. Each topic is represented by weighted terms (top 10 highest). The subtopics we focus on ("extension" and "bots") are highlighted in red, and the topics we choose for further in-depth analysis are highlighted in gray.**

was disseminated among all authors for the coding process. We adopted a hybrid technique that combined inductive and deductive coding for theme extraction [21]. Our coding process is based on the definitions by Corbin & Strauss [79] and is divided into three critical phases: Independent Coding (Open Coding), Codebook Development (Axial Coding), and Comprehensive Coding (Selective Coding).

During the Independent Coding phase, each author autonomously embarked on an initial, open-ended coding of a segment of the corpus. This approach ensured that every entry was evaluated without the predisposition of other viewpoints. We treated each entry as an independent observation. Each of the three primary coders was randomly assigned 300 comments from the corpus for individual assessment. The Codebook Development phase followed, wherein all authors congregated to compare and consolidate their independent coding results. This step involved a meticulous comparison of individual codes, striving to establish a unified label for each comment based on collective consensus. For clarity, comments were sequentially labeled: for instance, the first comment was encoded as '1. Extension workshop', the second as '2. refresh before expiration of subbed topic', and so on. The objective was then to ascertain the applicability of these codes to subsequent comments. At this stage, coders try to find the relationships between different concepts and

connect and organize the categories and concepts, and an embryonic codebook is formulated. The authors revisited the corpus with this preliminary framework to ensure coding uniformity. Discrepancies or ambiguities detected during this process were addressed and refined through iterative discussions. Multiple sessions culminated in finalizing the codebook, comprising 59 primary codes. For streamlined subsequent coding, two auxiliary codes, '0. irrelevant' and '60. not listed' were also integrated.

In the final Comprehensive Coding phase stage, three coders undertook independent coding of the residual data. Since the reliability will decrease as the number of coders and codes increase [51], we acknowledge that we had a low coefficient (Fleiss's kappa is .30) considering that we have three coders with a large codebook. Instead, we changed our strategies to have hours-long meetings to review and discuss each other's coding with notes and revisions [18]. Regular weekly meetings facilitated discussions on coding disparities, fostering alignment and consensus. During our meeting, we start by sharing our overall thoughts about the weekly coding updates. Then, we follow a protocol where a coder starts reading the comment and interpreting their code, followed by other coders interpreting it. Finally, we vote for one of the three codes or come up with an emerging code to better cover the core meaning of the comment. At the same time, coders exchange notes, which helps in generating topics and prospective categories. The coding process

lasts for about six weeks, and each meeting lasts for about two hours. This iterative process, intertwined with functional codes, led to the categorization of codes and the formation of overarching themes. The trio of coders subsequently regrouped to critically assess and compare their coding interpretations, aiming for optimal consistency and precision. In instances of disagreements or ambiguities, collective discussions paved the way for resolutions. Conclusively, we pinpointed high-caliber comments for each category, fine-tuning their alignment to distill the final themes. Themes and sub-themes with codes are summarized in [Appendix A](#).

### 3.3 Positionality Statement and Ethical Consideration

Though we're not developers, our expertise and research interests align closely with this topic. All authors are online community researchers. The lead author is a live streaming user and researcher and has observed this Discord community for two years. The other team members, also active Discord users, joined this particular Discord community months before data collection to understand TPDs' discussion about their practices. This study analyzes publicly available comments without direct interaction with the content creators. This approach aligns with the guidelines set by the University Institutional Review Board (IRB), which does not classify it as "human subjects research." In our commitment to upholding user privacy, we followed Internet research guidelines and have applied stringent anonymization measures to the research data [26]. This involves the meticulous removal of any identifiable information, such as user IDs, ensuring the anonymity of the data sources and protecting the privacy of individuals involved. The data reporting includes only textual comments.

## 4 FINDINGS

Discord provides a space for stakeholders to socialize, share information and resources, and have Q&A conversations in general. Regarding their tool development practices, there are mainly four high-level themes: (1) TPDs shared positive experiences interacting with streamers and other TPDs to exchange and innovate design ideas; (2) the platform's technical updates caused uncertainty in their working schedule and extra labor for tool compatibility; (3) they also complained about the policy at the different levels restricting their developing agency with the intent to initialize changes; and (4) they appreciated the TPD communities providing social support and opportunities to learn from each other. Findings are summarized in [Appendix B](#).

### 4.1 Design Ideas Exchange and Innovation

**4.1.1 Streamer's Need and Design Ideas.** TPDs talk with streamers in this discord server, and some of the TPDs are also streamers. Streamers mainly ask for advice or suggestions about extension developments that could fulfill the needs of their streaming, such as managing their channels, avoiding spammers and bots, and creating alternative tools that can edit their streaming interfaces easily for better customization or accessibility purposes. For example, a streamer wanted to create their stream editing tools to replace Streamlabs (a commonly used stream editing software).

"Hi guys, I'm curious if it's possible to self-host a streamlabs lookalike. Basically my problem is streamlabs' html/css editor is pretty wonky, and their documentation isn't that great, I'd like to make my own one for my channel instead. Is it possible?"

Many streamers want extensions that can provide rich interactivity to their audience. For example, some streamers want to be able to respond to their viewers promptly or interact with specific kinds of audiences, such as new followers; some want to enrich viewers' presentation ability; some seek flexible approaches of bit use; and some want to coordinate with bots more smoothly that viewers would not confuse broadcasters' and their bots' messages. Though streamers already have plenty of extensions to use, they still reach out to TPDs for help because 1) they want to create customized features that are too personalized to find existing extensions; even if they try to develop extensions by themselves, they need technical advice to implement their ideas; 2) some streamers work with TPDs to develop specialized extensions, yet they need help when their collaborators are unavailable.

**4.1.2 Exchange Design Ideas Among TPDs.** TPDs shared various kinds of extension design ideas on the Discord server. One of the most shared extension categories is supporting streamers to entertain, manage, and interact with their viewers, such as "finding the most popular follower of a given channel," "altering stream title," "making a link to a specific channel on Twitch using their streamer id instead of username," and "extended the chat functionality." Another commonly shared category is crisis management. TPDs shared bot designs that can prevent streamers from being attacked by malicious bots.

"gotcha! ya, I think a bot that could identify with some reasonable certainty, users that are bots on the fly, would be \*super\* useful. it would also make the tool distinguishable from all of the other cross-channel shared list ban bots. just an idea"

Some TPDs shared their sandbox version of their bots before publishing and collected feedback from their peers to advance the features and functions. For example, they would like to showcase their bots.

"Made this extension that will save you a lot of time! Let me know what you think. It isn't live yet. It costs like 5 dollars to publish an extension, so I just wanted to see if people actually want something like this. [link to the showcase video of the extension]"

Some TPDs might share their successful or failed experience, provide explanations, and exchange design ideas. After reading other people's design ideas, some TPDs provided their ideas as feedback. These ideas could be very different from their peers' ideas or an improvement built on the top of the proposed ideas. For example, when a TPD talked about extensions interacting with viewers, another TPD shared their idea that allows viewers to "create an incentive for the streamer to do a certain thing in a game" by just-in-time crowdfunding with bits. Furthermore, TPDs resonate with each other's designs and adapt them into their designs accordingly. As a TPD shared, "Sounds fun, i have something similar and started as a stream avatar clone I saw other streamers use and thought that

was *pretty cool to have so I wrote my own.*” They even enhanced the original design when making their own ones. The review process may deny ideas related to legal and ethical issues (e.g., a crypto tipping extension).

## 4.2 Impact of Twitch Technical update

**4.2.1 Schedule Uncertainty.** Some TPDs unfamiliar with the review process asked many questions about the update uncertainty. For example, a TPD team asked questions about the impact of a UI/UX update on the review process.

“So we’ve been working on a component extension for a bit, and have been building it for before the UI/UX update that takes effect on July 31st. We haven’t submitted it yet, but are wondering how this deadline affects us because we’d like to move into the next step for testing by the 31st (white listing a streamer or two and having their entire audience be able to interact with it). What should we know if we don’t submit by that deadline, or if we do and it’s sent back for updating and we don’t resubmit by the 31st. Just trying to cover all our bases.”

In this case, the TPD team was wondering how a new UI/UX update would affect the evaluation of their extension and if their extension testing would be valid if they tested it before the UI/UX update. They might need to adjust their development schedule accordingly.

Sometimes, the review time was so long with no feedback, especially before a long break, as a TPD expressed, *“i just hope they get to it before closing down the office till next year.”* They hoped the Discord community managers could personally contact the Twitch team to check what was going on since they had no feedback for months.

“It really is awesome that you guys are streaming etc... but we ([link to a third-party developed site]) are waiting already for 5 months (!!!!!!!!!) to get our extension finally approved. Can you please finally take care of it???? PLEASE!!!!!!!!”

**4.2.2 Extra Labor Caused by Technical Updates.** TPDs think any technical update needs coordination with TPDs and considering TPDs’ needs; otherwise, TPDs might be confused and have difficulties adapting. TPDs discussed various issues when the API version changes from v3 to Kraken (i.e., v5) to Helix.

First, the previously worked methods might not be supported anymore, and TPDs spend extra effort adapting to the new version. For instance, TPDs can not access Twitch APIs with methods other than the OAuth flow, the authentication flow designated. Furthermore, many features initially supported might still be under Twitch’s development roadmap, so TPDs need to figure out approaches to bridge this gap.

Even if TPDs migrated their extensions to the new version, they might feel nervous as they were unsure whether the migration would work. As a TPD shared, *“pretty nervous, just pushed out the update that migrates my browser extension from using v3 to using IDs and helix and some v5 fallbacks.”* With the extra workload, TPDs think their innovative momentum has been stopped, especially

when TPDs were just introduced to new Twitch tools to help them develop new extensions.

“when there have been so many amazing work and progress recently to help developers like me become familiar with developing Extensions and building out better tools like the new Dev Rig for us to just start being able to iterate so much more quickly and efficiently on them and brainstorm greater and more ambitious interactions... And not only that, all the time we’ve invested into learning how to work with Extensions just so we can start really pushing out code now has to be shifted to fixing up old code. That’s why this sucks.”

Second, TPDs are confused about Twitch API usage and wonder how to use it in what circumstances. Otherwise, the extensions might be banned. For example, a TPD was confused about why they received a warning email from Twitch to stop using v3 (an old API version) because it did not allow TPDs to use v3 APIs anymore. Yet, the TPD thought they had already updated their extensions. Another issue TPDs mentioned is that Twitch did not give them enough time to adapt to the new version. As a TPD complained, *“I have the same problem on my public extension. Just not had time since we got told about it two/three months ago to fix it.”*

Without coordination, TPDs used their ways to adapt to the technical updates and keep extensions working. Some of them might fake the new process to skirt the update; as a TPD shared, *“the biggest pain point for me moving my bot away from IRC chat commands to Helix API calls was faking the oauth flow once to get a token for it.”* Some TPDs try to predict Twitch’s future updates with some clues they observed and adapt to the prediction in advance. For example, they try not to build extensions with those APIs they guess would not be supported by Twitch anymore, even if there are merits in using these APIs. Some removed the features that they knew would no longer be supported. While some TPDs took precautions against new updates, some believed the old APIs would last until Twitch created new APIs that could fulfill the exact needs.

TPDs have different opinions about Twitch’s technical updates. Some TPDs think Twitch must improve its infrastructure for future developments, even though that means there would be some costs TPDs need to take. Yet, some TPDs think the technical updates worsened the Twitch platform and damaged the TPD community. For example, some TPDs complained that the new changes worsened the experiences of the platform users, such as streamers, especially when the change is drastic.

Moreover, the updates might deteriorate the extension ecosystem by discouraging users from installing extensions. For example, without clicking on the extension, users can only preview an extension via a small icon; a TPD said that *“it makes it so much tougher for any of our extensions to get used”*. Similarly, TPDs complained that the new extension framework made developers further compete with each other over users’ usage as Twitch reduced the number of extensions users could use at the same time. As a TPD complained *“all the exciting and innovative interactions that Extensions were promised to be are being forced to compete for the single Overlay slot and that’s really unfortunate.”* Some TPDs thought that their issues were not fixed because of the unimportant and little business

value of the TPD market on its platform and guessed the reason for the delay and the lack of feedback.

### 4.3 Policy and Regulations at Different Levels

**4.3.1 Platform Policy Update.** TPDs are confused about the platform policy implementation and use Discord as a channel to ask for information about how the policy is implemented or ask for resources about the location of the most updated and recent policy so that they can adjust their expectations about what they should do. Some TPDs are confused about their bot ban because they designed very simple bots following some simple commands with the Twitch policy, but they still got banned. Sometimes, TPDs might try to bypass the limitation of the policy and apply alternatives or some other available features to meet their needs.

“TPD A: Also there are anti-spam measures in place (the specifics of which aren’t disclosed). So if you send the same message frequently, or send unsolicited whispers, you may get caught by the anti-spam protections and blocked from whispers.

TPD B: I mean is it possible, for example sending one thousand whispers maybe in 1000 separate days?”

In this case, sending whisper messages with a bot might trigger the anti-spam measures, so the TPD considered another way to avoid the measure (e.g., in 1000 separate days). Some TPDs feel the policy enforcement is unfair and would be against the rules, though they already know the violation: *“I definitely do something that goes against their rules (logging chat messages from channels). But at the same time so does streamcharts, and they’ve been around for years.”*

TPDs complained about the Twitch bot review process, which lacked transparency and stability, so no one knew how the mechanism worked. A TPD explained that even though the bot might be verified successfully, it could still be blocked if it triggered Twitch’s anti-spam measure and highlighted the power Twitch had to make the final decisions: *“You may get verified, you may not and there’s no telling how long it may take for a decision. But don’t put your hopes on it.”* The uncertain review process led to folk theories and rumors about how the review process worked or the platform’s intentions regarding TPDs. As a TPD said, *“guess it’s like two people doing those reviews... Guess extensions don’t make so much money compared to the rest.”* The frustration might make TPDs think Twitch does not want to have TPDs develop extensions, as a feeling a TPD shared, *“why else would they give us such a hard time.”* Some TPDs considered the potential difficulties of the Twitch side and thought it was unstable due to many factors. As a TPD said, *“usually around 2 working days. Then it will depend of your extension and its complexity, how the review team can test it, how many extension have to be reviewed, etc.”*

TPDs also discussed the boundary of Twitch policy regulation and can’t control external extensions from other platforms like Chrome extension. Since different platforms had different policies, some platforms might approve the design, but Twitch would finally reject it. A TPD said, *“IGDB will allow images that \_don’t quite\_ meet the requirements, but the Twitch sync will reject images that don’t \_quite\_ meet the requirements.”* The need to use different platforms for bot development and the inconsistency of policy enforcement cause disruption and extra labor for TPDs.

TPDs asked about copyright issues and the protection of intelligent property. The property protection issue is from both the TPDs’ and platform sides. On one side, TPDs wonder if they have the power to control the ownership of their extensions and design ideas and have concerns that Twitch might take their bots with relevant benefits. For example, a TPD asked, *“okay @name and... do I have full control on my extension? like it’s my right? like twitch not going to steal it or take credit for it, and twitch in away would show that I made it and it is my and even to a certain extent provide legal protection.”* Someone is even afraid and joked that they should have a lawyer for a personal project because of the confusing policy regarding using content with explicit permission from the content creator. Conversely, some TPDs also questioned whether they could use Twitch’s property for their bot development, such as using Twitch emotes for their extension and ‘Twitch’ in their bot names. A TPD specifically mentioned that they developed an app for WatchOS and asked whether they call it *“Twitch Chat for WatchOS”* and then realized that it is not allowed. Otherwise, they might face lawsuits.

**4.3.2 Data Privacy and Policy.** TPDs need to access data for their apps to function properly while being careful about Twitch’s data privacy and policy in terms of services (TOS) and copyright law (the Digital Millennium Copyright Act, DMCA). If bots store unpermitted content (e.g., chat history without permission), Twitch can ban them.

Third-party apps need permission from streamers or viewers to access data on various occasions. For example, when they want to store data for their viewers, they need to identify viewers at first. ID share is useful for an extension to identify and store user data and support the work of an extension. Similarly, if TPDs want to get data from a channel, such as subscription data, they need to make efforts to contact the broadcaster for their permission. However, TPDs might think asking permission for data access is difficult for extension development. Some useful functions might require personal information that users are unwilling to share. For example, TPDs think viewers might not want to share email addresses even if it is the only reliable way for third-party apps to communicate with viewers.

TPDs might breach data policy due to various reasons. First, TPDs might be confused about what scenarios about storing user-generated content violate Twitch’s TOS. Sometimes Twitch provides data access APIs without preventing TPDs from using them accidentally as a TPD warned another TPD, *“just because the API may technically provide a means to certain data, does NOT mean you can use that data however you choose”*. Providing data access APIs does not mean that data access is legal, yet some TPDs might not be aware of it. Sometimes, TPDs think they own the data, such as chat logs in their own channels, but they are unsure whether it violates data policy. Second, TPDs might think Twitch would not punish them because Twitch would not find their policy violation, they are non-profit, or they have only a small user base.

However, other TPDs might behave more cautiously. A TPD suggested their peers take Twitch policy seriously, as there are several reasons why Twitch has not taken action against the apps that have broken the data policy.



“Just keep in mind, the legal process is slow so just because you haven’t gotten a Cease and Desist yet doesn’t mean they aren’t tracking activity and logging things for potential legal action at some point down the line. Secondly, even if you’re not making money or you have a small userbase, Twitch can and have sent C&D letters and shut down small sites in the past. Finally, you have no clue what legal agreements others may have with Twitch. Some sites do enter into agreements with Twitch to use the API in certain ways. So it’s pointless to say “well they do it!” when you have no clue if they have permissions from Twitch or not.”

In this case, the TPD reasoned why they suggested not to break Twitch policies. It is because it is possible that Twitch acts slowly or different apps have different legal arrangements with Twitch.

Even if Twitch and the copyright law protect data, some TPDs can still access data without permission from Twitch users. First, TPDs might consider getting the user’s permission from another third-party app. It is unclear if users are aware of the share of permission. A TPD suggested their peers, “... *interact with another 3rd party application that geolocates users when you have permission to do so.*” Second, data leaks can result from the trust between stakeholders, such as streamers, moderators, and TPDs. TPDs might be delegated by a moderator without the permission of the streamer: “*Also a streamer may trust their mods, but mods can delegate ban functionality to 3rd party apps unbeknownst to the streamer.*” Third, TPDs might omit Twitch’s data regulation and hide their misconduct. Neither Twitch nor Twitch users could avoid data breaches. Someone said, “*The bots that are worse such as those that log chat messages don’t even need to be logged in to chat, so you’ll never see them in chat and have no way to prevent it.*”

After TPDs get permission to access and store user data, TPDs could use Twitch services to validate their users before performing data access from databases. Yet, TPDs might leak data if they do not know how to keep the key to data access safe, as the process of validating users is complicated. For example, if TPDs do not know that they need to keep the authorization token and the refresh token safe, they might unknowingly leak data.

**4.3.3 Discord Community Policy.** TPDs think this Discord is supposed to be an inclusive space for various kinds of people, including TPDs with different levels of knowledge, streamers, and even viewers who want to discuss Twitch-related tool development.

“This is a community for everyone on Twitch. TwitchDev is the main focus but its more about creating and building tools for everyone. So broadcasters and viewers are definitely welcome to discuss the future of the platform in terms of tools.”

A few expectations and rules make the environment comfortable for everyone. TPDs expect people to have explored the questions before asking for help in Discord, as a TPD said while answering a question, “*I’d assumed you’d already looked in the docs before coming here.*” Besides expectations, this Discord server also sets its own rules. First, people should not directly message someone without permission to prevent people from getting bothered. They

especially think it is important not to message Twitch staff without their agreement directly, and disturbing Twitch staff might worsen the messengers’ situations. Second, people are allowed to post recruitment messages. Yet, they are not allowed to share their CVs or seek developer jobs or other roles, such as moderators, because it would disrupt the interaction in the community. Third, due to the slow reply speed of Twitch support and the lack of legal support from Twitch, Some TPDs resort to this Discord server for Twitch support or legal advice because some Twitch staff are here. Yet, the community prefers not to provide advice as they are not experts; as a TPD said, “*Well, some staff here. But this is not a Twitch support server nor a space to get legal advice from.*” Fourth, legal questions are commonly asked, but TPDs think it is inappropriate to provide advice for legal questions because they are not lawyers. Different people may interpret the law differently. Some TPDs would share their experiences addressing legal issues; when they do so, they disclaim that their sharing is not legal advice.

**4.3.4 Bot/Extension Design Rules.** To design Twitch extensions and bots, TPDs must be familiar with complicated Twitch services and understand how to interact with them. An easy way for TPDs to interact with Twitch services is by following Twitch’s design patterns. Yet, TPDs are confused about a specific function’s affordances, attributes, or constraints. Confusion about Twitch-provided functions can give TPDs different ideas about how to use Twitch resources to design a bot. Take the Whisper function as an example; one party suggested their peers not to use the Whisper function because it was not reliable, according to their experiences. Yet, the other party said the Whisper function worked fine in their bot systems. The suggestion differences might result from understanding the prerequisites for using the Whisper function, such as passing Twitch’s spam filter.

TPDs also questioned the extension development involving compatibility, dependency, and adaption on other platforms, such as extensions for the various browsers and mobile systems. When they develop Twitch extensions on other platforms, they also need to comply with the rules of other platforms. For instance, a TPD said they need Apple Developer Program members to develop extensions on iOS Twitch apps. The compatibility of developing tools was also a concern, as it can be the source of errors. To develop a Twitch extension, TPDs had to use multiple tools; as a TPD shared, “*I tried to use the websocket addon to control it, as we use BlackMagic kit for production, along with CasparCG and VMix, so being able to automate certain things would be huge.*” TPDs found it difficult to manage the tools and resources. As a TPD shared, “*Yeah I’ve been working with React on the Extension I’m doing and yeah creating the files was easy. It was everything after that and having to deal with 3 tools being thrown at me along with it.*”

TPDs discussed various kinds of risks and costs while developing Twitch extensions. For example, as an ecosystem, TPDs-made tools aim to support development. Yet, TPDs should evaluate if using non-Twitch-made tools would cost extra effort or contain hidden risks. Furthermore, TPDs discussed risks associated with security, such as the loss of secrets (i.e., keys to data access generated by Twitch) and data leaks to non-Twitch organizations. TPDs also consider operation and maintenance costs. Some developers reduced the costs by choosing cheaper but compromised solutions or closing

their services when paying more for server operations was no longer affordable. TPDs discussed various options to transfer the costs to the users or other TPDs, such as allowing other TPDs to use it for free at the beginning and charging a fee after they are used to the tools, and making extension open source so that streamers can use to create their version for their channels.

Bot design should also consider users' perceptions and experiences. For example, TPDs suggest peers not use the Whisper function because users do not have a good experience interacting with it since spammers abused the function. Broadcasters might hesitate to allow an extension to send chat messages on their behalf. There is a need to balance viewers and broadcasters' agencies regarding using an extension. TPDs need to consider efficiency and performance when designing an extension. They need to consider the worst-case scenarios and think if they can and how to handle them.

**4.3.5 Complain and Initiate Policy and Platform Changes.** TPDs discussed actions they can take to support better extension development. On the one hand, TPDs can voluntarily have a consistent and coherent pattern, such as using a unified extension repository to support better extension searchability.

“...the “best” way to simplify plugin installation is for us to have a remote plugin repository that users can download right in the program unfortunately. I’ve discussed this in detail with my other guys and that’s really the sort of route we’d end up having to go for it to be friendly towards users and satisfy the “plugin discovery” needs of plugin creators”

They also advocate for their peers to make extensions more friendly and accessible. For example, some TPDs call for awareness for streamers with disabilities (e.g., streamers with fibromyalgia).

“It’s a great topic to bring up though, a couple of my good friends who stream have fibro, one of which was even on a Twitchcon panel on streamers with disabilities and they are helping try to create discussion around this themselves. I’m not sure where the best place to have such a discussion would be, but it’s certainly something that needs to happen, and continue to happen to raise awareness.”

On the other hand, TPDs could use UserVoice (a Twitch platform that allows people to provide design ideas to Twitch) to suggest Twitch make their UI more friendly for extension use. TPDs use Discord to test water if other TPDs would support their suggestions. They also promote suggestions and ask people to upvote: “*Maybe, it is better to start small. I’m planning on creating a uservoice to introduce public reviews for extensions from streamers who have installed the extension? Would you guys support something like that?*” In particular, they mentioned that they think Twitch could consider changing the petition if a significant number of developers are advocating it. They also consider asking streamers to vote for their petitions, especially those they work for.

## 4.4 Seeking and Showing Support

**4.4.1 Information and Instrumental Support.** TPDs seek various information support in Discord because TPDs might not find good

tutorials, documents, or official tech support when they want to develop Twitch third-party extensions. For example, new TPDs seek advice for learning paths, tips, examples, and experiences for Twitch extension development, such as the languages to use, what kinds of projects are good to start with, and documentation to start quickly.

Some of the new extension developers are not new to programming but new to Twitch extension development. They seek templates or packages to guide them to pick up the developing patterns quickly. For example, a bot developer looked for methods to make their bot programs work as extensions. TPDs also seek practical development advice, such as analysis about choice for developing tools, efficient algorithms, feedback on current development plans, and suggestions for debugging. Usually, they provide detailed descriptions for the peer TPDs to understand the problems they encounter. Besides programming suggestions, TPDs expressed the need to have a proper tool kit to actualize certain design ideas: “*With a proper SDK I can try some of the ideas I had a while back for working with Blackmagic switchers/ATEM, in conjunction with our VT setup (either VMix or CasparCG depending on the situation).*”

There are a few roles TPDs want to seek support from. Usually, people ask the whole Discord server questions and appreciate any help. Besides asking questions in the Discord server, people could also look up official posts or consult commercial third-party agencies specializing in developing Twitch-related programs. However, they sometimes need help from specific roles, such as Twitch engineers, Twitch extension review staff, and people with legal expertise.

In general, TPDs are willing to help their peers. TPDs respond to support requests in various ways. For example, when there are already well-written or well-organized external resources suitable for answering the questions, TPDs redirect people to these, such as websites, blogs, or commercial third-party supportive agencies. TPDs also pointed out the malpractices or shared failure experiences to prevent peers from being trapped. Some TPDs provided instrumental support to benefit the TPD community. For instance, a TPD mentioned that they tried to build foundational programs that people could learn from and build upon. On the other hand, the Twitch DX team also actively looks for feedback from TPDs regarding Twitch UI, documentation, and products. However, there are some problems this Discord server could not help with, such as helping TPDs get a quicker response from Twitch staff.

**4.4.2 Socializing with other TPDs.** TPDs are glad that there is a Discord server for TPDs. Compared with the official Twitch developer forum, this Discord server serves as an interactive panel small enough that TPDs could feel included and talk more. TPDs here ask for or provide help and encourage their peers, share an appreciation for help, chat about their lives and leisure, update their developing status, and share personal thoughts. Furthermore, they share their experiences in Twitch meet-ups or Twitch-related events and discuss attending events, such as Twitch AWS Extension Challenge, Twitchcon, and extension workshops.

## 5 DISCUSSION

In this study, we redirected our attention from the tool user’s perspective and the scholarly focus on tools’ social and functional roles

to TPDs engaged in developing tools for community management. We highlighted the importance of TPDs as critical stakeholders and a distinct user group in community management. These findings can potentially guide future work related to novel platforms with a high demand for TPDs to supplement their platform ecosystems, such as TPDS for VR and generative AI platforms, as these platforms quickly evolve with advanced technological infrastructures. We confirmed that TPDs for tool development align with independent developers in other domains, such as gaming [29]. This consistency underscores how the emerging open and participatory production model is instrumental in forming new workforces and fostering innovative technological practices within the worldwide technology industry [23]. While the ecosystems with open-source development models foster innovation and creativity, the tensions and challenges regarding policy and power with platforms and different stakeholders restrict TPDs' agency to actualize innovation and creation.

### 5.1 Co-Creation for Idea Innovation with Tool Users

Prior work suggests that regular meetings with end-users helped to keep the software developer aware of the usability without having to devote extensive periods to it [70]. Engaging with streamers and viewers allows TPDs to gain insights into user needs, pain points, and preferences, resulting in tools that align more closely with user expectations. This collaborative approach can lead to innovative solutions that might not have been possible through TPDs' efforts alone. The Discord community mediates co-creation, idea innovation, and support with tool users, motivating TPDs' autonomy to develop practical tools to benefit the ecosystem.

The Discord community is a space that Twitch staff, Twitch users, and TPDs are all familiar with, as it inherits the community of Twitch streamers and their fans, and some TPDs are also streamers. It serves as a third space for stakeholder gathering. This space allows for exchanging ideas, feedback, and innovation, leading to improved tool development and enhanced engagement with the user community, as shown in prior work [46]. Discord's platform facilitates direct and real-time communication (unlike thread-like comments on Twitter and Reddit, more like chronological chat with optional voice communication [42]), enabling TPDs to stay connected with users and receive continuous input. In this sense, Discord, as a space, lowers the barrier for diverse stakeholders, facilitates co-creation, and results in a win-win situation for multiple stakeholders. This indicates the importance of considering not just the front-end users and the core platform in understanding the ecosystem's landscape but also those supporting the platform and other spaces facilitating interactions among stakeholders behind the scenes.

The legal and privacy concerns impose restrictions on obtaining tool users' data, limiting their ability to understand users' needs and impeding the actualization of these designs. Prior work shows that developers might request multiple permissions because of the confusion of permission scope and policy requirement [88]. The data permission and legal regulation intertwined with Twitch platform policy confuse TPDs and sometimes even cause violations and ethical issues of tool development. How to balance offering

user data permissions to stimulate design innovation and protecting users' privacy requires further investigation.

### 5.2 Twitch Infrastructure Update and Policy Dynamics: Conflict of Control

The relationship between TPDs and Twitch itself, particularly regarding technical updates, policy changes, and design rules, can create complexities in TPDs' efforts. Like independent developers who often lack the financial, social, and technical support to sustain their ongoing labor and production [29], TPDs lack socio-technical and political support from the platform, restricting their autonomy. These updates can directly impact their tool performance and their psychological well-being.

Platform's function and interface update can significantly broaden the development knowledge gap between the platform and third parties and increase TPDs' barriers to understanding and aligning their development and innovation with the platform's vision [24]. TPDs must quickly adapt their tools to frequent technical updates and evolving policies on Twitch. Evolving policies can impose a significant administrative burden, necessitating additional labor to stay informed and prevent operational disruptions. TPDs must continually monitor and adjust to these policy changes to maintain compliance. The lack of clarity in these policies leads to rumors and confusion among TPDs, impacting their decision-making and, potentially, their financial stability.

As a leading live streaming platform in this industry, Twitch can do whatever it wants, such as keep updating policies and infrastructure, causing much labor and confusion to the TPDs, who are valuable stakeholders of the platform ecosystem. Dangerously, many TPDs might act against the policy by circumventing the policy regulation when they feel it is unfair and might harm their communities. This could potentially deteriorate the ecosystem and drive TPDs away if they can not get their bot approved in time (e.g., they had tried many times to keep updating their bots), which takes more physical and physiological labor. Furthermore, clarifying these policies and disentangling them to simplify policy presentation while ensuring they are up-to-date and transparent remains challenging. This is especially true given the rapid pace of development and technological advancements in the industry, warranting further investigation.

Prior work has shown the platform crafted the boundary resources (the tools and regulations used to maintain the relationship between the platform and TPDs) for cultivating the platform ecosystem [33]. While Twitch provides the technical resources to empower TPDs' agency, it also provides multi-layered policies to regulate TPDs' autonomy. In our case, policies might have a more substantial impact than the technical resources, reflecting the platform's conflict of control. On the one hand, it hopes to embrace and help the platform's ecosystem flourish with open API and tool kits; on the other hand, it reinforces multi-layered policies at different development stages, from developing, reviewing, and deploying, to updating their tools. While TPDs use their boundary resources to build complementary applications, the development is a knowledge-intensive task that crosses organizational boundaries [43].

Building on previous research in HCI focused on technological design to enhance developers' skills and productivity (e.g., [34, 54, 55, 64]), we propose that future studies should incorporate an understanding of policy dynamics that influence TPDs' development capabilities within the socio-technical system design. This approach would integrate policy into the technological framework, providing a more holistic view of the factors that impact TPDs' effectiveness and efficiency.

We clarify that we don't want to criticize the platform, as an evolving and emerging technology [35], it shows its effort to improve its systems for its stakeholders, such as developing new tools, enriching existing tool features, and working on transparency report [65]. Updating and changing are good ways to improve the platform's user experience. However, it might cause harm to the ecosystem (e.g., TPDs felt they were not valued with all the barriers they encountered). Future work should explore how to improve the ecosystem's health with effective governance mechanisms [25] such as mitigating potential policy violations and balancing the empowerment and control, and ensuring that all stakeholders are synced and avoiding the extra labor in communication, coordination, and development.

### 5.3 TPDs and Community Governance

Different from independent game developers who often work as a small team [29] and can publish products in alternative ways such as self-funding and small studios [28], TPDs primarily work as individuals for tool development and have to rely on the platforms' governance politically and technically. TPDs take the semi-private space to form a community, exchange and innovate design ideas, socialize with peers, and show information and instrumental support regarding the platform's policy, technical updates, and potential ways to circumvent the policies and their lessons. As a reflection of constructive suggestions, these supports maintain their peers in the community [38].

TPDs also build community guidelines to facilitate constructive interactions. However, it might also cause issues regarding interactions between different stakeholders. We explicitly see evidence that Twitch developers welcome interaction, whereas the Discord community rules explicitly mention that they should not be bothered without permission unless there is an emergency. Defining a "emergency" issue in the community is unknown. The community guidelines seem to deter TPDs and demotivate the Twitch developers to engage.

Twitch's vague and inconsistent policies and regulations also motivate TPDs to initialize changes regarding tool development, with the collaboration with other stakeholders (e.g., streamers) to improve their voice and power to negotiate with Twitch, a good indicator of external developers to unionize and restructure power dynamics [28, 74]. That is a good start, but they also faced many challenges regarding the coordination of the community and the structure and governance of their communities. It seems like there is a clear power structure in this community (admins, Twitch developers, TPDs, and tool users). A core concern is how to effectively govern online communities while not demotivating multiple stakeholders. How to maintain the community can also be complex. In this study, we only know they started the idea to work with a formal

community to unify the bot development for their community. We know little about the work on this process, which needs further inquiry to support their community governance, with questions like what structure helps them thrive.

### 5.4 Design Implications

While TPDs shared positive experiences regarding idea innovation and support from the community, they also faced challenges from the Discord community governance and Twitch's technical and policy updates. We advocate for preserving the positive aspects of user experience while mitigating the negatives. Therefore, we propose designs that sustain innovation on Discord and enhance idea actualization on Twitch.

*5.4.1 Support the Ecosystem by Facilitating Collaboration and Maintaining Idea Innovation on Discord.* As the Discord community is the hub for communication and collaboration among stakeholders, we propose design implications to maintain idea innovation and facilitate stakeholder collaboration.

*Maintain the co-creation and innovation among TPD and tool users.* Though we found that Discord, as a third space, provides opportunities for stakeholders to exchange and innovate design ideas and that TPDs and peer learning thought conflict are minimal, and information and instrument support are abundant, we also identified some issues that would improve the community innovation if these are fixed, e.g., to integrate the Twitch forum and r/reddit into the communities. Prior work shows that some TPDs actively interact with streamers in the Twitch subreddit (r/Twitch) to understand their needs and challenges of their community management and moderation and develop a bot for them [7, 36]. Prior work shows developers leverage API to develop personalized bots for their core platform for collaborative purposes [57]. Similarly, TPDs and Discord designers might consider utilizing the Discord API to create bots that facilitate community maintenance, addressing specific needs identified by TPDs, such as identifying and summarizing the design ideas from various discussion [102] of tool users (e.g., streamers, moderators, viewers). Such an initiative would allow TPDs to synchronize all relevant information within the Twitch ecosystem, potentially leading to more effective co-creation for community management.

*Facilitate TPDs and Twitch developer collaboration.* Twitch developers might work as an oversight board but don't engage much about Twitch's most recent updates. The ineffective communication provides little support to TPDs. Prior work suggests that professional developers should maintain control and flexibility when interacting with open-source project developers [95]. We don't know whether Twitch developers face similar challenges when interacting with TPDs because of the platform policy and terms of service. However, we can still argue that the Twitch developers in the community should be more flexible and engaging under the Twitch terms of service. Improving communication and collaboration channels between TPDs and Twitch developers is crucial to address these challenges. This could involve creating more direct lines of communication and providing more explicit guidance on policy and technical changes. For example, the Discord community can leverage the social roles of bots to integrate a voting system

with a notification mechanism to remind a certain role in the community (e.g., Twitch staff) to intervene; such a way can also avoid the disturbing effect of frequent messaging from the community. When an idea is voted, it will stick to the top of the channel; when the vote achieves a certain number (e.g., 20 or 40), the system will send a reminder to all Twitch developers. Then, someone can intervene with a message or voice chat to clarify everything. This is a more socialized design with multi-modal interaction and real-time feedback within the TPDs' community, different from the Twitch UserVoice voting system, which is more centralized and static.

*5.4.2 Unify the Development Tools and Policies to Clarify the Development Workflow on Twitch.* According to the boundary resource model [33], Twitch might overly control the development process with technical and policy updates, restricting TPDs' autonomy; thus, the platform should provide more resources to empower TPD's agency and facilitate the idea actualization. Establishing mechanisms for TPDs to provide feedback on Twitch updates and policies can help bridge the gap between the platform and TPDs.

*Unify the toolkit for development.* Though Twitch has a tool kit for TPDs, many TPDs share their development practices with different programming tools. They faced debugging challenges, meaning that TPDs might need an enriched and unified tool kit to facilitate their development. Therefore, we suggest that the platform should expand and standardize the existing toolkits, incorporating feedback from the TPDs, as supported by Terry et al.'s research on the importance of collective input [90]. One possible approach is for the platform to create a matrix categorizing tool sections guided by bot/extension design principles and user needs identified in our research. This matrix would allow TPDs to contribute tools they have found helpful in development. Other TPDs could then vote on or add to these suggestions, ultimately leading to a consensus on a set of tools for each category. The finalized toolkit, enriched by this collaborative process, could then be integrated into the platform ecosystem, complete with relevant resources to support TPDs more effectively. However, the scale of the tool sections and the broad policy alignment among these tools might cause extra challenges for TPDs, warranting further exploration.

*Clarify the policy kit from development.* The inconsistent and opaque platform policy compounding legal issues limited TPDs' development autonomy. Though Twitch provides more than technical documentation such as blog posts and online communities (e.g., Twitch UserVoice), these are designated as self-service and have the potential to scale up like helpdesks [24, 44]. Prior work has shown that policy development can be categorized into a hierarchy to govern community [15]. We propose a policy kit to hierarchically structure the rules to guide novice and experienced TPDs. The policy tool kit provides professional developers and TPDs access to update relevant parts with one or two administrators to approve or decline the edit, similar to the crowd documentation [68], but in a structured format about policy and other necessary documentation, with the involvement of the direct stakeholders to articulate the procedure of governance [103].

## 5.5 Limitations and Future Work

There are several limitations. First, this is a preliminary study of TPDs' tool development. However, we only looked at the data in the main channel ("Lobby"). Other channels might have a more in-depth discussion of a specific category. Future work should extend this work and answer questions like how the TPDs build a network offline and how offline interaction affects their collaboration and innovation in the online space. Second, the qualitative study reveals the categories, yet we don't know how the events affect the interaction dynamics in this community. Future work can extend this study by running quantitative analysis to understand the interaction dynamics with community growth. Third, the findings mainly showed the policy and stakeholders' impact on tool development. We only have limited evidence about the tool development process. Future work can run qualitative studies, such as interviewing and observing TPDs in this community, to understand their design in-depth, such as the mental model of tool development. Fourth, our study is limited to Twitch's TPDs; however, TPDs across various live streaming platforms may operate within distinct hierarchies and political cultures, such as those in China. Future research should broaden its scope to include live streaming platforms across diverse contexts, aiming for a more comprehensive understanding of TPDs' practices. Lastly, the topic modeling approach in our study, focusing on two selected topics, may overlook nuanced details of other aspects. Future studies should employ more advanced techniques for topic filtration to fully capture TPDs' practices.

## 6 CONCLUSION

In this study, we delve into the practices and challenges faced by third-party developers (TPDs) within the Twitch live streaming platform, who play a vital role in mediating community management with tool development. Analyzing discussions in a semi-private TPD community on Discord, the study reveals that TPDs contend with intricate policy regulations that influence their tool development. Additionally, they serve as intermediaries between diverse stakeholders, including streamers, viewers, the platform, and professional developers. Their ability to manage these complex relationships and policies is crucial to strengthening their autonomy and innovation capabilities and actualizing tool development. This study contributes to the existing body of research on online community management by offering a unique perspective from the TPD's point of view. We underscore the importance of supporting their autonomy and agency in the development process. By recognizing the challenges and complexities they face, we can work towards creating a more effective and user-friendly ecosystem for online community management.

## REFERENCES

- [1] 2012. The Challenges of Dealing with Newcomers. In *Building Successful Online Communities*. The MIT Press. <https://doi.org/10.7551/mitpress/8472.003.0006>
- [2] Ava Bartolome and Shuo Niu. 2023. A Literature Review of Video-Sharing Platform Research in HCL. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–20. <https://doi.org/10.1145/3544548.3581107>
- [3] David M. Blei. 2012. Probabilistic Topic Models. *Commun. ACM* 55, 4 (apr 2012), 77–84. <https://doi.org/10.1145/2133806.2133826>
- [4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

- [5] Kevin J. Boudreau. 2012. Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation. *Organization Science* 23, 5 (Oct. 2012), 1409–1427. <https://doi.org/10.1287/orsc.1110.0678>
- [6] Kevin J. Boudreau and Karim R. Lakhani. 2009. How to Manage Outside Innovation. *MIT Sloan Management Review* (July 2009). <https://sloanreview.mit.edu/article/how-to-manage-outside-innovation/>
- [7] Jie Cai, Sagnik Chowdhury, Hongyang Zhou, and Donghee Yvette Wohn. 2023. Hate Raids on Twitch: Understanding Real-Time Human-Bot Coordinated Attacks in Live Streaming Communities. *Proceedings of the ACM on Human-Computer Interaction* CSCW (2023), 1–28.
- [8] Jie Cai and Donghee Y. Wohn. 2019. Categorizing Live Streaming Moderation Tools: An Analysis of Twitch. *International Journal of Interactive Communication Systems and Technologies* 9, 2 (July 2019), 36–50. <https://doi.org/10.4018/IJICST.2019070103>
- [9] Jie Cai and Donghee Yvette Wohn. 2021. After Violation But Before Sanction: Understanding Volunteer Moderators' Profiling Processes Toward Violators in Live Streaming Communities. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (Oct. 2021), 1–25. <https://doi.org/10.1145/3479554> Publisher: Association for Computing Machinery.
- [10] Jie Cai and Donghee Yvette Wohn. 2022. Coordination and Collaboration: How do Volunteer Moderators Work as a Team in Live Streaming Communities?. In *CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–14. <https://doi.org/10.1145/3491102.3517628>
- [11] Jie Cai, Donghee Y. Wohn, and Masha'al Almoqbel. 2021. Moderation Visibility: Mapping the Strategies of Volunteer Moderators in Live Streaming Micro Communities. In *Proceedings of ACM International Conference on Interactive Media Experiences*. 61–72. <https://doi.org/10.1145/3452918.3458796>
- [12] Jie Cai and Donghee Yvette Wohn. 2023. Understanding Moderators' Conflict and Conflict Management Strategies with Streamers in Live Streaming Communities; Understanding Moderators' Conflict and Conflict Management Strategies with Streamers in Live Streaming Communities. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. ACM, Hamburg, Germany, 1–12. <https://doi.org/10.1145/3544548.3580982>
- [13] Stevie Chancellor, Andrea Hu, and Munmun De Choudhury. 2018. Norms matter: Contrasting social support around behavior change in online weight loss communities. *Conference on Human Factors in Computing Systems - Proceedings 2018-April* (April 2018). <https://doi.org/10.1145/3173574.3174240> ISBN: 9781450356206 Publisher: Association for Computing Machinery.
- [14] Eshwar Chandrasekharan, Chaitrali Gandhi, Matthew Wortley Mustelier, and Eric Gilbert. 2019. Crossmod: A Cross-Community Learning-based System to Assist Reddit Moderators. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (2019), 1–30. <https://doi.org/10.1145/3359276>
- [15] Eshwar Chandrasekharan, Mattia Samory, Shagun Jhaver, Hunter Charvat, Amy Bruckman, Cliff Lampe, Jacob Eisenstein, and Eric Gilbert. 2018. The Internet's Hidden Rules: An Empirical Study of Reddit Norm Violations at Micro, Meso, and Macro Scales. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (Nov. 2018), 32:1–32:25. <https://doi.org/10.1145/3274301>
- [16] Eshwar Chandrasekharan, Mattia Samory, Anirudh Srinivasan, and Eric Gilbert. 2017. The bag of communities: Identifying abusive behavior online with preexisting internet data. *Conference on Human Factors in Computing Systems - Proceedings 2017-May* (May 2017), 3175–3187. <https://doi.org/10.1145/3025453.3026018> ISBN: 9781450346559 Publisher: Association for Computing Machinery.
- [17] Anubrata Das, Brandon Dang, and Matthew Lease. 2020. Fast, Accurate, and Healthier: Interactive Blurring Helps Moderators Reduce Exposure to Harmful Content. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 8, 1 (2020), 33–42. <https://ojs.aaai.org/index.php/HCOMP/article/view/7461>
- [18] Jessica T. DeCuir-Gunby, Patricia L. Marshall, and Allison W. McCulloch. 2011. Developing and Using a Codebook for the Analysis of Interview Data: An Example from a Professional Development Research Project. *Field Methods* 23, 2 (May 2011), 136–155. <https://doi.org/10.1177/1525822X10388468>
- [19] Bryan Dosono and Bryan Semaan. 2019. Moderation Practices as Emotional Labor in Sustaining Online Communities. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*. 1–13. <https://doi.org/10.1145/3290605.3300372>
- [20] Ian Drossos and Philip J Guo. 2022. The Design Space of Livestreaming Equipment Setups: Tradeoffs, Challenges, and Opportunities. In *Designing Interactive Systems Conference*. ACM, New York, NY, USA, 835–848. <https://doi.org/10.1145/3532106.3533489>
- [21] Jennifer Fereday and Eimear Muir-Cochrane. 2006. Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development. *International Journal of Qualitative Methods* 5, 1 (March 2006), 80–92. <https://doi.org/10.1177/160940690600500107>
- [22] Anna Filippova and Hichang Cho. 2015. Mudslinging and manners: Unpacking conflict in free and open source software. In *Proceedings of the 2015 ACM International Conference on Computer-Supported Cooperative Work and Social Computing*. 1393–1403. <https://doi.org/10.1145/2675133.2675254>
- [23] Jörg Flecker and Pamela Meil. 2010. Organisational restructuring and emerging service value chains: implications for work and employment. *Work, employment and society* 24, 4 (2010), 680–698. <https://doi.org/10.1177/0950017010380635>
- [24] Jens Foerderer, Thomas Kude, Sebastian Walter Schuetz, and Armin Heinzl. 2019. Knowledge boundaries in enterprise software platform development: Antecedents and consequences for platform governance. *Information Systems Journal* 29, 1 (2019), 119–144. <https://doi.org/10.1111/ijisj.12186> \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/ijisj.12186>.
- [25] Awdren Fontão, Bruno Ábia, Igor Wiese, Bernardo Estácio, Marcelo Quinta, Rodrigo Pereira dos Santos, and Arilo Claudio Dias-Neto. 2018. Supporting governance of mobile application developers from mining and analyzing technical questions in stack overflow. *Journal of Software Engineering Research and Development* 6, 1 (Aug. 2018), 8. <https://doi.org/10.1186/s40411-018-0052-6>
- [26] aline shakti franzke, Anja Bechmann, Zimmer Michael, Charles Ess, and the Association of Internet Researchers. 2020. Internet Research: Ethical Guidelines 3.0. <https://aoir.org/reports/ethics3.pdf>
- [27] Guo Freeman. 2016. Making Games as Collaborative Social Experiences: Exploring An Online Gaming Community. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion (CSCW '16 Companion)*. Association for Computing Machinery, New York, NY, USA, 265–268. <https://doi.org/10.1145/2818052.2869076>
- [28] Guo Freeman, Jeffrey Bardzell, Shaowen Bardzell, and Nathan McNeese. 2020. Mitigating Exploitation: Indie Game Developers' Reconfigurations of Labor in Technology. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW1 (May 2020), 56:1–56:23. <https://doi.org/10.1145/3392864>
- [29] Guo Freeman, Lingyuan Li, Nathan McNeese, and Kelsea Schultenberger. 2023. Understanding and Mitigating Challenges for Non-Profit Driven Indie Game Development to Innovate Game Production. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–16. <https://doi.org/10.1145/3544548.3580976>
- [30] Guo Freeman, Nathan McNeese, Jeffrey Bardzell, and Shaowen Bardzell. 2020. "Pro-Amateur"-Driven Technological Innovation: Participation and Challenges in Indie Game Development. *Proceedings of the ACM on Human-Computer Interaction* 4, GROUP (Jan. 2020), 04:1–04:22. <https://doi.org/10.1145/3375184>
- [31] Guo Freeman and Nathan J. McNeese. 2019. Exploring Indie Game Development: Team Practices and Social Experiences in A Creativity-Centric Technology Community. *Computer Supported Cooperative Work* 28, 3-4 (June 2019), 723–748. <https://doi.org/10.1007/s10606-019-09348-x>
- [32] Guo Freeman and Nathan J. McNeese. 2021. A Tale of Creativity and Struggles: Team Practices for Bottom-Up Innovation in Virtual Game Jams. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (April 2021), 76:1–76:27. <https://doi.org/10.1145/3449150>
- [33] Ahmad Ghazawneh and Ola Henfridsson. 2013. Balancing platform control and external contribution in third-party development: the boundary resources model. *Information Systems Journal* 23, 2 (March 2013), 173–192. <https://doi.org/10.1111/J.1365-2575.2012.00406.X> Publisher: John Wiley & Sons, Ltd.
- [34] Anja Guzzi, Alberto Bacchelli, Yann Riche, and Arie van Deursen. 2015. Supporting Developers' Coordination in the IDE. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, New York, NY, USA, 518–532. <https://doi.org/10.1145/2675133.2675177>
- [35] William A. Hamilton, Oliver Garretson, and Android Kerne. 2014. Streaming on twitch: Fostering participatory communities of play within live mixed media. In *Conference on Human Factors in Computing Systems - Proceedings*. ACM, New York, NY, USA, 1315–1324. <https://doi.org/10.1145/2556288.2557048>
- [36] Catherine Han, Joseph Seering, Deepak Kumar, Jeffrey T. Hancock, and Zakir Durumeric. 2023. Hate Raids on Twitch: Echoes of the Past, New Modalities, and Implications for Platform Governance. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (April 2023), 1–28. <https://doi.org/10.1145/3579609> Publisher: ACM PUB27 New York, NY, USA.
- [37] Jane Hsieh, Joselyn Kim, Laura Dabbish, and Haiyi Zhu. 2023. "Nip it in the Bud": Moderation Strategies in Open Source Software Projects and the Role of Bots. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW2 (Oct. 2023), 301:1–301:29. <https://doi.org/10.1145/3610092>
- [38] Wenjian Huang, Tun Lu, Haiyi Zhu, Guo Li, and Ning Gu. 2016. Effectiveness of Conflict Management Strategies in Peer Review Process of Online Collaboration Projects. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*. Association for Computing Machinery, New York, NY, USA, 717–728. <https://doi.org/10.1145/2818048.2819950>
- [39] Shagun Jhaver, Darren Scott Appling, Eric Gilbert, and Amy Bruckman. 2019. "Did you suspect the post would be removed?": Understanding user reactions to content removals on reddit. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (Nov. 2019), 1–33. <https://doi.org/10.1145/3359294> Publisher: Association for Computing Machinery.
- [40] Shagun Jhaver, Iris Birman, Eric Gilbert, and Amy Bruckman. 2019. Human-machine collaboration for content regulation: The case of reddit automoderator. *ACM Transactions on Computer-Human Interaction* 26, 5 (2019), 35. <https://doi.org/10.1145/3338243> ISBN: 10.1145/3338243.

- [41] Shagun Jhaver, Quan Ze Chen, Detlef Knauss, and Amy X. Zhang. 2022. Designing Word Filter Tools for Creator-led Comment Moderation. In *CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–21. <https://doi.org/10.1145/3491102.3517505> \_eprint: 2202.08818.
- [42] Jialun Jiang, Charles Kiene, Skyler Middler, Jed R. Brubaker, and Casey Fiesler. 2019. Moderation challenges in voice-based online communities on discord. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (Nov. 2019). <https://doi.org/10.1145/3359157> Publisher: Association for Computing Machinery \_eprint: 2101.05258.
- [43] Martin Kauschinger and Maximilian Schreieck. 2023. Detecting Feature Requests of Third-Party Developers through Machine Learning: A Case Study of the SAP Community. In *Proceedings of the 56th Hawaii International Conference on System Sciences*. 10. <https://doi.org/10.125/102747>
- [44] Martin Kauschinger, Maximilian Schreieck, Markus Boehm, and Helmut Krcmar. 2021. Knowledge Sharing in Digital Platform Ecosystems – A Textual Analysis of SAP’s Developer Community. In *Innovation Through Information Systems (Lecture Notes in Information Systems and Organisation)*, Frederik Ahlemann, Reinhard Schütte, and Stefan Stieglitz (Eds.). Springer International Publishing, Cham, 21–39. [https://doi.org/10.1007/978-3-030-86797-3\\_2](https://doi.org/10.1007/978-3-030-86797-3_2)
- [45] Charles Kiene, Kate Grandprey-Shores, Eshwar Chandrasekharan, Shagun Jhaver, Jialun “Aaron” Jiang, Brianna Dym, Joseph Seering, Sarah Gilbert, Kat Lo, Donghee Yvette Wohn, and Bryan Dosono. 2019. Volunteer Work: Mapping the Future of Moderation Research. In *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing (CSCW ’19)*. Association for Computing Machinery, New York, NY, USA, 492–497. <https://doi.org/10.1145/3311957.3359443>
- [46] Charles Kiene, Jialun Aaron Jiang, and Benjamin Mako Hill. 2019. Technological frames and user innovation: Exploring technological change in community moderation teams. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (Nov. 2019), 1–23. <https://doi.org/10.1145/3359146> Publisher: Association for Computing Machinery.
- [47] Soomin Kim, Jinsu Eun, Changhoon Oh, Bongwon Suh, and Joonhwan Lee. 2020. Bot in the Bunch: Facilitating Group Chat Discussion by Improving Efficiency and Participation with a Chatbot. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI ’20)*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376785>
- [48] Soomin Kim, Jinsu Eun, Joseph Seering, and Joonhwan Lee. 2021. Moderator Chatbot for Deliberative Discussion: Effects of Discussion Structure and Dissuasive Facilitation. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (April 2021), 1–26. <https://doi.org/10.1145/3449161>
- [49] Yubo Kou. 2021. Punishment and Its Discontents: An Analysis of Permanent Ban in an Online Game Community. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (Oct. 2021), 1–21. <https://doi.org/10.1145/3476075> Publisher: Association for Computing Machinery.
- [50] Yubo Kou and Xinning Gui. 2020. Mediating Community-AI Interaction through Situated Explanation: The Case of AI-Led Moderation. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 27. <https://doi.org/10.1145/3415173> ISBN: 10.1145/3415173.
- [51] Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*. SAGE. Google-Books-ID: q657o3M3C8cC.
- [52] Srijan Kumar, William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2018. Community Interaction and Conflict on the Web. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. ACM Press, New York, New York, USA, 933–943. <https://doi.org/10.1145/3178876.3186141> \_eprint: 1803.03697.
- [53] Tina Kuo, Alicia Hernani, and Jens Grossklags. 2023. The Unsung Heroes of Facebook Groups Moderation: A Case Study of Moderation Practices and Tools. *Proceedings of the ACM on Human-Computer Interaction* 7, CSCW1 (April 2023), 97:1–97:38. <https://doi.org/10.1145/3579530>
- [54] Lingyuan Li, Guo Freeman, and Nathan J. McNeese. 2022. Channeling End-User Creativity: Leveraging Live Streaming for Distributed Collaboration in Indie Game Development. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (Nov. 2022), 282:1–282:28. <https://doi.org/10.1145/3555173>
- [55] Ziming Li, Kristen Shinohara, and Roshan L Peiris. 2023. Exploring the Use of the SoundVizVR Plugin with Game Developers in the Development of Sound-Accessible Virtual Reality Games. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA ’23)*. Association for Computing Machinery, New York, NY, USA, 1–7. <https://doi.org/10.1145/3544549.3585750>
- [56] S Likhitha, BS Harish, and HM Keerthi Kumar. 2019. A detailed survey on topic modeling for document and short text data. *International Journal of Computer Applications* 178, 39 (2019), 1–9. <https://doi.org/10.5120/ijca2019919265>
- [57] Bin Lin, Alexey Zagalsky, Margaret-Anne Storey, and Alexander Serebrenik. 2016. Why Developers Are Slacking Off: Understanding How Software Teams Use Slack. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion*. ACM, New York, NY, USA, 333–336. <https://doi.org/10.1145/2818052.2869117>
- [58] Yao Lyu, Jie Cai, Anisa Callis, Kelley Cotter, and John M. Carroll. 2024. “I Got Flagged for Supposed Bullying, Even Though It Was in Response to Someone Harassing Me About My Disability”: A Study of Blind TikTokers’ Content Moderation Experiences. <https://doi.org/10.1145/3613904.3642148> arXiv:2401.11663 [cs].
- [59] Renkai Ma and Yubo Kou. 2021. “How advertiser-friendly is my video?”: YouTube’s Socioeconomic Interactions with Algorithmic Content Moderation. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (Oct. 2021), 1–25. <https://doi.org/10.1145/3479573> Publisher: ACM.
- [60] Regan L Mandryk, Julian Frommel, Nitesh Goyal, Guo Freeman, Cliff Lampe, Sarah Vieweg, and Donghee Yvette Wohn. 2023. Combating Toxicity, Harassment, and Abuse in Online Social Spaces: A Workshop at CHI 2023. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA ’23)*. Association for Computing Machinery, New York, NY, USA, 1–7. <https://doi.org/10.1145/3544549.3573793>
- [61] Enrico Mariconti, Guillermo Suarez-Tangil, Jeremy Blackburn, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Jordi Luque Serrano, and Gianluca Stringhini. 2019. “You Know What to Do”: Proactive Detection of YouTube Videos Targeted by Coordinated Hate Attacks. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (Nov. 2019), 1–21. <https://doi.org/10.1145/3359309> \_eprint: 1805.08168.
- [62] Market.us. 2023. API Management Market to Reach Valuation of USD 49.9 Billion at Cagr of 28% by 2032 - Report by Market.us. <https://www.globenewswire.com/en/news-release/2023/03/21/2631010/0/en/API-Management-Market-to-Reach-Valuation-of-USD-49-9-Billion-at-Cagr-of-28-by-2032-Report-by-Market-us.html>
- [63] Nolwenn Maudet, Germán Leiva, Michel Beaudouin-Lafon, and Wendy Mackay. 2017. Design Breakdowns: Designer-Developer Gaps in Representing and Interpreting Interactive Systems. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, New York, NY, USA, 630–641. <https://doi.org/10.1145/2998181.2998190>
- [64] 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 (CSCW ’12)*. Association for Computing Machinery, New York, NY, USA, 1175–1178. <https://doi.org/10.1145/2145204.2145377>
- [65] Ed Nightingale. 2022. Twitch streamers positive about new raid feature in response to harassment. <https://www.eurogamer.net/twitch-streamers-positive-about-new-raid-feature-in-response-to-harassment> Publication Title: Eurogamer.
- [66] Shirin Nilizadeh, François Labrèche, Alireza Sedighian, Ali Zand, José Fernandez, Christopher Kruegel, Gianluca Stringhini, and Giovanni Vigna. 2017. POISED: Spotting twitter spam off the beaten paths. In *Proceedings of the ACM Conference on Computer and Communications Security*. ACM, New York, NY, USA, 1159–1174. <https://doi.org/10.1145/3133956.3134055> ISSN: 15437221 \_eprint: 1708.09058.
- [67] Shuo Niu, Zhicong Lu, Amy X. Zhang, Jie Cai, Carla F. Griggio, and Hendrik Heuer. 2023. Building Credibility, Trust, and Safety on Video-Sharing Platforms. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA ’23)*. Association for Computing Machinery, New York, NY, USA, 1–7. <https://doi.org/10.1145/3544549.3573809>
- [68] Chris Parnin and Christoph Treude. 2011. Measuring API documentation on the web. In *Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering*. ACM, Waikiki, Honolulu HI USA, 25–30. <https://doi.org/10.1145/1984701.1984706>
- [69] Xuan-Hieu Phan, Cam-Tu Nguyen, Dieu-Thu Le, Le-Minh Nguyen, Susumu Horiguchi, and Quang-Thuy Ha. 2011. A Hidden Topic-Based Framework toward Building Applications with Short Web Documents. *IEEE Transactions on Knowledge and Data Engineering* 23, 7 (2011), 961–976. <https://doi.org/10.1109/TKDE.2010.27>
- [70] Suzanne Prior, Annalu Waller, Rolf Black, and Thilo Kroll. 2013. Use of an agile bridge in the development of assistive technology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Paris France, 1579–1588. <https://doi.org/10.1145/2470654.2466210>
- [71] Reinhard Prügl and Martin Schreier. 2006. Learning from leading-edge customers at The Sims: opening up the innovation process using toolkits. *R&D Management* 36, 3 (June 2006), 237–250. <https://doi.org/10.1111/J.1467-9310.2006.00433.X> Publisher: John Wiley & Sons, Ltd.
- [72] Jiali Qi and Aron Lindberg. 2021. Understanding the Reactive Practices of Third-party Developers. *Academy of Management Proceedings* 2021, 1 (Aug. 2021), 12641. <https://doi.org/10.5465/AMBPP.2021.12641abstract> Publisher: Academy of Management Briarcliff Manor, NY 10510.
- [73] Rifat Rahman, Md. Rishadur Rahman, Nafis Irtiza Tripto, Mohammed Eunos Ali, Sajid Hasan Apon, and Rifat Shahriyar. 2021. AdolescentBot: Understanding Opportunities for Chatbots in Combating Adolescent Sexual and Reproductive Health Problems in Bangladesh. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI ’21)*. Association for Computing Machinery, New York, NY, USA, 1–15. <https://doi.org/10.1145/3411764.3445694>
- [74] Paolo Ruffino (Ed.). 2020. *Independent Videogames: Cultures, Networks, Techniques and Politics*. Routledge, London. <https://doi.org/10.4324/9780367336219>

- [75] Joseph Seering, Tianmi Fang, Luca Damasco, Mianhong Cherie Chen, Likang Sun, and Geoff Kaufman. 2019. Designing User Interface Elements to Improve the Quality and Civility of Discourse in Online Commenting Behaviors. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–14. <https://doi.org/10.1145/3290605.3300836>
- [76] Joseph Seering, Juan Pablo Flores, Saiph Savage, and Jessica Hammer. 2018. The Social Roles of Bots: Evaluating Impact of Bots on Discussions in Online Communities. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (Nov. 2018), 157:1–157:29. <https://doi.org/10.1145/3274426>
- [77] Joseph Seering, Robert Kraut, and Laura Dabbish. 2017. Shaping Pro and Anti-Social Behavior on Twitch Through Moderation and Example-Setting. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, New York, NY, USA, 111–125. <https://doi.org/10.1145/2998181.2998277>
- [78] Joseph Seering, Tony Wang, Jina Yoon, and Geoff Kaufman. 2019. Moderator engagement and community development in the age of algorithms. *New Media & Society* 21, 7 (July 2019), 1417–1443. <https://doi.org/10.1177/1461444818821316>
- [79] Robert W Service. 2009. Book Review: Corbin, J., & Strauss, A.(2008). Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Thousand Oaks, CA: Sage. *Organizational Research Methods* 12, 3 (2009), 614–617. <https://doi.org/10.1177/1094428108324514>
- [80] Jeff T Sheng and Sanjay R Kairam. 2020. From Virtual Strangers to IRL Friends: Relationship Development in Livestreaming Communities on Twitch. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 34. <https://doi.org/10.1145/3415165>
- [81] Carson Sievert and Kenneth Shirley. 2014. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the workshop on interactive language learning, visualization, and interfaces*. 63–70.
- [82] Vandana Singh, Aditya Johri, and Raktim Mitra. 2011. Types of newcomers in an online developer community. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work (CSCW '11)*. Association for Computing Machinery, New York, NY, USA, 717–720. <https://doi.org/10.1145/1958824.1958953>
- [83] C. Estelle Smith, Irfanul Alam, Chenhao Tan, Brian C. Keegan, and Anita L. Blanchard. 2022. The Impact of Governance Bots on Sense of Virtual Community: Development and Validation of the GOV-BOTs Scale. *Proceedings of the ACM on Human-Computer Interaction* 6, CSCW2 (Nov. 2022), 462:1–462:30. <https://doi.org/10.1145/3555563>
- [84] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring Topic Coherence over Many Models and Many Topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (Jeu Island, Korea) (EMNLP-CoNLL '12)*. Association for Computational Linguistics, USA, 952–961.
- [85] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*. 952–961.
- [86] Gianluca Stringhini, Pierre Moulanne, Gregoire Jacob, Manuel Egele, Christopher Kruegel, and Giovanni Vigna. 2015. Evilcohort: Detecting communities of malicious accounts on online services. <https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/stringhini> ISBN: 9781931971232 Pages: 563–578 Publication Title: Proceedings of the 24th USENIX Security Symposium.
- [87] Sharifa Sultana, Mitrasree Deb, Ananya Bhattacharjee, Shaid Hasan, S.M.Raihanul Alam, Trishna Chakraborty, Prianka Roy, Samira Fairuz Ahmed, Aparna Moitra, M. Ashrafu Amin, A.K.M. Najmul Islam, and Syed Ishtiaque Ahmed. 2021. 'Unmochon': A Tool to Combat Online Sexual Harassment over Facebook Messenger. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–18. <https://doi.org/10.1145/3411764.3445154>
- [88] Mohammad Tahaei, Ruba Abu-Salma, and Awais Rashid. 2023. Stuck in the Permissions With You: Developer & End-User Perspectives on App Permissions & Their Privacy Ramifications. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–24. <https://doi.org/10.1145/3544548.3581060>
- [89] David J. Teece. 1986. Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. *Research Policy* 15, 6 (Dec. 1986), 285–305. [https://doi.org/10.1016/0048-7333\(86\)90027-2](https://doi.org/10.1016/0048-7333(86)90027-2) Publisher: North-Holland.
- [90] Michael Terry, Matthew Kay, and Ben Lafreniere. 2010. Perceptions and practices of usability in the free/open source software (FoSS) community. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Atlanta Georgia USA, 999–1008. <https://doi.org/10.1145/1753326.1753476>
- [91] Yuen-Hsien Tseng, Chi-Jen Lin, and Yu-I Lin. 2007. Text mining techniques for patent analysis. *Information processing & management* 43, 5 (2007), 1216–1247.
- [92] Jirassaya Uttarapong, Jie Cai, and Donghee Yvette Wohn. 2021. Harassment Experiences of Women and LGBTQ Live Streamers and How They Handled Negativity. In *ACM International Conference on Interactive Media Experiences*. ACM, New York, NY, USA, 7–19. <https://doi.org/10.1145/3452918.3458794>
- [93] Ruben van Wendel de Joode. 2004. Managing Conflicts in Open Source Communities. *Electronic Markets* 14, 2 (2004), 104–113. <https://doi.org/10.1080/10196780410001675059>
- [94] Joel West and Michael Mace. 2010. Browsing as the killer app: Explaining the rapid success of Apple's iPhone. *Telecommunications Policy* 34, 5-6 (June 2010), 270–286. <https://doi.org/10.1016/j.telpol.2009.12.002> Publisher: Pergamon.
- [95] Juliane Wissel, Michael Zagg, and Aron Lindberg. 2019. How Companies Govern Their Open Source Software Contributions: A Case Study. In *Fortieth International Conference on Information Systems*. 10.
- [96] Donghee Yvette Wohn. 2019. Volunteer Moderators in Twitch Micro Communities. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 1–13. <https://doi.org/10.1145/3290605.3300390>
- [97] Donghee Yvette Wohn, Casey Fiesler, Libby Hemphill, Munmun De Choudhury, and J. Nathan Matias. 2017. How to handle online risks? Discussing content curation and moderation in social media. *Conference on Human Factors in Computing Systems - Proceedings Part F1276* (May 2017), 1271–1276. <https://doi.org/10.1145/3027063.3051141> ISBN: 9781450346566 Publisher: Association for Computing Machinery.
- [98] Donghee Yvette Wohn and Guo Freeman. 2020. Audience Management Practices of Live Streamers on Twitch. *IMX 2020 - Proceedings of the 2020 ACM International Conference on Interactive Media Experiences* (June 2020), 106–116. <https://doi.org/10.1145/3391614.3393653> ISBN: 9781450379762 Publisher: Association for Computing Machinery, Inc.
- [99] Austin P. Wright, Omar Shaikh, Haekyu Park, Will Epperson, Muhammed Ahmed, Stephane Pinel, Duen Horng (polo) Chau, and Diyi Yang. 2021. Recast: Enabling User Recourse and Interpretability of Toxicity Detection Models with Interactive Visualization. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (April 2021). <https://doi.org/10.1145/3449280> Publisher: Association for Computing Machinery\_eprint: 2102.04427.
- [100] Deepika Yadav, Prerna Malik, Kirti Dabas, and Pushpendra Singh. 2019. Feedpal: Understanding Opportunities for Chatbots in Breastfeeding Education of Women in India. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (Nov. 2019), 170:1–170:30. <https://doi.org/10.1145/3359272>
- [101] Ryan Yen, Li Feng, Brinda Mehra, Ching Christie Pang, Siying Hu, and Zhicong Lu. 2023. StoryChat: Designing a Narrative-Based Viewer Participation Tool for Live Streaming Chatrooms. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23)*. Association for Computing Machinery, New York, NY, USA, 1–18. <https://doi.org/10.1145/3544548.3580912>
- [102] Amy X. Zhang and Justin Cranshaw. 2018. Making Sense of Group Chat through Collaborative Tagging and Summarization. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (Nov. 2018), 196:1–196:27. <https://doi.org/10.1145/3274465>
- [103] Amy X. Zhang, Grant Hugh, and Michael S. Bernstein. 2020. PolicyKit: Building Governance in Online Communities. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 365–378. <https://doi.org/10.1145/3379337.3415858>
- [104] Lei (Nico) Zheng, Christopher M. Albano, Neev M. Vora, Feng Mai, and Jeffrey V. Nickerson. 2019. The Roles Bots Play in Wikipedia. *Proceedings of the ACM on Human-Computer Interaction* 3, CSCW (Nov. 2019), 215:1–215:20. <https://doi.org/10.1145/3359317>

## A THEMATIC ANALYSIS WITH CODEBOOK DEFINITION

## B FINDING SUMMARIZATION WITH DESCRIPTION



Theme	Sub-themes	Code	Definition
Design ideas exchange and innovation	Streamer's needs and design ideas	22	streamer can't work for niche features
		24	streamer's idea about designing fun VIP status
		35	streamer: whisper or email as messaging tool
		41	streamer's idea about the audible indicator feature
	Exchange design ideas	6	first extension showcase
		14	first extension design idea
33		first extension practice	
Review extension	49	general design extension idea	
	25	time of approval of extension	
Impact of Twitch technical update	Review extension	55	time of review of extension
		2	refresh before the expiration of the subbed topic
	Technical updates	8	refresh token
		9	change of version
		13	Twitch boilerplate with React is straightforward
		23	OBS plugin move source
		28	something like API that can be public
		37	twitch extension challenge
		50	extensions availability on different platforms
		52	multiple tool use is difficult
Policy and regulations at different levels	Platform policy update	32	Twitch evaluates bots
		34	extension design rule /consideration/explanation
	Data privacy and policy	60	relevant but not listed: privacy, security, authentication, permission
		39	redirect to a specific discord channel
	Discord community policy	45	TPD dispute
		47	No legal inquiry and advice
		57	community guideline
Bot design rules	34	extension design rule /consideration/explanation	
Seeking and showing support to different stakeholders	Complain and initiate unified rules	60	relevant but not listed: dev group action
		60	relevant but not listed: extension use
	Socializing with other TPDs	1	Extension workshop
		3	integration with the dev forum
	Information and instrumental support	4	happy to have discord
		38	ping me to help me
		40	help test extension
		42	Twitch DX: feedback on dev.twitch.tv
		43	personalized bot better than other bots.
		44	aware of coder error by comparing with others
48		web app for mods to modify bot	
51		TPD facilitates peer learning	
53	extension's request cost		
60	relevant but not listed: seek for collaborators		
The following are general information sharing and support (Q&A) that are common in all online communities and are not explicit about tool development. We just briefly mentioned the findings and social support section at the beginning.			
Twitch ID and data access Q&A		10	opaque id as key to store data
		11	ID share for data access
		15	chat logs record which ID
		46	replacement to get chat history
		5	Reasons to use rig
Testing and debugging Q&A		7	browse through the bot log and browser
		19	app start errors
		20	distro is not debian like a current kernel
		26	Linux for local Redis testing
		27	the error of creating a new app
		36	testing backend
		56	time to sort out bugs
		59	frontend backend error
Visual and audio design issues		16	transcode options for streaming quality
		21	red hat logo
		29	icons' resolutions
		30	transition/animation
		31	generate names of items
		54	tips for image testing
		58	raw message format change
Other		0	Irrelevant
		12	traffic between viewer and broadcaster
		17	stability is essential for money for streamers
		18	Internet provider

**Table 2: Themes, sub-themes, codes with definitions. The developed codebook contains a total of 61 codes (59 primary codes and two functional codes, where code 0 indicates 'irrelevant' and code 60 indicates 'relevant but not listed')**

Section	Subsection	Description
Design Ideas Exchange and Innovation		How Twitch Dev. Discord members innovate, share, and give feedback regarding Twitch extension design.
	Streamer's Need and Design Ideas	*Streamers seek advice on developing extensions that fulfill various streaming needs. *Streamers want extensions that provide rich interactivity to their audience or specialized extensions.
	Exchange Design Ideas among TPDs	*Categories of extension design ideas shared by TPDs. *TPDs share sandbox versions of bots and seek feedback for improvement. *Experience sharing and exchange of design ideas.
Impact of Twitch Technical update		The change of technical infrastructures and its impact on TPD's tool development, such as API version change.
	Schedule Uncertainty	*TPDs have many questions about uncertain review update time. *Unstable review time.
	Extra Labor Caused by Technical Updates	*Extra efforts are required to adapt technical update. *Technical update leads to potential extension ban. *TPDs' own ways to adapt technical updates. *Mixed opinions about Twitch's technical updates.
Policy and Regulations at Different Levels		The policies and rules that various regulators make and how these policies influence TPDs, such as guidelines about addressing user data or how Discord members interact with different channels.
	Platform Policy Update	*The vague or opaque regulations would confuse TPDs, and TPDs would adapt this confusion in various ways. *TPDs' confusion about the platform policy implementation and consequent reactions. *Opaque and unstable bot review rules and the consequences. *Misaligned policy regulation among Twitch and other platforms. *Confusions about the use and ownership of copyrights and intelligent property.
	Data Privacy and Policy	*User's permission for data access. *Reasons for data policy breach. *Access data without permission from Twitch users.
	Discord Community Policy	Expectations and rules.
	Bot/Extension Design Rules	*TPDs' confusions about Twitch's vague design patterns and principles. *Compatibility, dependency, and adaption on other platforms. *Risks and costs. *Users' perceptions and experiences.
	Complain and Initiate Policy and Platform Changes	*TPDs use Discord for their voluntarily consistent and coherent pattern of extension. *TPDs use Discord to unify their suggestions to Twitch.
	Seeking and Showing Support	
Information and Instrumental Support		*Seek community support due to a lack of official support. *Support needs from new extension developers. *Roles TPDs seek support from. *Ways TPDs respond to support requests.
Socializing with other TPDs		Discord as a smaller and more inclusive space for TPDs.

Table 3: Results summary