

Exploring the Use of Chatrooms by Developers: An Empirical Study on Slack and Gitter

Mariam El Mezouar, Daniel Alencar da Costa, Daniel M. German, and Ying Zou

Abstract—Communication is critical for the software development teams to maintain project awareness, facilitate project co-ordination and avoid misunderstandings. The features offered in the chatrooms, such as private messaging, group conversations, and code sharing help accommodate the communication needs of the software development teams. Therefore, chatrooms have been increasingly adopted among the developers. Since the last study on Slack performed by Lin *et al.* in 2016, the audience of Slack has more than doubled possibly leading to an evolution of the ways Slack is used; while another rich community formed around Gitter and remains unstudied. In this paper, we perform an investigative study using qualitative and quantitative techniques to gain insights on the use of popular modern chatrooms, specifically Slack and Gitter. Based on the survey responses from 163 developers, the interviews with 21 developers, and the chatroom data collected from 11 Slack and 770 Gitter rooms, we are able to uncover the reasons behind the use of Slack and Gitter, the perceived impact on the associated projects, and the quality determinants of the two chatrooms. We find that the developers seek knowledge from the chatrooms to obtain timely feedback from experts, and in return share their expertise to build the project community and their reputations. Furthermore, it is perceived by the Gitter developers that the chatrooms have an impact on prioritizing the new features and the bug fixes. In Slack, the most reported impact concerns an increased project awareness, in terms of a better tracking of the work progress. As reported on the developers' survey, both Slack and Gitter chat services have a visible impact on mentoring developers, and sharing the best practices. In terms of quality determinants, a non-ephemeral history and a better history management (e.g., advanced search) could be keys for both chat services to reach their full potential.

Index Terms—chatroom, Slack, Gitter, software development teams, communication

1 INTRODUCTION

In software development, the communication and coordination of software development teams are key factors for the success of the software projects [1] [2] [3]. Specifically, informal communication among developers promotes project awareness [4]. However, as software development teams grow in size, it becomes harder to prevent misunderstandings in the developers' interactions. Examples of such misunderstandings are confusions regarding the software requirements, uncertainty about the right developer to contact for questions, and ambiguity about the work progress. In an attempt to address such challenges, communication channels and social media are often integrated with the modern development tools [5] (e.g., chat, email, or microblogging services) to facilitate the communication between the developers [6].

Modern developer chatrooms are designed to fulfill the communication needs of the developers, including messaging, file management, code sharing, and video calling. Contrary to other communication channels, such as, the mailing lists or the Q&A platforms, Modern developer chatrooms bring developers together in an informal setting with equal participation opportunity for everyone. Modern developer chatrooms can be either "public" or "private".

- M. El Mezouar is with the School of Computing, Queen's University, Canada. E-mail: mariam@cs.queensu.ca
- D. da Costa is with the Department of Information Science, University of Otago, New Zealand. E-mail: danielcalencar@otago.ac.nz
- D. German is with the Department of Computer Science, University of Victoria, Canada. E-mail: dmg@uvic.ca
- Y. Zou is with the Department of Electrical and Computer Engineering, Queen's University, Canada. E-mail: ying.zou@queensu.ca

Public chatrooms are generally used by open source projects, and much emphasis is placed on the formation of a friendly community who talks and shares knowledge. For example, Gitter¹ is an open source chatroom geared towards the open communities. The particularity of Gitter is the possibility of a Gitter room to be associated to a GitHub repository. The private chatrooms are usually used by private software projects for team communication. Slack² is an example of a proprietary chatroom that has gained rapid uptake recently. It was initially designed for corporate teams, but is more and more adopted by the open communities as well. More details on the two subject systems follow in Section 2.

Research efforts have been invested to better understand the role of the chatrooms in software development. First, the chatrooms are considered important in supporting the informal communication among the developers [4]. In a large scale survey conducted by Storey *et al.* [6], private and public chatrooms were deemed as some of the most important channels by nearly 15 percent of the survey participants. Second, the chatrooms (in this case Internet Relay Chat) are used by developers for discussions of technical nature [7]. Lastly, the chatrooms (specifically Slack) are used for personal, team-wide, and community-wide purposes, based on a survey with 104 developers by Lin *et al.* [8].

Despite the valuable results obtained by prior research, modern developer chatrooms still need a more thorough investigation for the following reasons: 1) The adoption of

1. <https://gitter.im/>
2. <https://slack.com/about>

Slack since the most recent study by Lin *et al.* [8] in 2016 has more than doubled, from 3 million to over 8 million users. As such, the uses of Slack might have evolved over time; 2) another rich community of developers has been built around the usage of Gitter chatrooms and remains almost unstudied, which led recent research—such as the work from Parra *et al.* [9]—to collate useful data and foster more research on Gitter; 3) the inclusion of both Slack and Gitter in the study lets us compare two different types of chatrooms (e.g., proprietary and open source), and two different types of communities (e.g., the open communities in both Slack and Gitter, and the corporate teams mostly in Slack), and 4) beyond the uses of the chatrooms, the impact of using the chatrooms on the software projects, and the quality determinants of the chatrooms are still unclear.

We designed and conducted a survey with developers who adopt Gitter or Slack. Our survey received 163 responses. We analyzed the survey responses, using thematic analysis [10]. We further conducted follow-up interviews with 21 developers to gain more insights and validate our results. Furthermore, we mined the chat data archives from 770 Gitter rooms and 11 Slack rooms, and compute the chat activity metrics, to compare the perception of the developers with the reality. We organize our study into the following three research questions:

RQ1. *Why do developers use the chatrooms?* We investigate the reasons that motivate the developers to use the chatrooms to ask and answer questions. Access to experts and a fast response time are the main reasons as to why developers ask questions in the chatrooms. In return, the developers take the time to answer questions to further build the project community, and to build their own personal reputation.

RQ2. *What is the perceived impact of the chatroom use on the software development process?* The most recurrent reported impacts of Slack and Gitter are (respectively) managing the communication (e.g., team updates) and guiding the project tasks (e.g., new issues). The developers are also able to learn the best practices of the projects, and produce better solutions. Finally, there appears that the chatrooms improve the productivity of the developers.

RQ3. *What defines the quality of the chatrooms and their related chat-service?* In terms of quality determinants, a ‘good’ chatroom is first characterized by a knowledgeable and friendly community. In terms of chat-services (e.g., the available features), the improvement of key features (e.g., search management) could help the chatrooms achieve their full potential. The majority of the surveyed developers (81.1%) report a *good to excellent* user experience.

Contributions. A summary of our contributions is as follows: 1) A better understanding of the motives and impact of the developers’ chatrooms, 2) a comparison between an open source (i.e., Gitter) and a proprietary (i.e., Slack) chat services, and 3) insights about the uses of chatrooms by the open communities (in both Slack and Gitter) and the corporate teams (mostly in Slack).

Paper organization. The rest of the paper is organized as follows. Section 2 present the subject systems and the background. Section 3 describes the research methodology. In Section 4, we show the results of the study. In Sections 5 and 6, we provide a discussion and list the limitations of the study. Finally, we review the literature and conclude in

Sections 7 and 8.

2 SUBJECT SYSTEMS

In our work, we focus on the Slack and Gitter modern chatrooms. In Appendix ??, we show a comparison between Slack and Gitter. Slack is a collaboration tool that offers features, such as persistent chatrooms (*channels*) organized by topic, private groups, and direct messaging. Slack was launched in August 2013, and is reported to be “the fastest-growing business application in history” with 8 million daily active users, and 500K organizations that use the tool to collaborate and communicate [11]. Slack allows searching, indexing and integration with external applications. The basic Slack version is free, with the possibility to upgrade to advanced priced features.

Gitter is an open source instant messaging tool for developers and users of GitHub repositories, which came out of beta in 2014. Gitter enables the creation of chatrooms (called *rooms*) for GitHub repositories, i.e., each Gitter room can be linked to a GitHub repository page. Gitter has over 800K users, and 300K rooms from 100+ countries [12]. The main feature of Gitter is a seamless integration with GitHub through GitHub flavored markdowns in chat messages. Gitter has a unique activity feed showing all changes to the associated GitHub repository (e.g., commenting on a pull request or closing an issue). Another integrated feature between Gitter and GitHub is the user hovercards, which are based on the GitHub profiles and statistics (e.g., the number of followers). Similarly to Slack, Gitter is also integrated with other applications, such as Jenkins³, Travis CI⁴, and Bitbucket⁵.

Overall, Slack is a proprietary chatroom, geared for the corporate teams but also accessible to the open communities. An important capability of Slack is the search feature of the chat data. However, the feature is limited in the free plans (i.e., only up to 10K messages). An important issue in Slack is the discoverability of the communities (i.e., a team can only be joined by finding its link and sending a request to join), which limits the ‘openness’ of Slack to the public. On the other hand, Gitter is an open source platform suited for the open communities. Gitter rooms can be easily found and joined. However, the content is lost quite easily, as searching the past messages is not as good as in Slack.

3 METHODOLOGY

We study (i) the motives to use modern chatrooms, (ii) their impact on the development process, and (iii) the characteristics of high quality chatrooms. Because our goal is to perform an in-depth study of instances (i.e., Gitter and Slack) of a phenomenon in its natural context and from the perspective of the participants involved in the phenomenon (i.e., the use of chatrooms) [13], we use a multiple case studies design in our research [14]. Using multiple case studies can increase the robustness of our observations by strengthening or contrasting the patterns observed in the different case studies (i.e., Gitter vs

3. <https://jenkins.io/>

4. <https://travis-ci.org/>

5. <https://bitbucket.org>

Slack) [14]. Our case studies consist of several data collection methods, such as surveys, interviews, and mining software repositories [15] (MSR). Therefore, we use the *mixed methods* research design to generate our findings (i.e., we use both qualitative and quantitative analyses) [16]. More specifically, our quantitative data is used to support our predominant qualitative observations, configuring our research design as a “QUAL + quan” mixed methods [17].

First, we employ our mixed-methods design on a *within-case* basis, i.e., we employ our approach to both Gitter and Slack, individually (i.e., survey, interviews, and MSR analyses). Next, we analyze our results on a *cross-case* basis by comparing our observations obtained for Gitter against the observations obtained for Slack [14]. At this last stage, two authors (both assistant professors) re-analyze all the obtained qualitative and quantitative results to identify the *reinforcing* and *opposing* patterns across our case studies. We provide the details of each data collection method in the following subsections.

3.1 Developers’ Survey

In this section, we describe the process of 1) designing, 2) distributing, 3) analyzing and 4) validating our survey. We design the survey in four parts: *a)* demographics, *b)* motivations, *c)* impact, and *d)* quality determinants.

Before crafting the survey for performing the research, we developed a *preliminary survey*, which served as a pilot study to check whether our research questions were worth investigating. Our pilot survey contained 5 preliminary questions and is available online.⁶ The goals of our questions were to investigate the reasons behind using chatrooms. We received 85 responses to our preliminary survey, which were not incorporated in our main study given the pilot nature of this preliminary survey. We observed that diverse reasons prompted the usage of chatrooms, from handling “database issues” to getting hints about “setup and configuration.”⁷ Therefore, we crafted a more thorough and detailed survey to answer the research questions of this paper. We herein refer to the more detailed survey as simply *survey*.

We obtained 163 responses to our survey, 114 and 48 from Slack and Gitter users, respectively. The remaining response selected *other* (Discord) as their most used chat-services. We codified the responses to perform our analysis. We conducted a set of 21 follow-up interviews to validate the results from the previous step. We provide more details of the four steps in the sections below.

1) Survey Design

We designed our survey in four main parts ⁸ (see Appendix ??). In the first part (Survey part 1), we inquire about the respondents’ **demographics** and their roles in their software projects. For example, we aim to understand if our respondents use the chatrooms from the perspective of a user, maintainer, or developer. If a respondent collaborates in many software projects, we asked him/her to answer the survey based on the project in which he/she is most active.

The remaining three parts are based on the three RQs presented in Section 7. In part 2 of our survey, we asked our respondents what motivates them to both answer and ask questions on Slack and Gitter. We further inquire about the instances when the chatrooms are used, instead of other channels (e.g., mailing lists).

Prior studies [6] [18] [19] [20] report productivity concerns related to the use of the communication and social channels by developers. Consequently, we design part 3 to ask the respondents whether the chatrooms have any impact on their productivity.

Hahn *et al.* [21] report that a developer is more likely to join a project when they have collaborative ties with the project initiator. This finding was confirmed by Casalnuovo *et al.* [22] who found that developers preferentially contribute to projects in GitHub where they have prior social links. Accordingly, we inquire whether the use of the modern chatrooms has a similar impact (i.e., attracting developers to the project associated with the chatroom).

In the final part of the survey (Survey part 4), we focus on the **chatroom quality** as perceived by the respondents. First, our respondents are requested to provide a satisfaction rating of the studied chatrooms. Then, we inquired about the features that they like most in the studied chatrooms as well as the features that they believe need improvement. Lastly, we asked our respondents if they were willing to be contacted for follow-up interviews. The survey has 21 questions in total, 14 of which are open-ended questions. The estimated time to complete the survey is 20 minutes. We opted for not placing any mandatory questions as a means to improve our response rate. For example, studies have shown that it is frustrating to have mandatory questions in surveys [23]. Mandatory questions also increase the probability of participants dropping the questionnaire, or, even worse, to provide low-quality answers in a way to skip the mandatory questions.

2) Survey Distribution

We used a combination of two methods to contact the respondents: 1) we posted our invitation in the chatrooms; and 2) we sent emails to developers. We include the invitation letter sent to the potential participants in Appendix ??.

Posting in the Chatrooms: In Gitter, we manually joined the public rooms that are visible on the *explore* page of Gitter (i.e., 770 rooms). In Slack, we were able to join a set of 29 public Slack rooms, by sending requests to join to the project owners. The only selection criterion to find respondents is whether they were members of the chatrooms at the time. For example, we did not consider activity levels of the respondents or membership duration, since it is our goal to obtain feedback from all kinds of respondents. We posted messages in the chatrooms to contact the respondents in a less intrusive manner. More specifically, we identified the administrators of the chatrooms, whom we contacted directly and asked for permission to distribute the survey. Some administrators (≈ 5) posted the survey request themselves, while others (≈ 20) gave us the permission to post our survey in one of the room channels (e.g., the *random* or *general* channels). Another 11 administrators declined our request. As a result, we received 47 responses from the

6. <https://www.surveymonkey.com/r/XK5K35K>

7. <https://www.surveymonkey.com/results/SM-8DNFCCPM7/>

8. <https://goo.gl/forms/oX4UqWUDRcykBP372>

Gitter respondents from different chatrooms, and another 47 from the Slack respondents from different chatrooms, bringing the total to 94 survey responses. With our survey distribution method, we cannot assume that all the members in a chatroom have read the survey request. Therefore, it is hard to assess the response rate. Because we sent our surveys separately to Gitter and Slack participants, we easily identify the chatroom to which our participants were referring when replying to our surveys.

Email: We contacted developers through emails. Since we are not able to specifically email developers that use Slack or Gitter, the selection criterion in the second phase is simply developer's participation in a GitHub repository (i.e., code commits). To identify whether our participants were referring to Gitter or Slack in their answers, we explicitly asked them to identify the chatroom to which they were referring. Our initial selection of GitHub developers with more than 10 commits—the median number of commits across all developers—resulted in over 4 million developers. We then chose a statistically significant random sample (confidence interval = 2 and confidence level = 95%), resulting in 2,400 developers. We emailed the survey request to the selected developers, receiving an additional 69 responses. At the end of both phases of deployment, we obtained a total of 163 survey responses. The only statistical difference between the demographics of the two deployments is an increase in the number of female respondents (from 3.2% to 13.3%, p -value = 0.016). A possible explanation for the raise of women respondents over e-mails is that women may feel less comfortable to participate in chatrooms due to reasons that must still be studied (e.g., chatrooms might be a male dominant environment).

Since the survey questions are non-mandatory, we find that the questions were skipped by the participants 10 times on average (i.e. not answered by $\frac{10}{163}$ or 6% of the respondents) with a median of 7.

3) Survey Analysis

After receiving the responses to our survey, we performed thematic analysis [10] to analyze the collected data. Thematic analysis is a process involving the qualitative examination of a dataset set to generate a set of themes that capture the intricacies of meaning within the data set. In the first phase of the thematic analysis (i.e., coding), a set of initial codes is generated by collapsing the data into labelling concepts. In our study, a survey response can be collapsed into one or more codes, depending on the complexity and richness of the response. In the second phase of the thematic analysis, the codes are combined into overarching themes that accurately depict the data. For example, the codes 'interactive' and 'real-time' can be categorized under the theme 'response time'. We show examples of the thematic analysis of our data in Appendix ??.

The first two authors collaboratively performed thematic analysis for every survey response, until consensus was reached. Both authors are assistant professors. Saturation is achieved after analyzing approximately 30 to 35 survey responses. The first phase of the thematic analysis took place in five coding sessions for Gitter and four coding sessions

for Slack of about 1.5 hours each. We obtained 56 and 137 codes for the survey part 2 for Slack and Gitter, respectively. An additional 67 and 122 codes resulted from part 3, for Slack and Gitter, respectively. Finally, part 4 of the survey yielded 33 and 24 codes, for Slack and Gitter, respectively. We then generate higher level conceptual themes in the second phase of the thematic analysis to answer our research questions (more details are shown in Tables 1, 2, and 3). In this paper, we report our results on all the obtained themes.

In the remaining of the paper, we refer to developers who participate in the survey as *the respondents*, and to developers with whom we conducted the interviews as *the interviewees*. We further refer to the individual respondents/interviewees using a code of the form S# for Slack and G# for Gitter (# is a unique ID of each respondent, e.g., S15). The Slack respondents span from S1 to S114, and the Gitter respondents from G1 to G48.

4) Follow-up Interviews

Setup. 21 respondents expressed their interest in being part of a follow-up interview. The goals of the follow-up interviews are: (i) to clarify the responses of the survey, (ii) to validate the themes obtained from the second phase of the thematic analysis, and (iii) to collect more details from the personal experience of developers. We have not performed an additional thematic analysis at this stage as we have reached saturation when analyzing our survey responses.

Our follow-up interviews followed a semi-structured format, i.e., we started with a script of questions but we let the respondents go off the script if he or she wishes to provide more information. We contacted the 21 interviewees through their preferred communication channel (e.g., chat, or voice call). The voice/video calls lasted in average 20 minutes, while the chat interviews lasted around 35 minutes. We provide our interview script in Appendix ??.

Analysis. As shown in Appendix ??, we prompted the interviewees with the themes generated during the survey analysis. As such, we used the interviews scripts to validate the generated themes, and the explanations accompanying the answer as illustrative quotes in the paper. In case an interviewee has additional thoughts about a given question, we analyzed their answer using thematic analysis to identify new themes, not generated during the survey analysis step. The results shown in Section 4 reflect the results obtained from both steps.

3.2 Messages Collection from the Chatrooms

We used the Gitter REST API to download archives of the public Gitter rooms. We then use the Gitter REST API to obtain the archives of the 770 joined Gitter rooms. We exclude the 19 Gitter rooms where no conversation has started yet.

In Slack, there is no central browsing page that displays the available public rooms. Instead, each public chatroom has a unique link used to request to join the community. A thorough internet search reveals the links of the top public Slack rooms (e.g., <https://slacklist.info/>). We were able to join 29 public Slack rooms. To collect the room archives, we used the Slack API which requires a unique token for each room. In some rooms, the token is made available to all the members; while it needs to be requested from the

administrators in others. In the end, we were able to collect the archives of 11 public Slack rooms. Our intention in collecting room archives is to verify/refute, when possible, the perceptions of our participants obtained through our qualitative analyses.

From the chat data collected, we computed the *activity level* of the individual chatrooms. The activity level is computed as the median elapsed time between two chat messages of a given chatroom. For example, if the median elapsed time between two messages is less than 2 seconds, this signifies a high activity level. This is important to verify whether the answers from the respondents regarding the activity levels of the chatrooms are in agreement with the actual chat traffic in the chatrooms.

In addition to the activity levels, we compute the *time to solution*. The *time to solution* is the time difference between the first and last message in a *thread*. A *thread* is a set of related messages in a discussion. For example, if a user asks a question and developers answer that question (possibly by having a discussion), the set of messages (e.g., answers or other follow-up questions) related to the question compose a *thread*. Automatically identifying *threads* is challenging because it requires research on using advanced *Natural Language Processing* techniques [24], [25], [26], [27], [28]. For instance, such an automatic approach would need a high accuracy on identifying that a user is asking a question. Even more challenging, the automatic approach would need to automatically identify that a later statement was referring to a specific previous question, thus answering that question. These tasks are challenging because Gitter and Slack do not enforce explicit references between questions and replies when discussions are being held. Thus, because the focus of our research is not on automatically identifying threads, we manually label 200 threads from a randomly selected chatroom (i.e., $Sample_{threads}$) for our investigations. We are aware that Slack offers the ability to provide nested messages, which could potentially be used to automatically identify *threads*. However, we observed that relying on the nested messages from Slack would be an incomplete solution, since we observed (through our manual analysis), that several related messages are not nested. Moreover, Gitter does not provide the feature of nested messages, which is another motivator for us to perform our manual analysis.

We compute the *time to solution* of the threads in $Sample_{threads}$, as the time difference between the first and last message in a thread. Wang *et al.* [29] used a similar approach to detect the question getting a fast response in Q&A forums.

3.3 Demographics

We present an exploratory analysis of the demographics data that we collect from the responses of the respondents. Fig. 1 shows a summary of the demographics data collected from the Slack and Gitter survey respondents.

Gender: The majority of survey respondents (i.e., 90.7%) in both Slack and Gitter have identified themselves as males.

Development experience: As shown in Figure 1a, almost half of the respondents (59 out of 114 and 20 out of 48 in Slack and Gitter, respectively) reported an experience of 10 or more years.

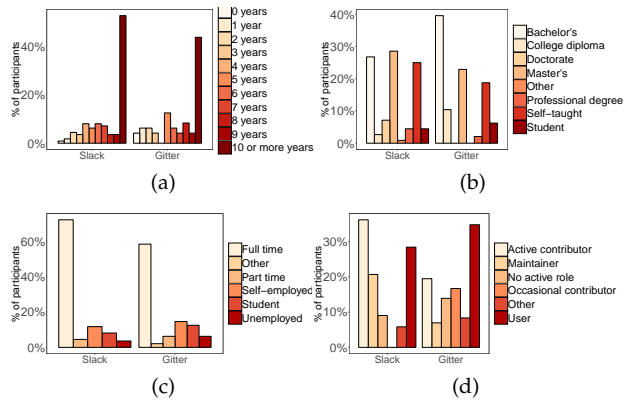


Fig. 1: Demographics of the Slack and Gitter respondents

Education: 61.4% and 62.5% of the respondents have at least a Bachelor degree in Slack and Gitter, respectively. A portion of the respondents (24.6% in Slack and 18.7% in Gitter) disclosed that they are self-taught. Overall, the majority of developers have received a formal education in software development (Figure 1b).

Employment: For the employment status, 70.1% and 58.3% of the Slack and Gitter respondents, respectively, are employed full-time (Figure 1c).

Project role: We asked the participants about their roles in the projects associated to the chatrooms. In Gitter, it was reported that over half of the respondents (i.e., 52.1%) are users of the projects, and do not make contributions to the code base of the projects (Figure 1d). The second most common reported role is “contributor”, with a distinction between the 29.2% active contributors (i.e., make frequent contributions), and the 25.0% occasional contributors (i.e., making sporadic contributions) (Figure 1d). In Slack, the most common reported role is the active contributor role (49.1%), followed by the user role (38.6%). 28.1% and 10.4% of the respondents in Slack and Gitter, respectively, are maintainers of the projects (i.e., have project privileges, such as reviewing code contributions).

We provide in our appendix additional analyses relating the (i) gender; (ii) experience; and (iii) employment status of our participants with the most recurrent themes obtained in each RQ (see Appendix ??).

4 RESULTS

In this section, we answer our RQs by reporting the most common themes (shown in **bold**) that emerge from the second phase of the thematic analysis, as well as some of the codes (shown in *italic*) that result from the coding phase of the thematic analysis. Moreover, we compare and contrast the results from Slack and Gitter. We further include quotes from the participants to provide more insights. The Slack and Gitter participants are anonymously referred to as *S#* and *G#* (respectively), with # as the numeric identifier of a participant. Aside from the quantitative observations described in this section, all qualitative observations are based on the opinions expressed by our participants.

RQ1. Why do developers use the chatrooms?

The first research question explores the reasons behind the participation of developers in the chatrooms. Table 1 shows the list of themes that emerge from the analysis for

TABLE 1: Survey results regarding the motivations of developers to use the chatrooms.

Theme	% respondents agreeing to theme		Illustrating quote
	Slack _{n=114}	Gitter _{n=48}	
Quality of the help	46%	33%	"Slack allows large group discussions in real time around technical features and bug fixes which can often lead to better solutions."
Response time	41%	35%	"I think the immediacy of Slack is one of its big benefits: there's nearly always someone who can answer any given question during the times the channels are busy. Or at least there will be someone who can direct the questioner to a channel that would be more likely to get an answer."
Giving back	40%	44%	"I'm using open libraries every day for free. And I want to help the open source [communities]."
Personal gain	31%	21%	"I would certainly say that as a hiring manager, I would use any impressions formed from interactions on Slack to guide my hiring decisions, if those developers applied to work for me"
Tasks	29%	0%	"In the company I work we use [Slack] like a planning tool (well, to discuss tasks to be done and so on)"
Internal communication	19%	0%	"Our company uses [Slack] for internal communication and collective knowledge sharing and we are expected to ask questions."
Requirement	18%	0%	"I have to use what the creators set up"
Company's goals	8%	2%	"I am motivated to provide clear and transparent answers as much as possible, to maximize the revenue and achieve the company's goals"
Custom support	1%	33%	"With Gitter I can chat with [the library authors] about custom solution, and I believe that this is the main difference"
Personal enjoyment	0%	29%	"Some do it to find people to collaborate with, and others do it to simply be amongst their peers"

both Slack and Gitter. The most recurrent themes for RQ1 are the **quality of the help**, the **response time**, and **giving back to the community**.

1) Similarities between Slack and Gitter

Quality of the help. In both Slack and Gitter, the most recurrent theme is the **quality of the help** provided by the chatrooms. The Slack participants report that asking questions on Slack provides *learning opportunities* including the *best practices*, *access to experts*, and *inside knowledge*. S18 reports that the Slack rooms are "*where the majority of knowledgeable people in my communities are*". The Gitter participants mention that the chatrooms are useful to provide *guidance*, *clarification*, *feedback on new ideas*, and possibly *better solutions*. G33 stated that he usually asks questions when he knows that "*there has got to be a better way / someone who made this already*". G16 goes further and says: "*I needed advice on the architecture of my commercial apps and the gitter chat saved me a lot of money which I'd have wasted on servers if I'd gone with something different*".

Response time. The participants mention that the **response time** in chatrooms is an important motivation for asking questions in both Slack and Gitter. The participants describe the conversations in these chatrooms as 'interactive', 'conversational', and 'instant'. We provide quantitative insights regarding this perception later.

Giving back to the community. As far as providing answers in the chatrooms goes, the most recurrent theme is **giving back to the community**. Specifically, the survey participants claim that they are eager to *reciprocate help*. Others believe that answering questions helps to *attract new contributors* to a project and *grow the community*. The communities in the chatrooms provide *low entry barrier* for the new participants, bringing a *sense of equality* among the members. Although most teams in the chatrooms establish community guidelines according to the interviewee S10, the moderation is considered moderate compared to other platforms, such as Stack Overflow. G33 explains that: "*there*

is no competition of reputation, but rather a nicely balanced space with equal opportunity for everyone to be heard". G33's response indicates that, unlike Q&A platforms such as Stack Overflow, which employ gamification mechanisms for their users to earn experience points (referred at times as "reputation"), chatrooms do not track these experience points, which likely reduces the sense of competition. G44 further mentions that there is no need to go through *pre-moderation* or restrictions based on a *score*.

Personal gain. In addition to altruistic reasons, the survey participants report that there is some **personal gain** from answering questions in the chatrooms. 15.3% (25/163) of the participants claim that answering questions is an opportunity to *learn by teaching*, by "*validating personal assumptions, and having one's personal knowledge and code be fact checked / sanity checked by more knowledgeable people*" (G42). Moreover, providing help is a means to *build a reputation* among peers. 11.6% (19/163) of the participants report that they answer questions to build reliance and gain respect from others in the same community. Finally, 6.1% (10/163) of the participants explain that they promote their own projects when providing help to others.

The participants mention further themes related to the **features** most appreciated in the chatrooms, such as the *volatile content*. S11 clarifies that "*the questions on Slack are less public, disappearing eventually from free Slack rooms, such as Vapor*". In Gitter, the most appreciated feature is the *seamless integration* to GitHub, such as the inline markdown and the repository update panel. In terms of the **topics** discussed, technical discussions, such as *custom code review* and *debugging*, are common. G27 and G32 explain that debugging questions are usually presented as follows: "*I got unexpected results, what did I do wrong?*", or "*Is it possible to do [task X] using this library?*". According to S14, "*the topics span a wide technical spectrum ranging from the architectural level, to the nuts and bolts of specific technical tasks*". The participants also report discussing project documentation issues, such as

the project configuration. G5 explains that “The configuration part would be solved with good documentation, but we never found a project with good documentation”, thus highlighting existing issues with projects documentation (described by the participants as *voluminous, ambiguous, or incomplete*).

2) Quantitative analysis of response time

Regarding the perceived **response time** from our participants, we investigate the chat data of the chatrooms that the participants qualify as fast and we compute the set of chatroom-level metrics, described in Section 3.2). For this purpose, we use our previously explained *time to solution* metric. In Appendix ??, we show the median activity levels of the chatrooms that were perceived as having fast responses. In the chatrooms from which we receive survey responses, the median activity level is 77 seconds. However, the median activity of all the chatrooms from the collected data set is 510 seconds. This indicates that our participants come mostly from active and large chatrooms.

We further investigate whether the speed of response has a relationship with the number of participants in a chatroom. We find a *very weak* correlation between the two factors (Spearman’s correlation = -0.17 and p -value = 8.56e-07). We conclude that the speed of response is not associated with the number of participants in a room. The number of participants has a *very strong* correlation with the number of messages exchanged in a chatroom, as expected (Spearman’s correlation = 0.98 and p -value < 2.2e-16).

Additionally, we compute thread-level metrics based on the manually labelled `Samplethreads`. We find that the median discussion in a chatroom is 101.57 minutes, and the median initial *time to solution* is 12.85 minutes. 61.9% of the threads get a response within an hour. In contrast, Wang *et al.* find that 69.2% answered questions get an answer within one hour on Stack Overflow [29]. As such, the perception of the survey respondents that the chatrooms are significantly faster than other platform such as Stack Overflow is not substantially reflected by the data at hand. Further empirical analysis based on larger samples is needed to further confirm this observation.

3) differences between slack and gitter

The sharing of expert knowledge happens for different reasons in each chatroom. For example, 17.5% of the Slack respondents (20 out of 114) mention that providing guidance to the new members over Slack is **required**, and 7.9% (9 out of 114) report that participation is encouraged to **achieve the company’s goals**. In this respect, S52 says the following: “my senior position requires mentorship over new junior members in our team who require some attention. I want this task to be fulfilled correctly with little distraction. Thus, I am motivated to provide clear and transparent answers as much as possible, to maximize the revenue and achieve the company’s goals”. On the other hand, the Gitter respondents share knowledge for **personal enjoyment**. The participants reportedly enjoy chiming in discussions about *corner cases*, or answer questions when they are *bored, curious, or procrastinating* on their work. Another example is that Slack respondents are concerned about building their reputations to maintain “a good employee image”(S41). In Gitter, the respondents are eager to build their reputations to be recognized as experts by their peers. Indeed, we performed a χ^2 test to check whether the reported themes

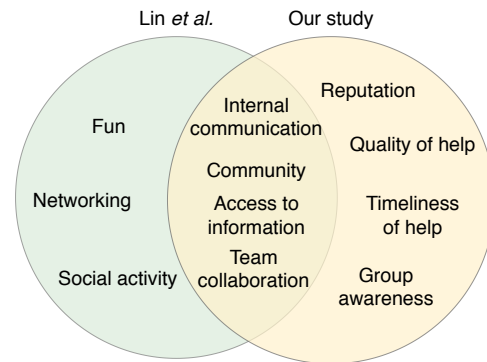


Fig. 2: Summary of the uses of Slack: a comparison of the work by Lin *et al.* [8] and our study

are significantly different depending on the chatroom used (e.g., Gitter or Slack). Our χ^2 test yields a p -value = $2.345997e^{-19}$, which strongly indicates that the reported themes are dependent on the chatrooms at use.

4) differences from Lin et al. [8]

With Slack gaining more adoption and Gitter remaining unexplored, we revisit Slack and investigate Gitter for the first time in our first *Research Question* (RQ): “Why do developers use the chatrooms?”. Our study reveals both similarities and differences with the uses of Slack, that were identified by Lin *et al.* [8]. We show a summary of the similarities and differences on the Venn Diagram, shown in Figure 2. Although there is an important overlap between the two studies, we do observe an evolution of the uses of Slack, from a ‘fun’ chat-service used partly for networking and social activities, to a more ‘corporate’ chat-service with more emphasis on reputation, quality, and awareness about other members.

5) Insights from interviews

Regarding the reasons to participate in Slack and Gitter chatrooms, our interviewees mention that corporate teams are mostly **required** to provide assistance. Mentorship takes place over the chatrooms, in order to ease the onboarding of the new members. The open communities, on the other hand, display different uses of the chatrooms. First and foremost, developers are drawn to the open communities chatrooms to learn more about the project, and get custom help for specific issues. Developers are happy to reciprocate help, and consider the participation in the conversation a leasurable activity. Another aspect mentioned by the interviewees is that chatrooms exhibit two classes of usage: 1) developer support, and 2) user support. The developer support is about providing guidance to developers in learning about, and potentially contributing to, the project or technology associated with the chatroom. The user support happens when technical assistance is provided to the users of a software (e.g., installation issues). As reported by our interviewees, the Gitter open communities are almost exclusively centered around user support. The Slack rooms, on the other hand, exhibit both types of usages even in the open communities. A Slack interviewee (S21) informs us that his work team has both public channels to interact with the users of the software, and private channels for internal communication. The Slack open communities also depend on the channel feature of Slack to direct the incoming

questions to the right audience (e.g., #team-support, #team-devops). Although 5 of the interviewed developers claim that the developer support is more common in Slack, it is not possible to verify as we do not have access to the data from the private channels.

Slack and Gitter are used for the **quality of the help** received within a short **response time**. Sharing knowledge in Slack is sometimes a **required** activity to discuss the projects' **tasks**; while discussions in Gitter may occur for **personal enjoyment**.

RQ2. What is the perceived impact of the chatroom use on the software development process?

The second research question explores the perceived impact of the chatrooms on some aspects of the development process (e.g., resolution of issues). Table 2 shows the list of resulting themes, for both Slack and Gitter. Some of the most common reported themes are **issues resolution, group awareness, and access to information**. Group awareness includes knowledge about members of the project, the project areas they are working on, their current progress, and their plans [4].

1) Similarities between Slack and Gitter

Help with issue resolution. According to *S15, S16, G7, and G14*, the actual **resolution time** of an issue is likely shortened because of the presence of many 'brains and eyes' to help out, and the possibility to tag the concerned developers. *S10* adds that a possible reason is that the discussions in Slack are much faster than in GitHub, where response sometimes takes several days. *S31* goes even further and states that sometimes issues that would have been reported on GitHub are not reported at all, as they are fixed through an exchange in the chatroom. However, *S13* argues that it does not apply to the resolution time of the more complex issues. Others explain that the impact is not on the resolution time itself, but rather on **the visibility** of the issues because 'louder voices tend to win'. *S17* explains that people pressing him on Slack for an issue fix are probably going to be prioritized, simply because of exposure. *G24* reveals that several issues have been found thanks to Gitter users asking questions, specifying that these issues would have gone undiscovered otherwise. Specifically to Gitter, the panel showing the GitHub repository updates in the associated Gitter room (e.g., a new comment on an issue) is important. *G17* claims witnessing a GitHub issue getting more activity because one person responded to it on GitHub, and others noticed it from the Gitter channel.

Access to information. The participants claim that one of the important impacts is **the access to information**, such as *the best coding practices* of a project and *the peripheral knowledge*. For example, it is the opinion of *S19* that "*the chatroom discussions allow for learning the best coding practices in the Vapor project, and for prompting developers to think of better coding approaches*". Eventually, developers are able to produce higher quality products, according to *S12*. Furthermore, developers are able to obtain **up-to-date resources** about a project/technology from the chatrooms discussions. *G23* mentions the Angular project as an example of a fast growing technology, where "*the tutorials*

from even a few months ago may be obsolete". In addition to the up-to-date resources, *G32* argues that "*the chatrooms' discussions contain quite a bit of 'peripheral' knowledge, such as discovering other technologies, patterns, and news*".

Help in brainstorming features. The design of the projects benefits from the feedback in the chatrooms. Specifically, *S14* explains that it helps shape new features, identify shortcomings of the design, and test pre-release versions. Similarly to Slack, the Gitter participants report that the use of the chatrooms has helped in brainstorming new ideas and resolving issues, with a reduced effort. In this regard, we find that Slack and Gitter are similar to the mailing lists, which are also used to discuss the activities of the projects [4].

Productivity. The productivity of developers is a much debated topic in the era of socially-enabled software development, and the developers' chatrooms are no exception. The responses from the participants reflect conflicting impact on their own perceived productivity. 40.5% of the survey participants report a **positive** impact on their productivity, as shown in Appendix ???. The chatroom discussions reduce the 'endless trials and errors' to determine the best way to do a task. *G32*, a maintainer of a GitHub project, explains that "*the chatrooms offload some of the support to the community. They are especially good at helping newbies. That saves the core team cycles to invest into the project.*" However, a non-negligible portion of the survey participants (15.9%) believe in a **negative** impact on the productivity. *S26* believes that the productivity is slowed down because of the FoMo (Fear of Missing out). *G38* confirms that there is always something going on in the chat, which is an incentive to read and participate. Other survey participants (37.5%) have a more nuanced opinion about the impact of the chatrooms on the productivity (i.e., **mixed or no impact**). The participants find that although productivity might be reduced, the benefits of using the chatrooms even out the damage caused. *S29* explains that the chatrooms help speed up cases where progress is stalled, however, it is sometimes misused to ask questions when an email is more appropriate. Related to that, the participants complain that the inability or difficulty to search for past discussions lowers their productivity. Overall, it seems that the use of the chatrooms requires self-discipline to avoid getting side-tracked, as explained by *S14*. Additionally, *S10* explains that it is possible to make use of the features provided in order to manage (i.e., filter or mute) the chat notifications.

2) Differences between Slack and Gitter

We find that the most reported impacts are the **group awareness** and **guiding the project tasks** (*bug fixes* and *new features*) in Slack and Gitter, respectively. Our χ^2 test yield a $p\text{-value} = 6.725e^{-8}$, which strongly indicates that the difference in the reported themes is dependent on the chatroom at use. Group awareness is critical for the software development teams, especially when working remotely. In Slack, the teams are able to *collaborate remotely, make decisions faster* in smaller teams, and *share updates*. It is also believed by *S25* that the Slack rooms help minimize the unnecessary talks and meetings for the collocated teams, and hence allow developers to dedicate more time to the quality of the product. However, *S41* explains that the negative side is that decisions are accumulated in the chatroom itself,

TABLE 2: Survey results regarding the perceived impact of the chatrooms on the development process

Theme	% respondents agreeing to theme		Illustrating quote
	Slack _{n=114}	Gitter _{n=48}	
Issue resolution	66%	63%	"It happens a lot that [participants in the chat share code snippets from their personal projects for bug detection or algorithmic optimization], especially with beginners trying to get help from mentors."
Access to information	53%	42%	"Having access to maintainers who provide undocumented insight is a good way of getting the users through the door"
Brainstorm features	31%	38%	"By helping people use [the project] effectively and by learning about the problems and wishes people have, we can improve the projects existing functionality and generate ideas for new features."
Group awareness	62%	15%	"In distributed teams (virtual, remote, etc) the chat is the closest to real-time, real-life contact we have. it allows for typing-while-thinking, working out problems in short iterative bursts, and developing awareness of what each one is working on."
Project tasks	14%	54%	"The flow that I've seen most often is: a developer asks a question in a channel about something they think is a bug (or about a possible enhancement), the project team (and sometimes community) discuss it briefly and either explain it away (not a bug, not an acceptable enhancement) or ask for a GitHub/JIRA issue to be created."

without being organized or centrally recorded. On the other hand, the discussions in the Gitter chatrooms help the maintainers learn about the problems and wishes people have in an informal setting, so they can improve the existing functionality of the projects and generate ideas for new features (G11). This confirms the use of Gitter for user support within the open communities. In addition to user support, the project maintainers utilize the chatroom to quickly 'gut check' an idea before putting together a full pull request on GitHub. It is reported that the discussions on Gitter are helpful in *raising issues* (that are possibly critical) in the project associated to the chatroom. The issues are further *discussed*, and possibly *solved* through code reviews. G14 explains that discussions on Gitter often lead to new pull requests and issues being opened and closed. However, G21 states that important issues related to the project are discussed solely on GitHub.

3) Insights from interviews

When it comes to the impact on the associated project, managing the internal communication of the team comes on top for the corporate teams. Developers report that 'pinging' a co-worker in the chatroom is less invasive than stopping by their office. Furthermore, the exchange in the chatroom allows for passive knowledge sharing among the co-workers, an important aspect of maintaining project awareness. In an interview with S10, a member in a dozen Slack teams both public and private, he reports that "*company Slack usage tends to be much more structured and proactively administered, by which I mean the use of different user roles and different channel access*". The projects associated to open communities benefit from the chatrooms by attracting new contributors, who consistently participate in the chatrooms.

Slack and Gitter reportedly help developers to support the **issue resolution process**, to have **access to information**, and to **brainstorm features**. However, it is perceived that Slack has more impact on improving the **group awareness**; while Gitter has an impact on **guiding the project tasks** (e.g., new features).

RQ3. What defines the quality of the chatrooms and their related chat-service?

To better inform the design decisions of the chatrooms, we investigate the elements that characterize the quality of the chatrooms in RQ3. Table 3 shows the list of the themes that emerge from the survey analysis. The survey participants report the following ratings of Slack and Gitter respectively: *Excellent* (33.9% - 41.3%), *Good* (47.2% - 41.3%), *Average* (12.3% - 17.4%), *Below average* (3.8% - 0.0%), and *Poor* (2.8% - 0.0%). The most common themes in RQ3 are **the community**, and **the features**.

1) Similarities between Slack and Gitter

Community. The most reported quality determinant in the chatrooms can be summarized as follows: "a friendly, inclusive community of folks with deep technical knowledge" (G19). S11 argues that in terms of expertise, "*it is better to have a good mix of experienced users and novices*". 55.8% (91/163) of the participants stress on the fact that a welcoming and safe atmosphere is key to encourage contribution and exchange. In addition, the occasional participation of the project leaders or maintainers in the discussions sets apart the good chatrooms from others.

Features. In terms of features, it appears that the Slack participants are concerned about improving the current features; while the Gitter participants request new features, such as *bots* and *history management*. In both chatrooms, it is a common complaint that knowledge is lost. In Slack, the conversations in the chatrooms under the free plans are volatile, and the search features provided are quite basic. In Gitter, the search option is almost useless according to G23. Consequently, a non-ephemeral history and a better history management (e.g., advanced search) could be key features for chatrooms to reach their full potential.

2) Differences between Slack and Gitter

Moderation. Based on the results shown in Table 3, moderation of the chat (e.g., staying on topic) is twice more important in Slack than it is in Gitter. Indeed, our χ^2 test yields a *p-value* = 0.01383, which strongly indicates that the importance given moderation is dependent on the chatroom at use.

3) Insights from interviews

Our interviewees did not provide much further insights compared to what we have captured from our survey. Both Gitter and Slack interviewees further stressed the

TABLE 3: Survey results regarding the quality determinants of the chatrooms

Theme	% respondents agreeing to theme		Illustrating quote
	Slack _{n=114}	Gitter _{n=48}	
Features improvement	75%	82%	"Was looking forward to topics, but they're very incomplete. While gitter is great for in the moment support, historical topics get lost quick. Would be great to have topics or pinned threads or something to keep quality info easily accessible." "Discoverability of public Slacks is very poor – pretty much every community Slack I'm in ends up running their own onboarding app on Heroku (because Slack is designed for closed, paid teams, rather than open, free communities)."
Community	52%	63%	"I think a big factor is the general attitude of the channel. It's a welcoming place founded upon mutual respect and an absence of elitism, and it's a large contributor to why I am so invested in it."
Moderation	43%	21%	"[Need for] AI that can watch for uncivilized behaviour is something that every site should be implementing even if it's just to flag conversations for moderator review with no action taken on the part of the software."

importance of a more structured search with the proper management of historical data.

The quality of the **community**, in terms of *friendliness*, *activity* and *expertise* are key determinants of a chatroom's quality in both Slack and Gitter. However, there is more request for **moderation** in Slack, compared to Gitter. Several **features** (e.g., bots and history management) could be improved in Slack and included in Gitter.

5 DISCUSSION

In this section, we provide further insights regarding the demographics of our participants, discuss possible implications from our findings, and outline key differences between the two chat-services, (Slack and Gitter).

Demographics Analysis: while we present here the key takeaways from our demographics, a more detailed discussion can be found in Appendix ???. In terms of *motivation to participate in chatrooms*, most experienced participants are mainly motivated by *giving back to the community*, whereas most inexperienced participants are motivated by receiving *fast responses* and *quality help*. Regarding the *impact of chatrooms in the software project*, we observe that less experienced participants perceive that *access to information* and *feature brainstorming* are the major impacts that chatrooms have in the projects, whereas more experienced participants stress that *issue resolution* and *tasks* are the aspects that are mostly impacted. Lastly, regarding the *perceived quality of chatrooms*, we note that very experienced participants (i.e., 10+ years of experience) praise the *community* aspect of chatrooms, which is also related to the previous motivation of "giving back to the community". As for less experienced participants (0-4 years of experience), the focus is on the *improvement of features*. We suspect that less experienced participants have a mindset set more towards productivity (so they can climb the ladders more quickly).

Possible implications: We discuss the following aspects as possible implications of the findings presented in Section 4.

Project promotion and adoption. The responses from the survey respondents indicate that Slack and Gitter are both appreciated for the *quality of help* received, within a

short response time. For the project owners, the previous finding could be an incentive to further support the online communities through the chat-services, in order to promote the project and boost its adoption. Another finding that supports this claim is the reported role of the chat-services in *community building*. A healthy and thriving developer community around a project could support its adoption.

Time to market. The time factor (i.e., interactive and real-time conversations) was a commonly reported theme as a motivation to use the chat-services. As such, we conjecture that a possible implication of using the chat-services, such as Slack and Gitter, could possibly enable a shortened time to market. For instance, in the case of a closed-source project, resolving pending issues or ambiguities among a team members in a timely manner can speed up the issue resolution process, or the clarification of a set of requirements.

Allocation of tasks. In both Slack and Gitter, the respondents report being able to build a reputation, and establish their expertise in certain areas. The implication of such finding is that the project owners could use this knowledge for more targeted allocation of tasks in the future, or for more efficient recruitment of future contributors. This implication applies to both the open-source and close-source projects.

Quality of the end product. Slack and Gitter allow for access to information possibly not available anywhere else, to brainstorm features, and to speed up the resolution process. For instance, access to information not available in other project documentation may improve the productivity of the new developers, or speed up the onboarding of new developers. Additionally, some of the survey respondents reported that the discussions in the chatrooms enable to optimize the solution to implement a feature or to resolve an issue. As such, a possible implication is an improved end product, thanks to the access to a pool of developers with intricate knowledge of the technology used to build a software product.

Retention of developers. Another commonality between the two chat-services is their contribution to community building. The use of the chat-services to build a sense of belonging, particularly for geographically distributed teams, could possibly encourage the retention of developers, and encourage future contributions. It is also important to

highlight that the observed motivations to use chatrooms (see Table 1) is what distinguishes modern developer chatrooms from other communication channels, such as GitHub pull-requests or issue discussions. Chatrooms provide a more immediate and informal access to resources.

Possible reasons behind the observed differences: both Slack and Gitter are chat-services that overlap in a number of the functionalities offered (e.g., group/private chat, code tagging, user tagging, and integrations with code hosting services). However, we still observe a number of key differences in the ways the two chat-services are used and perceived by their users. We discuss below the possible intertwined reasons behind the observed dissimilarities.

Focus of the discussions. In Gitter, the heart of the discussions appears to be the project itself. On the other hand, the Slack discussions are also focused on the people involved, and their management (i.e., who is doing what when). There is a number of the findings from Section 4 that support this claim. For example, discussions about internal communication and project tasks are mentioned in total by 37.9% of the participants in Slack, and 0 times in Gitter. Moreover, the highest reported impact in Slack is on group awareness (e.g., team updates and progress awareness); while Gitter reportedly impacts aspects of project improvement, such as opening new issues and pull requests. Maintaining group awareness through open-access chatrooms in Gitter may be quite challenging due to the nature of the chat services, that easily allows posts from anyone with no apparent structure within the discussion. Indeed, project awareness requires some knowledge of other teams members and of the ongoing tasks. This is enabled on Slack through access control and multiple channel-based discussions, each aimed at specific matters (e.g., design, deployment, testing, etc). On the other hand, the openness of Gitter enables to hear from a larger pool of participants (i.e., reducing the feedback barrier), which in turn enables to better guide the project tasks (e.g., prioritizing features or issues fixes). These findings can guide the choice of which chat-services is best suited for a given project.

Perception of the users. Other differences observed between Slack and Gitter could be explained by how the users of the chat-services perceive the tools (i.e., a company communication tool vs. a leisurely tool). Indeed, Gitter users are more likely to report that Gitter has no impact on their productivity (22.9%), compared to 6.9% of Slack users. Gitter users join on their own volition, with no requirement to ask or answer questions; while a number of Slack respondents (9%) report that Slack participation is required. Indeed, personal enjoyment was reported as one of the reasons to answer questions on Gitter (18.9%), but was never mentioned by the Slack respondents. This could possibly explain why Slack users have more mixed-feelings about their productivity when using Slack, as shown in Table ???. Similarly, moderation (e.g., staying on topic) is twice as mentioned on Slack as it is mentioned on Gitter, as a quality determinant of chatrooms.

6 THREATS TO VALIDITY

This section discusses the threats to validity of our study.

Threats to conclusion validity concern the relation between the treatment and the outcome. The conclusion validity threat mainly comes from the inherent bias that comes from dealing with human subjects. For instance, in terms of demographics, we observed that half of our participants report an experience of > 10 years in software development, 90.7% are male, and 100% are involved in the more active chatrooms, as measured by the number of participants (median = 709) and the number of exchanged messages within a year (median = 17,836). Therefore, our findings may be biased towards more experienced male developers participating in the active chatrooms. To offset this bias, we asked the participants in the interviews about their experiences with beginner developers, and with the use of the smaller chatrooms in terms of the number of participants and the number of messages exchanged. Additionally, almost 70% of the participants are Slack users. It was not our intention to recruit more Slack participants. However, the total number of Slack users is over 8 million; while Gitter has just over 800K users. Therefore, our population of participants reflects the popularity of each chat-service. To offset bias during thematic analysis (i.e., a manual process subject to human error), we performed the analysis collaboratively, and carried discussions until consensus was reached.

Another potential bias in our work lies in the fact that the survey distribution in the case of Slack is skewed towards public chatrooms, since we have not had access to private chatrooms. However, regardless of the limited access to private chatrooms (in the case of Slack), we were still able to study the motivations of participants that use Slack in a corporate setting. We envision that studying more private chatrooms could enrich our insights regarding the usage of Slack in corporate settings, since private organizations will likely use private chatrooms in Slack.

Lastly, another potential bias in our conclusions is due to the nature of the projects (e.g., size, domain, complexity) of our participants. It may be that their perception regarding the motivation to use chatrooms may be tightly related to the complexity of their project for example. However, our goal in this work is to study the overall perceptions of our participants instead of investigating the impact that the nature of the projects may have on the perceived usefulness of chatrooms.

Threats to internal validity concern our selection of subjects and analysis methods. To better understand the use of the chatrooms by developers, we opted to use a survey to reach the participants. The survey inclusion criterion in the first deployment phase is the membership in the chatrooms. This suggests a bias towards developers that favor the use of the chatrooms, over the general population that might have differing views on the chatrooms. To mitigate this bias, we targeted a more general population of developers in the second phase of the survey (active users of GitHub). We observed an agreement between the results of the two deployment phases.

Finally, there are some risks involved in our chosen methodology for the coding process. We adopted the same methodology used by Treude *et al.* [30] in which coders would sit together and discuss the codes until saturation was reached. Although our methodology saved us time, the

risk of this methodology is the generation of bias in case one of the coders has a more vocal personality than others. However, we do not think that this is our case, since we made sure that every coder had a reserved time to express his/her ideas in the coding sessions. Another limitation of not performing independent coding, is that we do not provide a measurement of *Inter Coder Reliability* (ICR) [31]. The main purpose of computing an ICR is to make sure that coders have a common understanding of the coding scheme that is produced. On the other hand, we believe that our sessions can also reduce misunderstandings effectively because every deviation in the interpretation of a given code was discussed instantaneously. Given that our coders discussed **all** responses together (as opposed to working with samples), we believe that our methodology is sound.

Threats to external validity concern the possibility to generalize our results. Our study focuses on two widely-used developer-oriented chat-services (i.e., Slack and Gitter). Although our findings are specific to Slack (a proprietary service geared towards corporations) and Gitter (an open source service aimed at the open source communities), many observations could be applicable to other chat-services that offer similar services, such as Microsoft teams, Cisco Jabber, or Chatter.

Although we strive for higher response rates (e.g., by providing incentives to 30% of our participants), we still face challenges. For example, we posted our survey solicitation in 770 Gitter rooms. However, we received only 47 responses. There are two possible reasons for this problem. First, as we also learn from our participants, messages get lost quite fast in chatrooms, which reduces the visibility of our survey requests. Additionally, in a number of chatrooms, the admins deemed our request as *off-topic* or as *not in line with the regulations of the chatroom*. As such, our requests were never posted or simply deleted from the chatrooms. Nevertheless, we still believe that our sample of participants can provide valuable insights for the software engineering community.

7 RELATED WORK

In this section, we discuss the related work and explain how they relate to our three research questions.

Motivation of Chatrooms Uses. developer chatrooms have attracted the attention of researchers for different reasons [32]. This is in part due to the important role of chatrooms in managing knowledge, as reported by Waska *et al.* [33]. Specifically, chatrooms support three types of knowledge transfer: the knowledge embedded (i) in developers' heads [18], (ii) in artifacts [34], and (iii) in communities [8]. Storey *et al.* [18] argue that the modern chatrooms mimic the face-to-face communication and allows the transfer of tacit knowledge. Shihab *et al.* [34] find that the project artifacts (e.g., test cases and source code) are constantly discussed in chatrooms. These discussions complement IDEs and *Version Control Systems* (VCS). To support the communities, Lin *et al.* [8] report that software developers congregate in chatrooms to discuss what the community has learned over time. Lin *et al.* [8] further reckon that communication, information discovery, and the

feeling of belonging to a community are some of the most reported motivations behind using Slack, for example.

The Impact of Chatrooms on Software Development. As the usage of the modern chatrooms is increasing at a fast pace, researchers have been investigating the impact of the chatrooms on the software projects. Gutwin *et al.* [4] studied the group awareness in open source projects, and reported that chatrooms (in their case IRC) are beneficial for ad-hoc communication and the overhearing of informal and technical discussions. An analysis by Hendel *et al.* [7] revealed that six distributed development teams use the chatrooms to achieve synchronous communication with occasional asynchronous exchanges. Chatterjee *et al.* [35] observed that developer chats in Slack provide more information on API mentions than Q&A (e.g., Stack Overflow) posts. Storey *et al.* [18] reported that the chatrooms support the one-on-one discussions and the knowledge exchange, among developers. Storey *et al.* [18] also explained that private chat still falls short in terms of supporting the sharing of contextual information.

Although previous studies have highlighted some advantages of using chatrooms in software development, the impact that the chatrooms have on the projects being developed is still unclear. The results of the survey administered in the scope of this study reveal the following impacts:

- Our study confirms the results by Gutwin *et al.* [4] that chat-services support group awareness, particularly in Slack.
- Both Slack and Gitter are found to enable access to information unavailable elsewhere (e.g., peripheral knowledge of a project), which ultimately enables the learning and use of the best practices.
- In Gitter, the feedback loop from a project's users is reduced, which possibly allows an evolution of the project that is more in line with the expectation of the users (e.g., in terms of which bugs to fix or features to implement).

The Quality of Chatrooms. The quality of the communication channels in software development is determined by the features that are offered by these channels (e.g., code highlighting) and their communities (e.g., friendly developers). For instance, Internet Relay Chat (IRC) is a communication channel that is commonly used by open source developers. However, Chowdhury and Hindle [36] identify a non-negligible amount of off-topic discussions in IRC discussions. Off-topic discussions can be a detriment to the efficiency of any chatroom service. For example, a considerable amount of off-topic discussions can hinder knowledge organization and may keep developers distracted. With this issue in mind, Chowdhury and Hindle [36] proposed classifiers to filter out off-topic discussions in programming IRC channels. The classifiers are trained using Stack Overflow data as positive examples of on-topic discussions, and YouTube video comments as the opposite. Although Stack Overflow is praised for the quality of the questions and answers, it is also criticized for its "unwelcoming community" and the high entry barrier for the beginners [37] [38]. For example, Vasilescu *et al.* [37] run a comparison of the representativeness and activity of genders between Stack Overflow and the mailing lists, and reveal that women disengage sooner despite similar activity

levels to men. Mamykina *et al.* [38] claim that the high entry barrier for beginners is established by a need for “reputation points”. For example, a user cannot vote for the quality of an answer if his/her reputation is not at a certain level.

Although the quality determinants of mailing lists and Q&A systems, such as Stack Overflow, have been well investigated, the quality determinants of modern chatrooms, such as Slack and Gitter, are still unclear. Investigating the quality determinants of modern chatrooms is important because their usage has increased exponentially. In this regard, our study reveals the following:

- Although it may be a conscious design decision to keep the chatrooms history volatile (or at the very least hard to search), the survey respondents report a need for mechanisms to manage the chat history, by providing better search features and history management. For example, a project design decisions may be discussed in the chat, and with time the rationale behind such decisions may be lost and hard to recover.
- Although good moderation is a reported quality determinant in both Slack and Gitter, the most reported theme is the quality of the community, in terms of both expertise and friendliness.

8 CONCLUSION

We design a survey to assess the motivations of the developers when using two widely used developers’ services, namely Slack and Gitter. We find that the chatrooms are used by the developers to exchange timely and expert knowledge, motivated by intrinsic reasons (e.g., support the project community), and extrinsic reasons (e.g., improve the reputations). We further attempt to identify the impact of the use of the chatrooms on the software projects. Our findings show that chatrooms reportedly impact the communication management, the development directions, and the resolution time of issues. The developers’ productivity can be positively impacted by the use of chatrooms. We find that quality determinants of chatrooms are mainly the activity levels and expertise of the community members. In our future studies, we will investigate further aspects in the chatrooms, such as (1) the quantitative impact on the projects (as opposed to the perceived impact reported in this study); (2) the nature of the topics discussed; (3) the participation patterns of the developers and non-developers; (4) whether different levels of experience with chatrooms influence the perceptions of our participants; and (5) how gender can influence the participation dynamics on chatrooms. Regarding the participation patterns of developers, we will study whether existing *Natural Language Processing* techniques are suitable for automatically identifying and classifying discussion threads [39]. Lastly, we plan to extend our future studies to include other modern developer chatrooms, such as *GitHub Discussions*.

REFERENCES

[1] F. P. Brooks Jr, *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition, 2/E.* Pearson Education India, 1995.

[2] B. Curtis, H. Krasner, and N. Iscoe, “A field study of the software design process for large systems,” *Communications of the ACM*, vol. 31, no. 11, pp. 1268–1287, 1988.

[3] J. D. Herbsleb and A. Mockus, “An empirical study of speed and communication in globally distributed software development,” *IEEE Transactions on Software Engineering*, vol. 29, no. 6, pp. 481–494, 2003.

[4] C. Gutwin, R. Penner, and K. Schneider, “Group awareness in distributed software development,” in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pp. 72–81, ACM, 2004.

[5] J. Bughin, R. Dobbs, C. Roxburgh, H. Sarrazin, G. Sands, and M. Westergren, “The social economy: Unlocking value and productivity through social technologies,” *McKinsey.com*, 2012.

[6] M. Storey, A. Zagalsky, F. F. Filho, L. Singer, and D. M. German, “How social and communication channels shape and challenge a participatory culture in software development,” *IEEE Transactions on Software Engineering*, vol. 43, pp. 185–204, Feb 2017.

[7] M. Handel and J. D. Herbsleb, “What is chat doing in the workplace?,” in *Proceedings of the 2002 ACM conference on Computer Supported Cooperative Work*, pp. 1–10, ACM, 2002.

[8] B. Lin, A. Zagalsky, M.-A. Storey, and A. Serebrenik, “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*, pp. 333–336, ACM, 2016.

[9] E. Parra, A. Ellis, and S. Haiduc, “Gittercom: A dataset of open source developer communications in gitter,” in *Proceedings of the 17th International Conference on Mining Software Repositories*, pp. 563–567, 2020.

[10] V. Braun and V. Clarke, “Using thematic analysis in psychology,” *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[11] “Slack - about us.” <https://slack.com/about>. Accessed: 2018-09-30.

[12] “Gitter - explore.” <https://gitter.im/explore/>. Accessed: 2018-09-30.

[13] M. D. Gall, W. R. Borg, and J. P. Gall, *Educational research: An introduction*. Longman Publishing, 1996.

[14] R. K. Yin, *Case study research and applications: Design and methods*. Sage publications, 2017.

[15] A. E. Hassan, “The road ahead for mining software repositories,” in *2008 Frontiers of Software Maintenance*, pp. 48–57, IEEE, 2008.

[16] J. W. Creswell, *Educational research: Planning, conducting, and evaluating quantitative*. Prentice Hall Upper Saddle River, NJ, 2002.

[17] N. Ivankova, J. Creswell, and V. Plano Clark, “Foundations and approaches to mixed methods research,” *First steps in research*. Pretoria: Van Schaik, pp. 253–282, 2007.

[18] M.-A. Storey, L. Singer, B. Cleary, F. Figueira Filho, and A. Zagalsky, “The (r) evolution of social media in software engineering,” in *Proceedings of the on Future of Software Engineering, FOSE 2014*, (New York, NY, USA), pp. 100–116, ACM, 2014.

[19] L. Singer, F. Figueira Filho, B. Cleary, C. Treude, M.-A. Storey, and K. Schneider, “Mutual assessment in the social programmer ecosystem: An empirical investigation of developer profile aggregators,” in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work, CSCW '13*, (New York, NY, USA), pp. 103–116, ACM, 2013.

[20] L. Singer, F. Figueira Filho, and M.-A. Storey, “Software engineering at the speed of light: How developers stay current using twitter,” in *Proceedings of the 36th International Conference on Software Engineering, ICSE 2014*, (New York, NY, USA), pp. 211–221, ACM, 2014.

[21] J. Hahn, J. Y. Moon, and C. Zhang, “Emergence of new project teams from open source software developer networks: Impact of prior collaboration ties,” *Information Systems Research*, vol. 19, no. 3, pp. 369–391, 2008.

[22] C. Casalnuovo, B. Vasilescu, P. Devanbu, and V. Filkov, “Developer onboarding in github: the role of prior social links and language experience,” in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pp. 817–828, ACM, 2015.

[23] S. Stieger, U.-D. Reips, and M. Voracek, “Forced-response in online surveys: Bias from reactance and an increase in sex-specific dropout,” *Journal of the American society for information science and technology*, vol. 58, no. 11, pp. 1653–1660, 2007.

[24] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *Journal of machine learning research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

- [25] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60, 2014.
- [26] P. Chatterjee, K. Damevski, N. A. Kraft, and L. Pollock, "Software-related slack chats with disentangled conversations," in *Proceedings of the 17th International Conference on Mining Software Repositories*, pp. 588–592, 2020.
- [27] P. Chatterjee, K. Damevski, and L. Pollock, "Automatic extraction of opinion-based q&a from online developer chats," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, pp. 1260–1272, IEEE, 2021.
- [28] O. Ehsan, S. Hassan, M. E. Mezouar, and Y. Zou, "An empirical study of developer discussions in the gitter platform," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 30, no. 1, pp. 1–39, 2020.
- [29] S. Wang, T.-H. Chen, and A. E. Hassan, "Understanding the factors for fast answers in technical q&a websites," *Empirical Software Engineering*, vol. 23, no. 3, pp. 1552–1593, 2018.
- [30] C. Treude, F. Figueira Filho, and U. Kulesza, "Summarizing and measuring development activity," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pp. 625–636, 2015.
- [31] C. O'Connor and H. Joffe, "Intercoder reliability in qualitative research: debates and practical guidelines," *International Journal of Qualitative Methods*, vol. 19, p. 1609406919899220, 2020.
- [32] R. Alkadhi, J. O. Johanssen, E. Guzman, and B. Bruegge, "React: An approach for capturing rationale in chat messages," in *Proceedings of the 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 175–180, Nov 2017.
- [33] M. M. Wasko and S. Faraj, "'it is what one does': why people participate and help others in electronic communities of practice," *The Journal of Strategic Information Systems*, vol. 9, no. 2-3, pp. 155–173, 2000.
- [34] E. Shihab, Z. M. Jiang, and A. E. Hassan, "Studying the use of developer irc meetings in open source projects," in *Proceedings of the 2009 IEEE International Conference on Software Maintenance*, pp. 147–156, Sept 2009.
- [35] P. Chatterjee, K. Damevski, L. Pollock, V. Augustine, and N. A. Kraft, "Exploratory study of slack q&a chats as a mining source for software engineering tools," in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, pp. 490–501, IEEE, 2019.
- [36] S. A. Chowdhury and A. Hindle, "Mining stackoverflow to filter out off-topic irc discussion," in *Proceedings of the 12th Working Conference on Mining Software Repositories, MSR '15*, (Piscataway, NJ, USA), pp. 422–425, IEEE Press, 2015.
- [37] B. Vasilescu, A. Capiluppi, and A. Serebrenik, "Gender, representation and online participation: A quantitative study of stackoverflow," in *Proceedings of the 2012 International Conference on Social Informatics, SOCIALINFORMATICS '12*, (Washington, DC, USA), pp. 332–338, IEEE Computer Society, 2012.
- [38] L. Mamykina, B. Manoim, M. Mittal, G. Hripscak, and B. Hartmann, "Design lessons from the fastest q&a site in the west," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 2857–2866, ACM, 2011.
- [39] O. EHSAN, S. HASSAN, M. EL MEZOUAR, and Y. ZOU, "An empirical study of developer discussions in the gitter platform," *ACM Transactions on Software Engineering and Methodologies*, p. toAppear, 2020.



Mariam El Mezouar is an assistant professor at the School of Computing at Queen's University (Canada). She received her PhD in Computing from Queen's University in 2019. Her research interests include mining software repositories, collaborative software development, and the social aspects of software engineering.



Daniel M. German is professor in the Department of Computer Science, University of Victoria, where he does research in the areas of mining software repositories, open source software ecosystems and the impact of intellectual property in software engineering.



Daniel Alencar da Costa is a Lecturer (Assistant Professor) at the University of Otago, New Zealand. Daniel obtained his PhD in Computer Science at the Federal University of Rio Grande do Norte (UFRN) in 2017 followed by a Postdoctoral Fellowship at Queen's University, Canada, from 2017 to late 2018. His research goal is to advance the body of knowledge of Software Engineering methodologies through empirical studies using statistical and machine learning approaches as well as consulting and documenting the experience of Software Engineering practitioners.



Ying (Jenny) Zou is the Canada Research Chair in Software Evolution. She is a professor in the Department of Electrical and Computer Engineering, and cross-appointed to the School of Computing at Queen's University in Canada. She is a visiting scientist of IBM Centers for Advanced Studies, IBM Canada. Her research interests include software engineering, software reengineering, software reverse engineering, software maintenance, and service-oriented architecture. More about Ying and her work is available online at <http://post.queensu.ca/~zouy>