

Software Engineering at the Speed of Light: How Developers Stay Current using Twitter

Leif Singer
University of Victoria
Victoria, Canada
lsinger@uvic.ca

Fernando Figueira Filho
Universidade Federal do Rio
Grande do Norte
Natal, Brazil
fernando@dimap.ufrn.br

Margaret-Anne Storey
University of Victoria
Victoria, Canada
mstorey@uvic.ca

ABSTRACT

The microblogging service Twitter has over 500 million users posting over 500 million tweets daily. Research has established that software developers use Twitter in their work, but this has not yet been examined in detail. Twitter is an important medium in some software engineering circles—understanding its use could lead to improved support, and learning more about the reasons for non-adoption could inform the design of improved tools.

In a qualitative study, we surveyed 271 and interviewed 27 developers active on GitHub. We find that Twitter helps them keep up with the fast-paced development landscape. They use it to stay aware of industry changes, for learning, and for building relationships. We discover the challenges they experience and extract their coping strategies. Some developers do not want to or cannot embrace Twitter for their work—we show their reasons and alternative channels. We validate our findings in a follow-up survey with more than 1,200 respondents.

Categories and Subject Descriptors

H.5.3 [Group and Organization Interfaces]: Computer-supported collaborative work

General Terms

Human Factors

Keywords

Social Media, Microblogging, Twitter, Awareness, Learning

1. INTRODUCTION

Like many disciplines that rely on human knowledge and invention, Software Engineering is rapidly transforming. Developers have to stay informed, maintain relationships, and keep their domain knowledge up to date, relying on many forms of communication media to manage it all.

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 ACM 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.

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

ICSE'14, May 31 – June 7, 2014, Hyderabad, India
ACM 978-1-4503-2756-5/14/05
<http://dx.doi.org/10.1145/2568225.2568305>

Email is used to deliver commit messages and to support code reviews [15], IRC is used during co-development activities, and software forges support community development [13]. Developers blog about their experiences and disseminate them among their readers [12]. Stack Overflow helps developers access a crowd of experts willing to help them with their challenges [9]. They use developer-specific social media to connect and collaborate with one another [17]. Despite—or because of—the availability of these tools, it can be a challenge keeping up with and making sense of new information, tools, and practices; the importance of social media and related tools in supporting software developers is accepted. However, other communication tools are not yet well understood in a software engineering context.

One of these tools is Twitter, the popular microblogging service. More than 500 million registered users¹ post over 500 million tweets a day². Many software developers use it to communicate about software engineering topics [2, 18], but we do not know why some developers adopt it and fervently use it, while others do not and question its value. While most developers will have a hunch regarding the benefits and drawbacks of microblogging, research has yet to provide empirical evidence for such hunches. Uncovering how developers have appropriated Twitter for their work could help software engineering research better understand developers' needs and challenges regarding communication, learning, and collaboration. Understanding the reasons for non-use could shape the design of better tools that would help software engineers make sense in this fast-moving world.

We report on a qualitative study focused on discovering the perceived benefits that Twitter brings to adopters, as well as understanding why others reject it. We conducted an exploratory survey with 271 GitHub users, 27 interviews, and a validation survey with 1,207 GitHub users. We find that developers use Twitter to filter and curate the vast amount of information available to them as part of their programming activities. The benefits we find it brings can be categorized across three themes: *awareness* of people, activities, news, trends, and practices; *learning* of new technologies, practices, cultures, tools, and concerns; as well as building *relationships*. Developers who feel that Twitter benefits them rely on a variety of strategies for posting and reading Twitter content.

¹http://semioast.com/publications/2012_07_30_Twitter_reaches_half_a_billion_accounts_140m_in_the_US

²<https://blog.twitter.com/2013/new-tweets-per-second-record-and-how>

Furthermore, we discover several reasons why developers may choose not to use Twitter. Notably, some of the reasons for non-adoption are what adopters praise about the service: for example, some feel constrained by Twitter’s 140-character limit, though others welcome the resulting succinctness of tweets. Non-adopters also worry about Twitter’s information overload, while adopters talk about how it helps them manage information.

This paper is structured as follows. We review related work in Section 2 and introduce our study design in Section 3. Section 4 presents our findings, which are then discussed in Section 5. We conclude in Section 6.

2. RELATED WORK

Microblogging is the practice of posting short thoughts, ideas, and other content to the Web [11]. Several different services exist, and each may implement the concept in a slightly different way. Twitter, currently the most popular microblogging service, restricts its users to posting up to 140 characters at a time, and integrates features of social network sites: users can follow each other’s posts (*tweets*) and read the tweets by those they follow in a combined, time-ordered list—often called timeline or newsfeed.

In an early study on Twitter, Java et al. [7] analyze the connections between users and the content of the messages they post. Their research identifies several core activities of Twitter users, such as conversations, reporting news, and sharing information through URLs. The most common activity was what Java et al. call *daily chatter*: people talking about their current actions and their plans for the day.

Honeycutt and Herring [6] investigate conversations in an early version of Twitter, which did not yet support mentioning other users or conversational threads. They show how users appropriated Twitter through conventions and note that some started to use Twitter for collaboration as well.

Boyd et al. [3] examine the phenomenon of *retweeting*, a convention that emerged among early Twitter users. A user who wants to pass on someone else’s tweet to their own followers would add a prefix such as “RT: ” to the tweet and post it from their own account. Boyd et al. classify retweeting as a conversational practice.

Marwick and Boyd [10] examine how Twitter users perceive their followers and how they adapt their content to this imagined audience. Because different kinds of people might be following a user, people try to make their tweets relevant and non-confrontational.

In their analysis of the entire Twitter database, Kwak et al. [8] investigate social network topologies, influential users, and the spreading behavior of content through retweets over time. Among their findings, they report that Twitter supports fast dissemination of information.

Yammer is a microblogging service much like Twitter, but designed for corporate use. Zhang et al. [20] investigate how employees of a large enterprise use Yammer and how its usage differs from Twitter. They find that employees use Yammer more for news about groups and less for posting content about themselves. Conversations on Yammer seem to be longer than on Twitter. Study participants reported that Yammer helps them stay aware of what colleagues are working on, and that it helps them make new connections.

In their exploratory study, Zhao and Rosson [21] interviewed Twitter users working in corporate settings. They find that even in the corporate world, Twitter helps people

keep in touch with both friends and colleagues. It can increase awareness regarding personal and work-related events, and through this, can support the formation of common ground and rapport between employees.

2.1 Twitter and Software Engineering

Bougie et al. [2] conducted quantitative and qualitative analyses of 11,679 tweets by developers. They find that software engineers’ tweets involve conversation and information sharing more often than comparable studies examining more general populations. In their qualitative analysis of a sample of 600 tweets, they identify four developer-specific categories of content.

Tian et al. [18] chose a random sample of tweets containing a set of potentially development-related hashtags, such as #scrum, #java, and #testing. The authors then determined content categories for a subset, the tweets’ popularity, and analyzed which categories are retweeted the most. Relatedly, Prasetyo et al. [14] automatically classified tweets according to their relevance to software engineering.

Wang et al. [19] analyzed 568 tweets from 24 Twitter accounts associated with the open source content management system Drupal. The authors find that the Drupal project uses Twitter to communicate issues, documentation, and blog posts to its community. Twitter also serves as a channel to solicit contributions from users.

To the best of our knowledge, there are no in-depth qualitative studies on why and how software developers use Twitter. Also, there is no literature on why software developers might *not* want to use Twitter for their work. To fill this gap, we set to understand *why* and *how* software developers may or may not use Twitter. The following section describes our study design in detail.

3. STUDY DESIGN

We conducted a Grounded Theory-based [4] study, which consisted of three phases of data collection (exploratory survey, interviews, validation survey), and iterative phases of analysis.

Our goal was to involve developers who do and do not use Twitter. In previous work [13], we were able to recruit such participants from GitHub³. GitHub is also a popular code hosting site with a large user base that could provide a relatively diverse pool of potential participants. For these reasons, we used GitHub in our recruiting efforts.

First, we sent an online survey with open questions to 1,160 GitHub users⁴. In this exploratory survey, we asked about their reasons for reading and posting on Twitter, benefits and challenges of Twitter use, as well as the process for discovering and following users. We included a section targeting non-adopters, asking them about their reasons for not using the service. The questions in the exploratory survey were open-ended, and we received 271 responses.

In the second phase, we interviewed some of the survey respondents who had volunteered for interviews. Prior to this interview phase, we analyzed the exploratory survey responses and used them as a guide for the semi-structured interviews.

During our analysis of the exploratory survey and inter-

³<https://github.com>

⁴Selected from GitHub’s public event stream in May 2013, choosing users with public email addresses.

view data, we wrote memos about recurring themes and emerging concepts, constantly comparing our findings on different levels of abstraction. Through this exploratory process, we found that: Twitter can provide *value* to some software developers, and some developers are constantly facing various *challenges* with adopting or using Twitter. To counter this, they use diverse *coping strategies*. Some software developers do not want to or cannot use Twitter at all. This analysis led to our five research questions, detailed in the next section.

We then employed axial coding, iterating through our exploratory survey responses and interview transcripts, to answer our research questions. The final set of themes that emerged informed the third phase of our research: a validation survey sent to 10,000 GitHub users, receiving over 1,200 responses.

The questionnaires and the script for the semi-structured interviews are available in a technical report [16]. In our coding and memoing activities, we used the work by Corbin and Strauss [4] as a methodological guideline. We further used the work by Hoda et al. [5] for methodological advice that is more specific to software engineering research based on Grounded Theory.

3.1 Research Questions

The first three research questions are focused on the value Twitter can provide to software developers. Our first research question approaches the *awareness* aspect:

RQ 1: How does Twitter increase software developer *awareness* of people, trends, and practices?

Another theme that showed us how Twitter can provide value was that it may help developers *extend their knowledge* about new technologies, practices, tools, and software engineering concerns (such as security or usability). This leads us to our second research question:

RQ 2: How does Twitter help software developers *extend their software knowledge*?

The third main theme related to Twitter’s value to software developers was that of *forming and maintaining relationships with others*, as well as *building trust and rapport* with them:

RQ 3: How does Twitter *nurture relationships* between software developers?

While analyzing the responses to our exploratory survey, it became clear that Twitter also poses *challenges* to software developers:

RQ 4: What are the *challenges* faced by software developers using Twitter, and how do they cope with them?

Finally, several survey answers simply stated that the respondent was not using Twitter. We believed it was important to understand the perspective of *non-adopters* as well:

RQ 5: What are reasons for *non-adoption of Twitter* by software developers?

3.2 Participants

This research targeted users of GitHub, a popular code sharing site. This meant we would not restrict ourselves to Twitter users in general, but that we would be able to access software developers who might also be using Twitter. To characterize the participants of our exploratory survey and interviews, we downloaded their account details for

Twitter and GitHub, where available. The exploratory survey had 271 survey respondents (response rate: 23%); from these, we received the details for 188 Twitter accounts and 254 GitHub accounts. 94 participants volunteered for an interview, however, we achieved saturation of the themes we identified after 27 interviews. Interviews were conducted via Skype, recorded, and transcribed. Each interview lasted on average 38 minutes (median: 36 minutes). For these 27 interviewees, we found the data for 26 Twitter accounts and 27 GitHub accounts. The interviewees were from 9 geographic regions: North America, Middle America, South America; Europe; Africa; West Asia, Central Asia, East Asia; and Australia / New Zealand.

3.2.1 Exploratory Survey Respondents

270 of the 271 survey respondents said they develop software. Of these, 221 were professional developers (82%). 172 developers said they worked on private projects, and 155 said they worked on one or more open source projects. 160 respondents said they use Twitter at least once a week (59%). 37% were from North America, and 25% were from Europe. For 24%, we were unable to find their location.

Account ages ranged from 4 months to 6 years and 11 months. The number of tweets posted per account ranged from 0 to 41,079. The number of followers ranged from 0 to 11,469, with a median of 144 (average: 490). The number of accounts people followed ranged from 0 to 2,600. On GitHub, users can indicate whether they are available for hire: 80 respondents said they were for hire and 171 said they were not. Account ages ranged from 4 months to 5 years and 8 months.

3.2.2 Interviewees

All of the 27 interviewees said they develop software in some way. Of these, 25 were professional developers (93%). 20 developers worked on private projects, and 20 worked on one or more open source projects. 22 respondents said they use Twitter at least once a week (81%). 44% were from North America, and 30% were from Europe. For 4%, we were unable to find their location.

26 interviewees had Twitter accounts. Account ages ranged from 1 year and 7 months to 6 years and 2 months. The number of tweets posted per account ranged from 4 to 29,644. The number of followers ranged from 11 to 7,921. The number of accounts people followed ranged from 23 to 1,999. On GitHub, 9 interviewees said they were for hire; 18 said they were not. Account ages ranged from 5 months to 5 years and 7 months.

3.2.3 Validation Survey Respondents

1,207 GitHub users answered our validation survey. 987 of them were professional software developers, 634 said they were contributors to open source projects, and 812 of them were using Twitter at least once a week. 677 respondents indicated that they used Twitter for their development-related work. As this is the population that most of our research questions aim at, the results we report from the validation survey are filtered for these 677 software developers who use Twitter in their work.

3.2.4 Summary

The majority of our population consisted of professional software developers—82% in the exploratory survey, 93%

in the interviews, and 82% in the validation survey. While this is a good indicator for research that will be relevant to practitioners, our participants were still special. All of them used GitHub and many worked on private and / or open source projects. Over half of them used Twitter at least once a week. These facts indicate that our population was comprised of people that were relatively sympathetic to social media and open source software development. This may be a stark contrast with developers in large enterprises who might not be allowed to participate in open source. Yet, in our view, the most relevant quality of our participants was that they were not only novice or hobby programmers, but instead professional developers working in software companies. Having targeted only users of GitHub, this was a surprising result. Most participants were from North America or Europe.

Yet, as the data collected from our participants' GitHub and Twitter accounts show, our population was relatively diverse internally. On GitHub, users' numbers of followers ranged widely from 0 to 1,321. On Twitter, that value ranged from 0 to 11,469. Similarly, our participants were not just veterans or newcomers to the services: account ages ranged from 5 years and 8 months to 4 months for GitHub, and from 6 years and 11 months to 4 months for Twitter. This indicates that at least internally, our results may be considered diverse.

4. FINDINGS

This section reveals the main themes we found in our study. To illustrate the different aspects of each theme, we provide a selection of quotes from the exploratory survey (indicated by E#) and interviews (P#). Where applicable, we provide quantitative results from the validation survey.

Fig. 1 shows the results for the Likert-type scale questions from our validation survey as net stacked distributions: neutral answers are not reported, percentages are reported for disagreement and agreement. We found relatively high agreement with our findings, even though some questions—such as those related to job opportunities or business networking—were seen as less agreeable. This could partly be explained by the idea that such activities require certain contexts and skills that might be less prevalent. However, even the lowest figure—28%—represents 186 professional software developers who agree that Twitter helps them access job opportunities.

4.1 RQ 1: How does Twitter increase developer awareness of people, trends, and practices?

The themes that emerged from the data can be categorized into **activities** that Twitter users reported and the **impact** the activities have on their awareness of people, trends, and practices.

Activity: Following Developers and Projects. “I think the main advantage is to be in contact with people who are developing things that I use. [...] I can communicate with them through Twitter.” [P11]

Exploratory survey respondents and 13 interviewees said they follow specific developers and projects relevant to their work. This provides a channel to other developers that create the code they use. Direct conversations with these developers are possible, providing information that would have been more cumbersome to obtain otherwise. For some,

On Twitter, I follow leaders in my technological niche, which helps me stay current about the latest technologies and practices.



Twitter helps me promote projects and technologies I work with.



Twitter helps me keep up to date about technologies and tools I use for software development.



Twitter helps me stay aware of new trends and practices in software development.



Twitter helps me extend my knowledge by exposing me to technologies and practices I should learn in software development.



Twitter has helped me become a better programmer.



Twitter helps me learn about things I wasn't actively looking for.



Twitter helps me manage my own image as a developer.



Twitter helps me build community around projects I contribute to.



Twitter helps me do business networking.



Twitter helps me discover interesting software developers.



Twitter helps me build trust or rapport with other developers.



Twitter helps me receive validation from others for my work.



Twitter helps me give and get feedback about projects I work with.



Twitter helps me access job opportunities.



I carefully decide whom I follow in order to avoid information overload on my Twitter feed.



On Twitter, I usually follow people on a trial basis and unfollow them if they post irrelevant or too much content.



Figure 1: Results from our validation survey.

Twitter is a richer alternative to RSS. Being able to directly contact developers and projects was important to them.

Activity: Following Technical News Curators. “So I follow people who do read a lot on Digg and I’ll follow their posts and they usually have some interesting things to say on the technology, so I get most updates, technology-wise, from Twitter actually.” [P15]

Many Twitter users selectively tweet URLs from news sites, blogs, and other tweets. Following such accounts provides important updates without having to sift through lots of content themselves. Developers followed people whose judgement they trusted. Over time, they knew who had

posted interesting content previously, and used that knowledge to decide which content would be worth looking at. Five interviewees told us that they follow such accounts.

Activity: Following Thought Leaders. *“But the majority of the people I follow are just [...] leaders in whatever it is they do, and it’s just that they usually have a lot of insight [...]” [P27]*

Developers follow thought leaders in their respective niches to keep on top of what a community is talking about. These leaders are said to “[shape] the community” [P24] by pushing relevant content to their followers. Participants were not only interested in updates to specific projects, but also in which technologies thought leaders were interested in. This phenomenon may result in a community that is confined to the perspectives of a few individuals who define what is interesting. Seven interviewees reported to follow thought leaders. From the validation survey, we see that the majority of survey respondents follow leaders in their niche to stay current about the latest technologies and practices.

Activity: Promoting Project Activities. *“I try to promote [my project] [...] hoping that someone will at some point be interested enough to follow through and get some more information from the links that I post. [...] [More] end users will write articles, blog posts, making the project more well known, which in turn strengthens its brand, which makes it more interesting for customers.” [P19]*

To provide awareness, developers actively post content. Some use Twitter to promote their own projects, which helps them get feedback and attract contributors. It might also help with their careers, as popular projects can be an important part of a developer’s profile. However, their promotion is not restricted to code—they also promote blog posts. Similarly, developers use Twitter to promote local events, hoping to reach a more diverse audience. Promoting and advocating projects and practices can also serve a strategic role: if technologies become more popular, there will be more demand for the expertise of their creators and early users. Six of our interviewees told us about these strategies. From the validation survey, 62% of the respondents (416) indicated that Twitter helps in the promotion of projects and technologies they work with.

Impact: Just in time Awareness. *“It was evolving way faster than I was able to keep up with it. And the only way to keep up was to follow some Node.js people on Twitter. It was remarkable for that.” [P8]*

Some developers work in an industry that is evolving quickly and they have to adapt frequently. They use Twitter in an effort to stay current with technologies, projects, and tools. The service helps them to access new tools and practices as they became available. This allows them to become familiar with technologies that might become popular—at their own pace and with less time pressure. A consequence is that developers may become more productive through better tools and practices and, in turn, become more competitive. This impact was reported by 16 interviewees. In the validation survey, about 70% of the respondents agreed that Twitter helps them keep up to date with changes to the technologies they are already using, and to become aware of new trends and practices in software development.

Impact: Access to Diverse Opinions. *“I think the thing about Twitter is, there’s no sort of restrictions about what you’re talking about, you know? With GitHub, all you’re really ever talking about is that specific bit of code*

that is relevant. And the community that is interested in that little bit [...] if you could contrast them I guess it’s the diversity of Twitter.” [P18]

Exposure to diverse information sources benefits developers in different ways. Six interviewees said they get information about niches that are interesting and relevant to them, but which would not come up in their own news resources. As such, they are exposed to new perspectives. Developers also gain access to a broader set of topics not limited to code-related resources in their own niches. By being exposed to diverse viewpoints, developers think more broadly about software development.

Impact: Dissemination of Knowledge. *“Sometimes I just dig into topics because I have a problem that needs solving. Then I think it is rather rude not to describe the [solution], because [other] people have to go the same way and discover the same things. So I think it’s sort of thinking a bit about humanity.” [P1]*

Developers feel a need to share their discovered solutions with others. However, disseminating knowledge isn’t restricted to posting on Twitter. They also take advantage of other tools, such as Skype group chats or IRC channels, to pass on the resources they find on Twitter. Some developers act as information brokers, e.g., P20 describes himself as an “in-between guy” using these different channels. Three interviewees told us about this impact.

Impact: Increased Adoption. *“The [Node.js] people did a lot of evangelism because they wanted to get their product adopted and they were very effective at that. They did that quite well but that also meant that the ecosystem was built to be aware of social media and aware of how to contact more people more effectively.” [P22]*

Some developers use social media in general and Twitter in particular to push technologies and practices they have an interest in. In other instances, developers suspect that whole communities were set up for promotion through social media by key people in the respective technologies. Two interviewees talked to us about this phenomenon.

4.1.1 Summary

On Twitter, developers follow a) other developers, b) projects, c) news curators, and d) thought leaders to stay aware of current technologies and practices. This lets them access diverse opinions. Some use Twitter to promote their own activities, content, and projects—this helps them disseminate knowledge and increase adoption of technologies.

4.2 RQ 2: How does Twitter help developers extend their software knowledge?

Our second research question asks how Twitter supports developers’ learning. We found that developers value learning about new technologies, and that Twitter plays a role in building concrete and tacit knowledge of technologies, tools, and practices.

Activity: Asking and Answering Questions. *“If you broadcast a question, you’re likely to get interesting answers or opinions that you may not have thought of.” [P9]*

Participants said that they ask questions on Twitter, and some reported that this helped them understand and solve their problems better—with only one interviewee being really successful at it. Most interviewees mentioned that they do not have enough followers to actually receive any answers. For those with a smaller audience, the public questions and

answers of others were interesting as this exposed them to diverse opinions. While several developers saw answering questions as an opportunity for everyone involved to learn something new, others simply wanted to be nice when they answered questions from others.

Activity: Following Experts. “[...] because he’s a respected person in the industry, as far as Twitter goes, when he posts something like “you shouldn’t do this” or “you should do this” or “this is interesting”, I value that more because I don’t need to figure out what’s the value of someone’s tweet, because I can assume that it’s better or it’s a high value tweet.” [P15]

Following experts helps developers learn and tap into experiences they might otherwise not have access to. Developers use a person’s status to judge the value of their tweets. Following experienced developers further provided an opportunity to learn about the behaviors of successful people, and thus, about the culture in certain projects and communities. Five interviewees told us about this.

Activity: Participating in Conversations. “It’s much easier to learn about new things when you’re part of the conversation about it. [...] It’s easier because you listen to how it’s made, you listen to why they did some things the way they did.” [P13]

Twitter allows developers be part of conversations between the people who work on things developers use. Being part of these conversations makes learning easier as it lets them gradually digest new information and better understand the rationale behind decisions. Two interviewees also mentioned gaining insights into what happens internally by following high-profile developers in projects and companies.

Impact: Learning As Investment. “I think the learning aspect is [...] the greatest value I get from it.” [P2]

Eight interviewees stressed that Twitter provides access to diverse resources, and that they value this diversity. They said that learning diverse things helps them become broader developers and gives them a competitive advantage. In the validation survey, we asked participants whether Twitter exposes them to technologies and practices they should learn in software development: 62% (421) of respondents agreed. When asked whether Twitter had helped them become a better programmer, 219 developers (32%) agreed, while 38% disagreed.

Impact: Learning Serendipitously. “I think on Twitter there is the possibility for me to learn things I’m not looking for.” [P11]

Twitter allows developers to learn things in an undirected, serendipitous manner. A software architect (P19) told us that this learning mode is valuable when he needs to advise developers on his team: when a colleague approaches him with a problem, he sometimes does not need to search for an answer, as he had heard about a suitable approach on Twitter without having looked for it. Social signals such as status and approval from others helps surface what is valuable, saving developers time. We talked to six interviewees who told us about this effect. In the validation survey (cf. Fig. 1), the majority agreed that Twitter helps them learn things they were not actively looking for.

4.2.1 Summary

Developers use Twitter to ask and answer questions. They follow experts, which lets them participate in conversations that provide them with expertise that would usually be hard

to access. They see constant learning as an investment into their careers and like the serendipitous learning that Twitter facilitates.

4.3 RQ 3: How does Twitter nurture relationships between developers?

Research question 3 explores how Twitter helps developers form and maintain relationships. We find that Twitter can foster the development of larger communities, but also support relationships between distant colleagues by helping create trust and rapport. It can also create collaborations between random strangers.

Activity: Managing One’s Image. “So I guess if I start talking to someone on Twitter, it lets them know who you are as well and lets you build a bit of a personality about you rather than just being like another Twitter handle.” [P18]

Four interviewees thought that connecting with others on Twitter may be influenced by the image one projects. They deliberately think about how to create a personality on Twitter so that people can more easily assess what kind of person they are communicating with. Two interviewees strategically searched Twitter for mentions of themselves or their content, seeing these as opportunities to follow-up with interested others. Apart from being enjoyable, developers also said that this would help them build their online image, possibly improving their career opportunities in the future. In the validation survey, over half of the respondents agreed that Twitter helps them manage their image as a developer.

Activity: Building Community. “[...] we need a lot of people to use it and start sending back patches, feedback, reporting bugs, testing the whole thing and that’s what open source is all about, it’s humanity. [...] I see my role as being a motivator for this whole thing. [...] People need to talk in order to get the great ideas.” [P1]

Twitter helps build communities around open source projects and interest groups. Key individuals from these projects use Twitter to engage existing and new community members. Some developers also used Twitter for business networking. Twitter can be an entry point for more isolated developers and they value it for providing them with a window into the world.

Hashtags were used for conferences or specific topics of interest to create connections within a community. One interviewee told us about the #pairwithme hashtag on Twitter: developers use it to find others who are interested in pair programming remotely. The goal is for developers to pair regularly with strangers to learn something new. Three interviewees told us about Twitter’s role in building communities. In the validation survey, 44% agreed that Twitter helped them in building communities. Only one third of the respondents (222) found it helpful for business networking.

Impact: Discovery of Interesting Developers. “I identify with that community and if I find a Rubyist who is also like [...] a hacker / maker type of person [...] then that’ll be like a much stronger case that I should follow them as well.” [P5]

Developers use Twitter to find interesting strangers. They decide whom to follow based on whether they work in a similar niche or seem to have similar technology-related values. For discovery, interviewees and the initial survey respondents both use Twitter and Google+, preferring Twitter for finding strangers and Google+ for keeping up with people they already know. Three interviewees discussed these is-

sues with us. In our validation survey, 69% of professional developers agreed that Twitter helps them discover interesting software developers.

Impact: Achieving Trust and Rapport. “You wouldn’t have to break the ice and could just be sort of friendly and could get down to business. Rather than like making small talk, trying to get to know each other, that kind of thing.” [P8]

Work relationships involve building trust and rapport. Twitter helps developers connect with remote colleagues, either working on the same project or with colleagues “in the broader open source sense.” [P9] Some interviewees noted that contact through Twitter helped break the ice in meetings. Twitter wasn’t the only communications medium they had used. However, they attributed the change in the relationships at least partially to the service. Three of our interviewees claimed that the increase in trust improved their collaboration: “I think it made collaboration easier because you could maintain a friendship on Twitter.” [P8] In the validation survey, over half of the respondents—346 developers—agreed that it was helpful in building trust and rapport.

Impact: Work Validation. “So it meant that people were actually looking for the information I was providing. [...] I always got these favorites and these retweets, and then I got followers on GitHub on the project.” [P15]

Some developers see Twitter as a backchannel that provides them with validation for what they produce. The three interviewees who mentioned this were excited about positive feedback received through favorited tweets or retweets. Developers were especially enthusiastic about validation from well-known developers and projects. They said that it showed them that what they produced was valuable enough to reach “important people”. In our validation survey, 37% of developers agreed that Twitter provides validation of their work.

Impact: Feedback on Projects. “It’s a very kind of public “hey, good job.” And I think that ... at least from my position, I prefer to have someone say hey, I really like your thing in a public space, because it could encourage other people to check it out.” [P4]

Six interviewees said that Twitter was a good channel to give public feedback on projects. They regarded public compliments as more valuable than private ones. Even though blog posts often include a commenting facility, some participants were more comfortable with discussing a post on Twitter. P24 commented that he was more confident commenting on Twitter than responding to a blog post, as the latter may seem like they were “attacking somebody”. In the validation survey, 45% of the respondents agreed that Twitter helps get feedback on projects.

Impact: Formation of Communities of Practice. “Quite literally, every single person I follow is a software developer and all of my followers are software developers too, so that’s pretty neat. I feel more of a community than I feel on Facebook.” [P6]

Twitter helps developers find out where a community is moving and what people are excited about: “Without Twitter, I wouldn’t find out about all this new stuff, you know. That’s how I communicate with everybody.” [P18] Six interviewees discussed this with us. Some developers found it challenging to stay connected to a niche community when their day-to-day work was in another niche. They then followed a core group on Twitter, which then relayed what was going on in the community.

Impact: Job Opportunities. “Indirectly, I ended up in this job through Twitter. By getting to know some of the other developers in Vancouver and knowing who is hiring and things like that.” [P9]

Three interviewees reported that Twitter let them access new work opportunities. We saw three different ways in which Twitter was used for this purpose—some of them reported by multiple interviewees. Firstly, initial contact was made through short conversations on Twitter, but then switched to a more private channel. In some cases, such discussions lead to new collaborations or jobs. The second kind occurred when retweets carried a message across the boundaries of a social network. Finally, the third kind of opportunity were based on business networking. Developers focused, for example, on a certain technical niche or city. This allowed them to make connections that later led to employment. In the validation survey, 28% of respondents agreed that Twitter is helpful in finding job opportunities.

4.3.1 Summary

Developers on Twitter manage their public images. This helps them discover interesting developers to connect with, and for existing relationships, to build trust and rapport. The service provides some rudimentary validation of one’s work, but we also heard of more elaborate feedback being exchanged. These activities could also have an impact on job opportunities. Developers also use Twitter to build communities around the things that interest them.

4.4 RQ 4: What are challenges faced by software developers using Twitter, and how do they cope with them?

In research question 4, we investigated the challenges of using Twitter as a developer, and the strategies used to cope with them. We found two major challenges: building and maintaining a relevant network, and consuming content efficiently. We now discuss each challenge followed by the coping strategies our participants used to address them.

Challenge: Maintaining a Relevant Network.

“When you follow 1000 accounts, many things you see are just not for you.” [P16]

Getting value out of Twitter is challenging: users need to curate their networks. The service might otherwise become irrelevant to them, especially when their networks get too big. Issues also occur when a developer moves from one niche to another: they might need to start over to build a more relevant network.

Strategy: Following Relevant Developers. Twelve interviewees who were active Twitter users reported that they find key developers from the niche they are interested in and follow them on Twitter. They then try to find similarly interesting people mentioned by these key developers. P5 even discussed looking at source code before deciding to follow someone. Others used real-life recommendations from colleagues and others to find developers to follow. Finally, some used location to determine relevance. Developers from areas with lower Twitter adoption struggled to find local peers. Twitter’s profile features were also used to assess others: the “bio”—a short description in user profiles—was used to form first impressions. The number of followers of a had as well as the ratio of users someone follows vs. the number of users following them also influenced others: the

lower the ratio, the more important a user seemed to be. Finally, developers assessed a user’s tweet volume to decide whether they’d be able to keep up.

Strategy: Unfollowing Developers on Twitter. Network curation is continuously necessary, as a developer’s interests and those of the people they follow may change. Thus, the decision to follow someone on Twitter is not final: following someone was often seen as a *trial* of whether their content was relevant and whether the volume was manageable. Some interviewees had a routine for purging their following list: they go through their list once in a while and remove accounts no longer worth following. We talked to 11 interviewees who told us about this strategy.

Challenge: Consuming Content.

Developers use several strategies to decide whom to follow and to manage their networks. However, content consumption was also a challenge, as the volume of tweets could become overwhelming. Developers adopted four main **strategies** for consumption:

Strategy: Filtering. “I’m using TweetDeck with some filters. I think Twitter could provide that. I filter every post with [keyword], for example.” [P23]

Strategy: Skimming by Profile Pictures. “I just notice when I skim across their display picture, I go “oh, I wonder what they’re saying” [...] They catch more eyes more than others.” [P18]

Strategy: Skimming Often, Reading Later. “If there’s a link in there that I feel I need to read, I don’t read it at the time, I save it to Pocket. And then I’ll read that later.” [P27]

Strategy: Reading Routine. “If I checked it obsessively, I’d be much less productive. [...] I tend to do all that kind of stuff in batches, like at the end of the day.” [P10]

4.4.1 Summary

Developers on Twitter struggle with two main challenges. Firstly, they have to constantly maintain their networks to keep it relevant. They use technical niche, status, and location to discover potentially interesting users and then use different signals to assess them. Unfollowing others regularly is part of network maintenance. Secondly, consuming lots of content poses a challenge. While the character limit helps, developers also use other strategies—filtering, skimming, saving links to read them later—to manage their consumption.

4.5 RQ 5: What are reasons for non-adoption of Twitter by software developers?

Research question 5 investigated why developers do not adopt Twitter for their work. We find they avoid using Twitter in face of the potential waste of their time and effort. They value conversations with other developers and use other tools to fulfill this need. We did a quantitative analysis of non-adoption patterns in the exploratory survey, as their replies were unambiguous and thus easy to code consistently. We report these numbers as well as quotes from the interviews (indicated by P#) and the exploratory survey (E#).

Too Much Noise. Some developers (35 exploratory survey respondents, 5 interviewees) felt **drowning in the noise** caused by too much information on Twitter. They use other tools to stay updated and build networks. P14 builds his net-

work on Google+, with a preference for the *circles* feature. Likewise, P4 commented: “On G+ you have that control, you can say I wanna see more from this person, I wanna see less from this person.”

140-Character Constraint. Whereas many adopters appreciated the concise tweets, nine respondents from our exploratory survey mentioned that they did not like this limitation. This limitation also has many users shorten Web links—three respondents complained about this practice: “Twitter messages are too short to contain anything useful with context.” [E29]

Poor Support for Conversations. Twitter was not designed for long conversations. As interviewees told us, multiple conversation threads quickly become unmanageable. Additionally, conversations are quite ephemeral, as tweets may only be accessible for a few days. Some study participants reported that they preferred to move conversations from Twitter to a more suitable medium such as email or IRC. Three interviewees and six exploratory survey respondents mentioned having had problems with Twitter’s **lack of support for conversations**. Some mentioned a preference for GitHub and Google+, which provide better support for conversations.

Unsure of Benefits. Finally, 22 exploratory survey respondents said they did not use Twitter because they did not have a reason or did not understand why they should use it: “I don’t understand it and I don’t see any purpose for it.” [E43]

4.5.1 Summary

We found several reasons for non-adoption of Twitter by developers. Too much noise on the service makes it hard to manage the volume of tweets. Others did not like the 140-character constraint as it could lead to misunderstandings more easily. Relatedly, Twitter’s poor support for conversations is another reason why some developers said they do not use Twitter. Finally, some were unsure of the benefits the service could provide them.

5. DISCUSSION

The insights provided by the non-adopters and infrequent users are valuable for understanding the limitations of Twitter in software engineering. A large number of participants did not fully understand how Twitter could play a role in their professional activities. This is not surprising: as we learned from the adopters, using Twitter successfully requires several strategies. Likewise, many respondents were concerned that Twitter would be a distraction or would waste their time. These are noteworthy concerns as some that had used Twitter previously stopped for these reasons. Again, adopters used strategies to curate content and reduce distractions.

Many participants also provided insights on how other social media tools provide the benefits that Twitter brings. In particular, several participants appreciated that Google+ provides more context and better supports conversations. Twitter non-adopters use news aggregators such as *Hacker News* to curate technology updates. However, Twitter adopters said that Twitter lets them add yet another layer of content curation from such sites.

More subtle reasons for both adoption and non-adoption of Twitter were related to the peculiarities of the tools. Each social media service supports its own set of such charac-

teristic features: 140 characters per post for Twitter, circles on Google+, issues on GitHub. Their special roles and the conflicting opinions among our study participants became more apparent as we investigated the interplay between these tools.

5.1 Contributions

Previous work investigated the role of Twitter in conversations [7, 6, 3] and information dissemination [7]. Others explored the role of microblogging in increasing awareness and connectedness among co-workers in organizational settings [20]. To the best of our knowledge, our study is the first in-depth investigation with a diverse population of developers. By including the perspectives of non-adopters, we contrast the viewpoints from developers who have adopted Twitter with the ones who have not. We also highlight the interplay between Twitter and other social media.

Software developers work in a rapidly-evolving field where staying current is a requirement and a challenge. They use social media to form and maintain relationships with co-workers within the same organizational context, but also to connect with and stay aware of other developers around the globe.

5.2 Implications

This section discusses some implications our study might have on research, software development organizations, and individual software developers, respectively.

5.2.1 Research

Twitter is used by many to keep up to date and stay connected in software engineering, but is it the best tool for doing so? Non-adopters mentioned a number of limitations and barriers. Microblogging is likely to play a more important role in software engineering. More research is needed to understand why it is used, what benefits it brings, and how tools and practices around it can be improved.

Opportunities for future work include an investigation of how Twitter impacts the diffusion of innovations in software engineering. Social media help disseminate knowledge about technologies and practices, but too little is known about how this takes place among software developers. Quantitative studies are needed to determine the prevalence of our findings in practice, and future qualitative studies should investigate alternative populations, such as corporate settings or non-Western societies.

Software developers use a multitude of channels to communicate and collaborate, with both strong and weak ties. However, our study only considered Twitter as *one* of many channels. Future research should investigate the interplay of several different channels and which developers use which tools in their work. One study participant told us about his idea that development tools are generational and that there are many communities that each uses their own subset of the available tools and channels. Ideally, we would like to see a map of the kinds of developers there are today, their values, the practices they use, and the tools they use to create software, collaborate with each other, and communicate.

We provided some details about the population we surveyed in our study, but plan to conduct a more in-depth analysis of our participants' characteristics, demographics, and work contexts. Relatedly, we are planning to analyze the validation survey answers in light of such demographic

data—do Web developers tend to agree to different statements than low-level systems programmers? Do users of statically typed languages use social media and microblogging in different ways than those of dynamically typed languages?

In our study, it became clear that Twitter only provides value with the right network. At the same time, it is currently the most popular microblogging platform. Thus, thought leaders and key people from crucial projects can be found on Twitter instead of alternative platforms. This also means that Twitter—the company—has control over an important aspect of software development. Future work should look into the consequences of this dependency, the relevance of alternative microblogging platforms, and other crucial services with similar monopolies—e.g., Stack Overflow for programming questions, GitHub for code hosting, or Google for Web search. Can and should research strive to provide alternative platforms for these services so developers are less dependent on commercial entities?

5.2.2 Practice

Social media in general and Twitter in particular are not only interesting for research, but—as our study has shown—can provide developers and their organizations with strategic advantages.

Organizations. Twitter lets developers find out about technologies and practices that might improve their efficiency and effectiveness, as well as software quality. Such innovations are talked about while they are still being created. Organizations could use the availability of such information to their advantage. However, not every employee might want to use Twitter themselves. This might not be necessary, though: the software consultancy Thoughtworks has a small group of employees that regularly creates a technology radar and also provides a white paper that may help others create their own technology radars⁵.

Employees monitoring Twitter, social media, and potentially other sources to create such a technology radar would need to be comfortable using social media, and be able to judge the merit of new, potentially hyped technologies and practices. Thus, people filling such a role will need to be innovative and interested in new developments, but at the same time, will need to be reflective, knowledgeable about the past of software development, and possess a healthy skepticism. In addition, explicit engagement with external developers on Twitter might help an organization shape the direction of certain practices and technologies that may or may not become crucial knowledge in the future.

Practitioners. From our study, individual developers can derive a set of recommendations on how to use Twitter, and possibly also related tools, to their advantage, improving their productivity and furthering their careers. In the following, we summarize what we now believe to be appropriate strategies for using Twitter as a software developer.

Follow niche leaders: A few thought leaders in one's technological niche should be the first people to follow. These could be core developers of relevant projects, community organizers, or standards writers. Where appropriate, engage these people with questions. As elsewhere on the Internet, it is good practice to research a problem before asking a question.

⁵http://nealford.com/memeagora/2013/05/28/build_your_own_technology_radar.html

Build a network organically: Consider following some of the users the above users follow, retweet, and talk to.

Avoid looking like a spammer: Have a profile picture that is not the default picture. One does not have to reveal one’s identity, but it seems helpful to appear like an authentic person. Related to this, users should watch the ratio of people who follow them vs. those they follow, as this ratio is usually a good indicator of spam bots and promotional accounts. Ideally, one follows only a few relevant developers that one can keep up with. Looking like a spammer decreases the probability that real people will follow someone, thus hurting that account’s accumulation of weak ties—a core value of public microblogging.

Following incurs a cost: Every account followed adds more tweets to one’s timeline, which then takes more time to read. Thus, following should be on a trial basis only, with accounts being pruned from time to time.

Share what you learn: While we found that consuming content on Twitter already has several benefits, posting and interacting have advantages as well. On the one hand, posting interesting content will attract followers, which increases the number of one’s weak ties. On the other hand, interactions are a way to strengthen these ties and to learn from others. As we saw, Twitter can be used as a space for collaborative learning—the logical consequence, then, is to tweet about the things one learns, e.g., by writing short blog posts about problems one solves in their day-to-day work or side projects. While some things may seem trivial to post about, there is a high probability that *someone* will find that information useful. A low effort variation of this could be to tweet a single new lesson one learned at the end of the day. This would have the added benefit of having to reflect on what one achieved that day—a habit shown to be motivating [1].

Use Twitter for short conversations: We saw that Twitter’s constraints make longer discussions unfeasible. Thus, one should switch to more suitable channels, such as email or chat, for more elaborate arguments. The short format also increases the potential for misunderstandings, so one should be careful—good manners are important on the Internet, but much more so in such a constrained format.

For questions, use hashtags: Study participants complained that they cannot ask questions on Twitter because they do not have enough followers to receive any answers. As Wang et al. [19] have shown for the Drupal framework, however, some development communities actively monitor hashtags on Twitter to engage new users and contributors. Using a technology-related hashtag in questions should thus increase the probability of receiving answers to one’s questions.

Finally, we suspect that these same strategies could also be helpful for software engineering researchers. Twitter—or an alternative service—might help us learn from and connect with each other, enable faster iterations on our research, and provide us with faster access to current tools and practices in software engineering research. It already helps a sizable subset of us connect at conferences.

5.3 Limitations

Because of the exploratory nature of this work, we chose Grounded Theory as our research method, and this has some implications regarding the limitations of our study. While we achieved saturation regarding the topics we focused on in our research, there are other populations that might add new insights. Findings from our study may not apply for every-

one. Concepts and themes that emerged from our analysis cannot be generalized.

We invited active, but random users of GitHub to our surveys and interviews. In all cases, participants were self selected: the population we collected data from was comprised of individuals who used GitHub, had time and motivation to answer our survey questions, and in some cases, to be interviewed. These were often from Western countries.

Choosing GitHub as a source of participants introduced a bias towards this population into our study. Other populations might have different characteristics—they might use Twitter less or other tools more, might not be allowed to use microblogging in their jobs, or have vastly different demographics. It is currently unknown what kinds of developers use GitHub exactly, but it seems likely that they are not reluctant to use social media. Future work should collect data about the demographics of GitHub users to better inform this, as well as past and future studies.

Our quantitative validation survey has similar limitations regarding its external validity: participants were self-selected, but randomly chosen users of GitHub. We created the statements in the survey from our findings and used Likert-type scales, an accepted instrument to assess agreement.

The majority of our study participants were software developers—many of them working professionals. Our validation survey found high agreement for several of our findings. Therefore, we believe that even in the light of these limitations, we have uncovered valuable insights regarding software developers’ use of Twitter.

Yet, how transferrable are our findings to other microblogging platforms? Our study has shown that the value of Twitter lies in its wide adoption and the network each user can build for themselves. Since Twitter’s adoption—especially by key developers of important open source projects—seems to have such a strong impact, we believe our findings may not necessarily translate to other microblogging platforms or even microblogging in general.

6. CONCLUSIONS

We reported on the first in-depth qualitative study of how some software developers use Twitter, why some developers do not use it, and how these non-adopters fulfill the needs that would otherwise be catered to by Twitter.

In our analysis, we extracted themes that allowed us to explain the value Twitter can provide to software developers that need to stay current in an accelerating field. We learned about the challenges they encounter, the strategies they use to cope with them, and why some developers do not adopt Twitter. A survey validated many of our findings.

These results help us understand how developers keep up in their field, learn, and connect with others by taking part in software development communities, following thought leaders, and seeking encounters with like-minded strangers. The usage strategies we discovered can now inform individuals and organizations that need to stay current in and connected with their professional communities.

7. ACKNOWLEDGMENTS

We thank Cassandra Petrachenko for her dedicated support in improving this report. We are deeply grateful to the participants of our surveys and interviews. We thank our reviewers for their constructive comments.

8. REFERENCES

- [1] T. Amabile and S. Kramer. *The progress principle: Using small wins to ignite joy, engagement, and creativity at work*. Harvard Business Press, 2011.
- [2] G. Bougie, J. Starke, M.-A. Storey, and D. M. German. Towards understanding twitter use in software engineering: preliminary findings, ongoing challenges and future questions. In *Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering, Web2SE '11*, pages 31–36, New York, NY, USA, 2011. ACM.
- [3] D. boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10, 2010.
- [4] J. Corbin and A. Strauss. *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage Publications, 3rd edition, 2008.
- [5] R. Hoda, J. Noble, and S. Marshall. Developing a grounded theory to explain the practices of self-organizing agile teams. *Empirical Software Engineering*, 17(6):609–639, 2012.
- [6] C. Honey and S. Herring. Beyond microblogging: Conversation and collaboration via twitter. In *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, pages 1–10, 2009.
- [7] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, WebKDD/SNA-KDD '07*, pages 56–65, New York, NY, USA, 2007. ACM.
- [8] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web, WWW '10*, pages 591–600, New York, NY, USA, 2010. ACM.
- [9] L. Mamykina, B. Manoin, M. Mittal, G. Hripcsak, 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, CHI '11*, pages 2857–2866, New York, NY, USA, 2011. ACM.
- [10] A. E. Marwick and d. boyd. I tweet honestly, i tweet passionately: Twitter users, context collapse, and the imagined audience. *New Media & Society*, 2010.
- [11] P. McFedries. Technically speaking: All a-twitter. *Spectrum, IEEE*, 44(10):84–84, 2007.
- [12] D. Pagano and W. Maalej. How do developers blog?: an exploratory study. In *Proceedings of the 8th Working Conference on Mining Software Repositories, MSR '11*, pages 123–132, New York, NY, USA, 2011. ACM.
- [13] R. Pham, L. Singer, O. Liskin, F. Figueira Filho, and K. Schneider. Creating a shared understanding of testing culture on a social coding site. In *Proceedings of the 2013 International Conference on Software Engineering, ICSE '13*, pages 112–121, Piscataway, NJ, USA, 2013. IEEE Press.
- [14] P. Prasetyo, D. Lo, P. Achananuparp, Y. Tian, and E.-P. Lim. Automatic classification of software related microblogs. In *Software Maintenance (ICSM), 2012 28th IEEE International Conference on*, pages 596–599, 2012.
- [15] P. C. Rigby and M.-A. Storey. Understanding broadcast based peer review on open source software projects. In *Proceedings of the 33rd International Conference on Software Engineering, ICSE '11*, pages 541–550, New York, NY, USA, 2011. ACM.
- [16] L. Singer, F. Figueira Filho, and M.-A. Storey. How Developers Stay Current Using Twitter: Supplemental Materials. Technical Report DCS-353-IR, University of Victoria, Victoria, BC, Canada, 2014.
- [17] L. Singer, F. F. 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 ACM 2013 conference on Computer Supported Cooperative Work and Social Computing, CSCW '13*, New York, NY, USA, 2013. ACM.
- [18] Y. Tian, P. Achananuparp, I. Lubis, D. Lo, and E. Lim. What does software engineering community microblog about? In *Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on*, pages 247–250, 2012.
- [19] X. Wang, I. Kuzmickaja, K. Stol, P. Abrahamsson, and B. Fitzgerald. Microblogging in open source software development: The case of drupal using twitter, 2013.
- [20] J. Zhang, Y. Qu, J. Cody, and Y. Wu. A case study of micro-blogging in the enterprise: use, value, and related issues. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10*, pages 123–132, New York, NY, USA, 2010. ACM.
- [21] D. Zhao and M. B. Rosson. How and why people twitter: the role that micro-blogging plays in informal communication at work. In *Proceedings of the ACM 2009 international conference on Supporting group work, GROUP '09*, pages 243–252, New York, NY, USA, 2009. ACM.