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

Article *in* Information Security Technical Report · November 2013

DOI: 10.1145/2568225.2568305

Leif Singer
University of Victoria

Fernando Figueira Filho
Universidade Federal do Rio Grande do Norte

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

Each day, more than 500 million users post over 500 million tweets through the Twitter microblogging service. Research has established that many software developers use Twitter in their work, but how or why they use it has not been analyzed. Understanding how developers use Twitter could lead to improved tool and process support, and learning more about the reasons for non-adoption could inform the design of better tools and services.

In a qualitative study, we surveyed 271 and interviewed 27 developers that are active on GitHub, a prominent code sharing site. We found that Twitter helps developers keep up with the fast-paced development landscape; they use it to stay aware of industry changes, for learning, and for building relationships. We discovered the challenges they experience and extract their coping strategies. Some developers do not want to or cannot embrace Twitter for their work—we discuss their reasons and alternative channels. We validated our findings in a followup survey with 1,413 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; software systems, programming languages, devices, and tools are experiencing an ever-increasing rate of change. 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.

For example, email is used to deliver commit messages and to support code reviews [13], IRC is used during co-development activities, and software forges support community development [11]. Many developers blog about their experiences and disseminate them among their readers [10]. Stack Overflow helps developers access a crowd of experts willing to help them with their challenges [7]. Social media is widely used by some developers to connect and collaborate with one another [14]. Despite—or because of—the availability of these tools, it can be a challenge to keep up with and make sense of new information, tools, and practices. Yet, other communication tools are not yet well understood in the context of software engineering.

One of these tools is Twitter, the popular microblogging service. More than 500 million registered users[1] post over 500 million tweets a day[2]. Many software developers use it to communicate about software engineering topics [1, 15], but we do not know why some developers adopt it and fervently use it, while others do not and question its value. 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-adoption of Twitter could shape the design of improved tools or suggest alternatives 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, followed up with 27 interviews, and then hosted a validation survey with 1,413 GitHub users. We found that developers that have adopted Twitter use it to filter and curate the vast amount of information available to them as part of their programming activities. The benefits we discovered in our analysis 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.

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

University of Victoria Technical Report DCS-350-IR

[1] http://semiocast.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

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 from the exploratory survey and interviews. The validation survey for these findings is presented in Section 5. This is followed by a discussion of the findings, validation survey, main contributions, limitations of the work, and future work in Section 6. We then conclude the paper in Section 7.

2. RELATED WORK

Microblogging is the practice of posting short thoughts, ideas, and other content to the Web [9]. 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. 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. [5] analyzed the connections between users and the content of the messages they post. Their research identified 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 [4] investigated 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.

Boyd et al. [2] examined the phenomenon of *retweeting*, a convention that emerged among early Twitter users. A user who wants to pass along 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. found that retweeting could be classified as a conversational practice.

Marwick and Boyd [8] examined 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. [6] investigated social network topologies, influential users, and the content spreading behavior of 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. [17] investigated how employees of a large enterprise use Yammer and how its usage differs from Twitter. They found that employees use Yammer 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 [18] conducted interviews with Twitter users working in corporate settings. They found 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. [1] conducted quantitative and qualitative analyses of 11,679 tweets by developers. They found that software engineers' tweets involve conversation and information sharing more often than comparable studies examining more general populations. In their qualitative analysis of a 600-tweet sample, they identified four developer-specific categories of content.

Tian et al. [15] chose a random sample of tweets containing a set of potentially development-related hashtags, such as #scrum, #java, or #testing. The authors then determined content categories for a subset of these tweets, determined their popularity, and analyzed which categories were retweeted the most. Relatedly, Prasetyo et al. [12] automatically classified tweets regarding their relevance to software engineering.

Wang et al. [16] analyzed 568 tweets from 24 Twitter accounts associated with the Drupal open source content management system. The authors found 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 out to understand *why* and *how* software developers may (or may not) use Twitter. The following section details our study design.

3. STUDY DESIGN

We conducted a Grounded Theory [3]-based study consisting of three phases of data collection (exploratory survey, interviews, validation survey) and iterative phases of analysis. First, we sent an online survey of open questions to 1,160 GitHub[3] users[4]. In this exploratory survey, we asked users about their reasons for reading and posting on Twitter, benefits and challenges, as well as the process for discovering and following other users. We also included a section targeting non-adopters, asking them about their reasons for not using the service. We received 271 responses to the exploratory survey.

In the second phase of our study, we interviewed 27 of the survey respondents who had volunteered to speak with us. Prior to the interview phase, we analyzed the exploratory survey responses and used them as a guide for the semi-structured interviews.

During our analyses of the survey and interview 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 provides *value* to some software developers, yet other developers constantly face *challenges* using the tool; 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 used axial coding, iterating through our exploratory survey responses and interview transcripts, to answer

[3] `https://github.com`

[4] Selected from GitHub's public event stream in May 2013, choosing recently active users with public email addresses.

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[5], receiving 1,413 responses.

3.1 Research Questions

The first three research questions are focused on the *value* Twitter can provide to software developers. From our analysis, we found three dimensions in which Twitter can provide value to developers: *awareness*, *knowledge*, and *relationships*. 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*, and *building trust and rapport*:

> ***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 answers from the exploratory survey stated that the respondent was not using Twitter. We believed it was important to understand the perspectives of *non-adopters* as well:

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

3.2 Participants

Our 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 a broader population of software developers, some of which might be using Twitter. Such a population should provide us with a more diverse picture of Twitter use among software developers than simply targeting Twitter users alone.

To characterize the participants of our exploratory survey and interviews, we downloaded their Twitter and GitHub account details, where available. The exploratory survey had 271 survey respondents (response rate: 23%), and from these, we obtained 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. 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. The validation survey had 1,413 survey respondents (response rate: 14.1%), and from these, we obtained the details for 958 Twitter accounts and for 1,381 GitHub accounts.

3.2.1 Exploratory Survey Respondents

270 of the 271 survey respondents said they develop software, and 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, 25% were from Europe, and for 24%, we were unable to find their location.

Twitter 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, and 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, 30% were from Europe, and 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 and 18 said they were not. Account ages ranged from 5 months to 5 years and 7 months.

3.2.3 Validation Survey Respondents

1,413 GitHub users answered our validation survey. 1,412 of them said they develop software, and of these, 1,145 were professional developers (81%). 814 developers said they worked on private projects, and 743 said they worked on one or more open source projects. 940 respondents said they use Twitter at least once a week (67%). 958 respondents included their Twitter username.

For these Twitter accounts, the account ages ranged from 1 month to 7 years and 2 months. The number of tweets posted per account ranged from 0 to 155,210. The number of followers ranged from 0 to 31,678, with a median of 154 (average: 518). The number of accounts people followed ranged from 0 to 4,056. 786 respondents indicated that they used Twitter for their development-related work.

1,381 survey respondents included their GitHub username. 482 respondents said they were for hire and 899 said they were not. Account ages ranged from 3 days to 5 years and 9 months.

[5] Obtained from the same dataset that the exploratory survey used.

When reporting results from our validation survey, we only include the 1,412 respondents who indicated that they develop software. Among these, we distinguish between adopters (940 respondents—those who use Twitter at least once a week) and non-adopters (473 respondents) of Twitter.

3.2.4 Summary

Our population consisted of mostly professional software developers—82% in the exploratory survey, 93% in the interviews, and 81% 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, and 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 projects. 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. Across surveys, GitHub users' numbers of followers ranged widely from 0 to 1,321. On Twitter, that value ranged from 0 to 31,678. Similarly, our participants weren't exclusively veterans or newcomers to the services: account ages ranged from 3 days to 5 years and 9 months for GitHub, and from 1 month to 7 years and 2 months for Twitter.

4. FINDINGS

This section reveals the main themes we found in the first two phases of our study, organized according to each research question. To illustrate the different aspects of each theme, we provide a selection of quotes from the exploratory survey and interviews. Participants are identified using the following conventions: E# for the exploratory survey and P# for the interviews. The succeeding section reports on the results from our validation survey.

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

We found several themes that relate to developer awareness. We categorize these themes 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. There are people developing libraries and I can communicate with them through Twitter." [P11]*

Many of our initial survey and interview respondents said they follow specific developers and projects, such as frameworks or libraries, that are relevant to their work. This results in a direct channel to the developers that create the code they use. This is beneficial because it enables direct conversations with developers who can provide information that would have been otherwise cumbersome to obtain, such as surfacing security issues or estimated release dates. Several survey respondents and interviewees mentioned that Twitter was a richer alternative to RSS, and that the social side of Twitter—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 (i.e., curate) posts from news sites, blogs, and other Twitter posts. Several of the study participants found that following these individuals provided them with important updates without having to weed through content themselves—developers followed people whose judgement they trust. Over time, they remembered accounts who had previously posted interesting information and used that knowledge to decide which new content would be worth reading. The associations with a Twitter user would effectively elevate the content posted by trusted users above the content posted on more generic technology news sites.

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 [...] so I follow a lot of other programmers that I think are pretty awesome and usually have interesting things to say that I would benefit from." [P27]*

Developers follow certain thought leaders in their respective niches to keep on top of what that community is talking about. These leaders are said to *"[shape] the community"* [P24] by pushing what is relevant to their followers. Participants were not only interested in the updates to the projects created by those they followed, but also in which technologies these people were using themselves, and which new technologies were available. P24 noted that the presence of these leaders may, however, result in a community that is confined to the perspectives of a few individuals who define what is interesting.

Activity: Promoting Project Activities.

> *"This is an OSS I'm currently working on, and I've been working on it for 3 years [...] I try to promote it [...] 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 that they think others should be aware of. Some use Twitter to promote their own projects or to broadcast important updates. This helps them get feedback from a larger user base and attract contributors. But it might also help them career-wise, as popular projects can be part of developers' personal branding efforts (cf. *Managing One's Image* in section 4.3).

However, their promotion is not restricted to actual code. Developers also promote their blog posts to get the word out. Similarly, developers use Twitter to promote events to local and remote participants, hoping to reach a more diverse audience.

Promoting and advocating one's projects and practices on Twitter can also serve a very strategic role; it can help technologies become more popular, which could increase market demand for expertise in the technology.

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]*

The developers we heard from work in a rapidly evolving industry. To them, this means they have to adapt frequently. They use Twitter in an effort to stay current with bleeding edge technologies, projects and tools, and to stay up to date with ongoing changes. One advantage of Twitter that participants mentioned regarding staying current in their field was that it allowed them to access new tools and practices as they became available. This helps them become familiar with technologies that might become popular, at their own pace, and with less time pressure. As some interviewees mentioned, such technologies have the potential to make developers more productive and, in turn, more competitive.

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 talk 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 several different ways.

Participants said they would get information about technical niches that are interesting and relevant to them, but that would not otherwise come up in their own news resources. They increase their awareness of a global population of software developers who are willing to share their experiences. By doing so, they are exposed to new perspectives, something which is very important in a quickly changing environment like software engineering.

They also gain access to a broader set of topics not limited to code-related resources in their own technological niche, such as business advice or ethical issues. By being exposed to and reconciling diverse perspectives, developers get to think more broadly about software development. Respondents also reported that Twitter amplifies other channels, such as blogs or conferences.

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]*

Several developers we interviewed feel a need to share solutions with others. They believe this will spare others from having to go through the same discovery process. However, disseminating knowledge isn't restricted to posting to Twitter. Developers also take advantage of a myriad of tools, such as Skype chats or IRC channels, to transfer the resources they find on Twitter. Some developers further act as information brokers using different channels as needed. For example, P20 describes himself as an *"in-between guy"* when using these different channels.

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 consciously use social media (in general) and Twitter (in particular) to push technologies and practices they are passionate about. They want certain technologies to succeed because they believe in them or because they have an investment in them. In other instances, developers suspect that whole communities were set up for promotion through social media by key people in the respective technology.

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

The second research question asks how Twitter supports developers' learning efforts. Our study revealed that developers value learning about new technologies and that Twitter allows them to be aware of what they need to learn (as discussed in the previous research question), but that it also plays a role in building concrete and tacit knowledge of technologies, tools, and processes. The themes that emerged during the analysis of this research question can also be categorized into the **activities** that Twitter users reported, as well as the **impact** these activities have on how they learn or support the learning of others.

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 in public, and some reported that the conversational nature of Twitter and access to diverse opinions helped them understand and solve their problems better. However, several developers mentioned that they did not have enough followers to actually receive any answers. For these, the public questions and answers of others were sometimes said to be interesting because the many answers provided would expose them to diverse opinions.

While several developers saw answering questions as an opportunity for everyone involved to learn something new, others answered questions because they wanted to be nice. For example, P2 said the following when we asked why he answers questions: *"Just to be a nice guy I guess. I don't have a good reason for it, and I don't expect anything from that. For that same reason I hang a lot in the PHP channel on IRC, to help people just because I can help people. It's not like I get a reward for that or something. It's a bit of a hobby." [P2]*

Activity: Following Experts.

> *"OK, well I think, 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]*

Just as following leaders plays a role in building and gaining awareness, following experts provides an opportunity for developers to learn and tap into experiences they might otherwise not have access to. Developers used their impression of an individual's status (number of followers, tweets, and other signals) to judge the value of insights posted by thought leaders. Following experienced developers further provided an opportunity to learn the behaviors of successful people, and thus, the culture in certain projects and communities.

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. You listen to it gradually." [P13]*

Twitter enables developers to be part of conversations between the people who work on things developers actually use. Being part of these conversations makes learning easier as it allows people to gradually digest new information and better understand the rationale behind decisions. Some participants also mentioned gaining insights into what happens in companies or projects by following high-profile developers. For example, P20 said that for him it was *"like being in the company and having a conversation around the coffee table about a piece of code that someone just wrote or new functionality that someone just added to something."*

Impact: Learning As Investment.

> *"I think the learning aspect is the most ... the greatest value I get from it. And that's it." [P2]*

Developers again stressed that the resources they get access to through Twitter can be diverse and that they greatly value this diversity. They claimed that learning more diverse things would help them become a broader software developer and give them a competitive advantage. As such, this mode of learning is a long-term investment into their careers and may lead to future employment.

Impact: Learning Serendipitously.

> *"I think on Twitter there is the possibility for me to learn things I'm not looking for.[...]Twitter allows for some sort of "casual" learning, things that are outside that bubble of things that I have to learn on demand." [P11]*

Twitter allows developers to learn in an undirected and serendipitous manner: they learn about things they were not actively looking for. A software architect (P19) told us that this learning mode is valuable when he needs to advise developers on his team. When approached with a problem, he sometimes did not need to search for the answer as he had already heard about a suitable approach on Twitter. Other participants noted that this mode of learning is not always immediately useful, but they would often make a mental note of and possibly bookmark things for future use.

The social approval of content from others helps determine what is valuable, saving developers time and effort otherwise spent on sifting through dead ends or less appropriate solutions. Some developers mentioned that Twitter helps them find and learn about things that they would not have been able to search for themselves, such as emerging technologies that are too new to appear in web searches. As P18 commented: *"They're just really cutting edge, but you won't know about them on Google so I use Twitter. You find the right people and you use it as a tool basically."*

Impact: Learning is Fun and Rewarding.

> *"It's just a lot fun! That's why I do it, it's a passion for learning, I guess. If I don't have anything to learn, I just get bored." [P2]*

Several developers stressed that they enjoy learning and liked that Twitter gave them access to a broad range of resources. Many of the developers mentioned that they used Twitter for fun and entertainment as well as for software development. When we dug into this area further, we discovered that many respondents also find learning about software development entertaining and intrinsically rewarding. A developer who states that she uses Twitter for entertainment may thus actually mean that she uses it for learning.

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

Research question 3 explores the ways in which Twitter helps developers form and maintain relationships with one another. We found that Twitter can foster the development of larger communities, but also support relationships between distant teammates by helping create trust and rapport. In addition, it can 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]*

Developers recognize that connecting with others on Twitter can be influenced by the image one projects on the site. They deliberately think about how to create a personality on Twitter so people can more easily assess what kind of person they are communicating with. They may not explicitly respond to other people's posts about their projects or blogs, but they do try to answer related questions that appear in tweets, especially if posed by influential people, as P27 mentions: *"Especially with people I think are influential. That way they think I'm cool."*

Some developers even strategically searched Twitter for mentions of themselves or their content, seeing these as an opportunity to follow-up with people who had seemed interested in them. Apart from being enjoyable, developers also said that this helps them build their online image, possibly improving their career opportunities in the future.

Activity: Building Community.

> *"So to get this technology into working status—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 local interest groups, such as meet-ups. Connecting with people who are interested in the same technological niche may help create opportunities to become more involved with the development of a certain technology. Key individuals from these projects actively use Twitter to strengthen the involvement of existing and new community members. Some developers were also very keen about using Twitter to connect with others for business networking.

Twitter can be an entry point for more isolated developers and they value Twitter for providing them with a window into the world. As P6 noted: *"You feel isolated to some extent, to be part of a community of software developers is always very neat."*

Another interesting activity that emerged was the judicious use of hashtags to create connections within a community; hashtags were used for conferences or specific topics of interest. One interviewee told us about the `#pairwithme` hashtag on Twitter. Developers use it to find others who would be interested in pair programming remotely. The goal is for developers to pair regularly with strangers to learn something new.

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, or you know, something else, then that'll be like a much stronger case that I should follow them as well." [P5]*

Developers reported that they use Twitter to meet new, interesting people. They would decide whether they were interested in a person based on whether they worked in a similar niche or seemed to exhibit similar values related to programming. For discovery of interesting people, interviewees and the initial survey respondents reported using both Twitter and Google+. However, many reported that they preferred Twitter for discovering interesting strangers, whereas they used Google+ for keeping up with people they already knew. For instance, P4 mentioned that discovery is harder on Google+.

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]*

An important aspect of strengthening relationships involves building trust and rapport over a distance. A few developers said that Twitter helped them connect with remote colleagues who were either working on the same project or with colleagues *"in the broader open source sense." [P9]* Some interviewees further noted that contact through Twitter helped to break the ice when meetings occurred. Twitter wasn't the only communications medium they had used (cf. section 5.5). However, they attributed the change in the personal relationships at least partially to the service. 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]*

Impact: Work Validation.

> *"So it meant that people were actually looking for the information I was providing. But then when I tweeted again and again and again, every week I tweeted once about new releases and new versions or questions or whatever, 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 the code and blog posts they produce. Interviewees who mentioned this were excited about positive feedback received by way of favorited tweets or retweets. Developers were especially enthusiastic about validation from well-known developers and projects. They said it showed them that what they produced was valuable enough to reach "important people". Developers also view Twitter as a channel to build their reputation.

Impact: Feedback on Projects.

> *"And at the same time it was also kind of me saying hey, I really like your project, have you ever thought about adding this? 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]*

Some developers said that Twitter was a good channel to give public feedback on projects, especially when updates are announced on Twitter or blogs, and that they valued public compliments more than private ones. This publicity could, in turn, increase interest in the project from other developers. Even though blog posts often include a commenting component, we heard from a few developers that some were more comfortable discussing a post on Twitter instead of directly in the blog post. P24 commented that he was more confident commenting in the *"Twitter universe"* as opposed to responding to a blog post as the latter may seem like he was *"attacking somebody"*.

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 allows developers to discover the direction a community is moving, what the present issues are, and what everyone is currently excited about. It was also seen by some as the main place for communicating with the communities that were important for developers' work. *"Without Twitter, I wouldn't find out about all this new stuff, you know. That's how I communicate with everybody." [P18]*

Twitter also helped certain developers stay connected to multiple communities. Some found it challenging to stay connected to a niche community when their day-to-day work is in another niche. The approach they followed in this case was to follow a core group of people from the community on Twitter as these people relayed what was going on and created a connection to the community as a whole.

Twitter also led to face-to-face connections. A few developers mentioned that when they had existing acquaintances in a city they were visiting, they might publicly ask on Twitter whether anyone would be up for dinner or drinks. They preferred the public format as that might result in the attendance of other interesting people.

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]*

Some interviewees reported that Twitter provided access to new work opportunities. We saw three different ways Twitter was used in job opportunities. Firstly, intial contact was made through short conversations on Twitter, but then switched to a more private channel suitable for longer conversations, such as email. In some cases, such discussions would lead to new collaborations, if not to jobs directly. The second way Twitter was used in job opportunities occurred when Twitter's diffusion through retweets carried a message across the boundaries of a social network. One example was a developer who found contract work through a series of retweets that at some point reached his Twitter feed. *"Someone was looking for someone who could do frontend work and that got retweeted and retweeted until it landed in my feed. I responded to that, and that's how I got a job a few times." [P2]* Finally, the third way Twitter was used in job opportunities was for business networking. Twitter allows developers to form new weak ties, and developers can, for example, choose to focus on a certain technical niche or city. This allowed some developers to make connections that later lead to employment.

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

This section discusses research question 4: the challenges of using Twitter as a software developer and the strategies developers use to cope with those challenges. We found two major challenges: building and maintaining a relevant network, and consuming content effectively. In the following, we discuss each challenge followed by the coping strategies our study participants used to address them.

Challenge: Maintaining a Relevant Network.

> *"When you follow 1,000 accounts, many things you see are just not for you." [P16]*

Getting value out of Twitter is a challenging endeavor: users need to carefully curate their networks. The service might otherwise become uninteresting and 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..

> *"I guess the major motive of me finding other people to follow is through expanding my existing network. So I follow person A, they follow person B. And working outward like that." [P9]*

Several developers who were actively using Twitter reported on the strategies they used to build up and extend their networks. They would find key developers from the niche they were interested it—for example, those contributing to central infrastructure or flagship projects—and follow them on Twitter. Reading these users' tweets, they would try to find similarly interesting people mentioned by these key developers and consider following them. Some would also look at what is being produced and evaluate how personally relevant it was for them. P5 discussed going one step further and looking at source code before deciding who to follow: *"I'll check that out to see what kind of work they've done. I think in programming it's kinda nice that we have this luxury of, I guess, you can look at the source code. [...] You can be your own credentialing system."*

Others said that they used real-life recommendations, such as from co-workers, to find developers on Twitter that are worth following. Another strategy for finding interesting developers, as mentioned by our interviewees, was to look for individuals who are geographically close. However, developers from areas where Twitter adoption was lower struggled with finding local peers.

Finally, several features of Twitter itself were also used in making decisions of whom to follow. On Twitter, users can choose to display a short description of themselves called the *bio* as part of their profile. Developers used this description to form a first impression of whether a user was relevant to their own interests. The number of followers a user had would also influence their opinion of whom to follow. Tweet volume was also important: interviewees might not follow someone if they were worried about keeping up with *"100s of tweets per day." [P3]*

Strategy: Unfollowing Developers on Twitter..

> *"When I find their recent tweets somewhat interesting, then I tend to follow people. Then I just follow them for a while, and when I start noticing that they annoy me, then I fairly quickly also unfollow. So it's on a trial basis. [...] I follow and unfollow people on a regular basis." [P19]*

Network curation is a common task as a developer's interests and the interests of the people they follow change over time. Users have to continuously adapt their networks to this change. Thus, the decision to follow someone on Twitter is not final—developers constantly change their networks to keep it interesting and manageable. Several developers reported that they see following someone more as a trial of whether the content posted by that person is relevant enough and whether they can keep up with the volume.

A few developers mentioned that they have routines for purging their following lists. They would go through the list once in a while and remove accounts that, for example, they had not seen updates from for some time: *"I also tend to go through every once in a while—once a month or once a week, whenever I think about it—just quickly go through my following list and just unfollow things I haven't seen updates from in a while, just to keep it down to things I'm only interested in." [P27]*

Challenge: Consuming Content.

Developers presented several strategies for deciding whom they should follow and for managing their networks. However, content consumption was also a challenge, and to help deal with this, developers adopted four main **strategies**:

Strategy: Filtering..

> *"I'm using TweetDeck with some filters. I think Twitter could provide that. I filter every post with [keyword], for example." [P23]*

To ease consumption of content, especially for developers that follow many others, they use various filtering mechanisms to streamline the content they read.

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]*

When skimming over a tweet stream, profile pictures help developers focus on particular tweets by specific individuals; the profile pictures play an important role in helping them skim content more efficiently. Some interviewees mentioned that the native retweet feature introduced by Twitter some time ago disturbs this routine a bit, as retweeted tweets would appear with profile pictures the reader was not necessarily familiar with.

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]*

Some respondents discussed bookmarking interesting content and reading it at a later time, essentially switching between two different modes of skimming and reading.

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]*

Some respondents recognized that Twitter could impact productivity by leading to distractions. As P10 mentioned, setting aside a specific time each day to read it helped offset potential distractions.

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

Research Question 5 investigates the reasons why many software developers do not adopt Twitter for work purposes. We found they actively 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 the replies were unambiguous and easy to code consistently. We discuss these numbers as well as illustrative quotes from the interviews and surveys.

Barrier: Too Much Noise.

> *"It's a time sink. It just consumes a lot of time to read." [P25]*

Some developers (35 exploratory survey respondents and 5 interview participants) felt they were drowning in the noise caused by an excess of information received on Twitter each day. Some use other tools to stay updated and build networks. P14 decided to build his network on Google+ with preference for the Google+ 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."* P26 discusses the use of IRC channels: *"I get a lot of news about these things on IRC as well. Like as I said I have many technology minded friends so they share links about new stuff there."*.

Barrier: Peer Adoption.

> *"In Taiwan most of the Internet users use Plurk. [...] Two social networks is too much." [P21]*

Several study participants showed that the adoption of social media tools depends on one's social context. For instance, P21 is a Taiwanese software developer who prefers to use a tool more broadly adopted by their peers. In consequence, P21 told us that they do not have enough time to also maintain an active Twitter account—even though they said that it could possibly help them connect with Western developers. Similarly, 8 respondents from our exploratory survey mentioned that Twitter was a poor match for their social networks, i.e., it was not used by their friends or colleagues or even in their country.

Barrier: 140 Character Constraint.

> *"Posts are too short, with almost zero context, with a low signal to noise ratio. I want full articles, preferably with media embedded so I don't have to make multiple clicks to find out what the post is about." [E155]*

Whereas many of the adopters appreciated the brief and concise postings, 9 respondents from our exploratory survey mentioned that they did not like this limitation. The microblog limitation also has the impact that many posters shorten web links and 3 respondents specifically mentioned this: *"Twitter messages are too short to contain anything useful with context. Also people often use link-shorteners in their messages which is a dangerous practice." [E29]*

Barrier: Poor Support for Conversations.

> *"I like conversation. I like having a conversation. I don't like having big conversations, but having two or three people in a conversation, talking about a topic ... it's nice. And I don't find I get conversation on Twitter very much". [P4]*

Twitter was not designed to promote long conversations and multiple conversation threads quickly become unbearable to manage. Additionally, conversations on Twitter 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 interview participants and six exploratory survey respondents mentioned having a hard time using Twitter for its lack of support for conversations. Some mentioned a preference for GitHub and Google+ which provide better support for conversations, especially those that may involve criticism of work or complex explanations.

Barrier: Unsure of Benefits.

Finally, 22 respondents from our exploratory survey said they did not use Twitter because they didn't have a reason or didn't understand why to use it: *"I don't understand it and I don't see any purpose for it." [E43]*

5. VALIDATION SURVEY

The previous section discussed the themes we found from the exploratory survey and interviews for each of the five research questions. We now report on a validation survey that we conducted to gauge how pervasive these themes are among GitHub users.

As reported in section 3, we emailed our survey to 10,000 GitHub users that had been recently active on the site. The questionnaire is available online[6]. We derived the survey questions from the themes found through the exploratory survey and interviews.

1,413 GitHub users answered our validation survey. Fig. 1 provides some details on the demographics of these respondents. All except one respondent said they develop software in some capacity, with the majority (81%) being professional software developers. Roughly two-thirds of our respondents said they use Twitter at least once a week (*adopters*). 70% wanted to be notified of the results of our study.

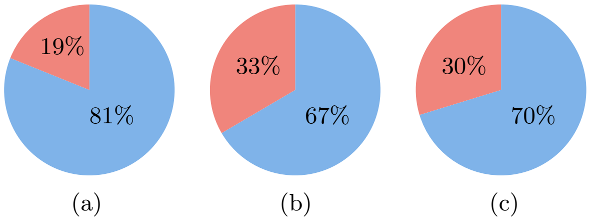

Figure 1: Validation survey respondents. (a) 81% (1,145) professional, 19% (267) non-professional developers, 1 non-developer; (b) 67% (940) use Twitter at least once a week, 33% (473) do not; (c) 70% (993) were interested in study results.

A large part of the survey consisted of statements, each related to a theme from our findings. For these statements, we measured agreement using a five-point Likert-type scale.

[6] https://docs.google.com/forms/d/1FxuT3ORoSjMqQks7hF2gRyncjfJ8DOQBXJaqfmy97pU/viewform?entry.1821350311=example

Figures 2–5 report the results for these statements. Agreement and disagreement are represented by shaded bars, percentages, and absolute numbers. Neutral answers are not shown explicitly, but can be derived from the white space in a chart. We chose this representation to emphasize agreement and disagreement. For these statements, we only report on answers from Twitter adopters.

We revisit the survey responses according to each research question.

5.1 RQ1: Developer Awareness

Our first research question asked how Twitter can help developers become or stay aware of people, trends, and practices related to software development. We found that developers follow other developers, projects, news curators, and thought leaders. This allows them to stay aware of new practices and resources in a timely manner and provides them with access to diverse opinions. Developers also promote their projects and activities, which may in turn help the dissemination of knowledge and increase the adoption of practices and tools.

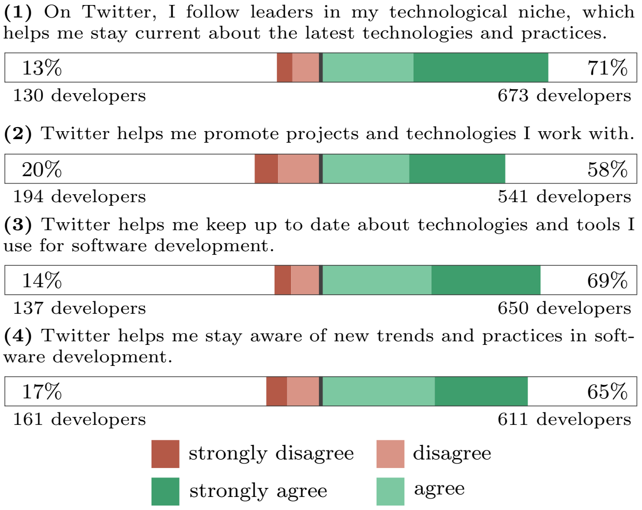

Figure 2: Results from our validation survey (RQ1) for developers who use Twitter at least once a week.

In our validation survey, we included statements for four themes related to this research question (cf. Fig. 2):

(1) 71% of survey respondents (673) agreed or strongly agreed that they follow leaders in their respective technological niches and that this helps them stay current with technologies and practices.

(2) 58% (541) agreed or strongly agreed that Twitter helps them promote the projects and technologies they work with.

(3) 69% (650) agreed or strongly agreed that Twitter helps them stay up to date about technologies and tools they already use.

(4) 65% (611) agreed or strongly agreed that Twitter helps them stay aware of new development trends and practices.

These answers strengthen our finding that many developers use Twitter to stay current with software development practices and tools. This also implies that some developers either feel that they *have to* stay current or that they *want to*. The latter is in agreement with our finding from RQ 2 that developers enjoy learning, but some developers also reported that they thought that they had to stay current to stay competitive.

The statement about promotion shows the lowest agreement in this set of questions. This makes sense: consuming, using, and learning about technologies and practices has to come before one is able to produce anything that one would want to promote or teach to others. As we have access to the Twitter and GitHub usernames for many of the respondents, we will analyze this aspect in future work: it's possible that less-experienced developers disagree with this question because they have yet to produce anything they feel would be worth promoting.

In summary, our initial findings from the exploratory survey and interviews, as well as the results from our validation survey, indicate that Twitter plays an important role in the diffusion of software development practices, tools, and other resources—at least for many members of the population we examined in our study.

5.2 RQ2: Learning

Our second research question asked how Twitter can help developers extend their knowledge. We found that developers ask and answer questions on Twitter, follow experts to benefit from their experience, and feel that participating in conversations helps them learn. Developers said that the qualities and constraints of Twitter enabled serendipitous, undirected learning, sometimes giving them access to resources they wouldn't have been able to find themselves. While participants viewed learning as an investment, they also thought it was fun and rewarding.

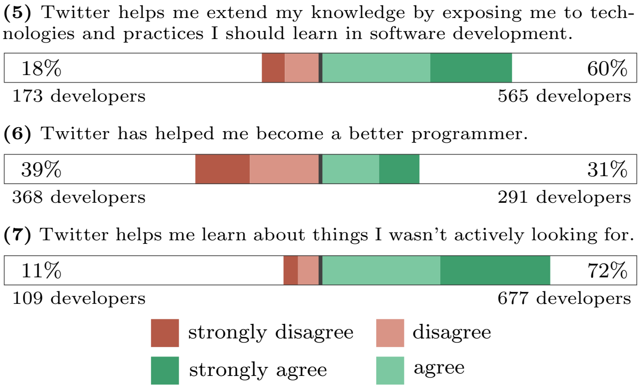

Figure 3: Results from our validation survey (RQ2) for developers who use Twitter at least once a week.

In our validation survey, we included statements for three themes related to this research question (cf. Fig. 3):

(5) 60% of survey respondents (565) agreed or strongly agreed that Twitter exposes them to technologies and practices that they should learn in software development. Only 18% disagreed.

(6) 31% (291) agreed or strongly agreed that Twitter has

helped them become a better programmer—whereas 39% (368) disagreed.

(7) 72% (677) agreed or strongly agreed that Twitter helps them learn about things they were not actively looking for.

These results confirm that exposure to practices and tools through Twitter influences a significant proportion of developers in what and how they learn. This makes it apparent that the thought leaders mentioned in the previous research question really do have a certain influence on their followers. Relatedly, the responses to statement (7) in the survey show that developers use Twitter for *serendipitous* discovery and learning.

Conversely, there were divided opinions on whether Twitter actually helped developers become better programmers. It's possible that Twitter as a *channel* for content is viewed as less important than the actual content itself—more extensive learning materials can only be linked to on Twitter. Similarly, developers might learn less about practices on Twitter than about tools and other resources—after all, tools and libraries can easily be linked to within a tweet and then tried out, while practices take more effort to trial. Future work should investigate this relationship further. Still, 31% of survey respondents, or 291 software developers, agreed that Twitter had made them better programmers—in our view this is still an impressive effect.

5.3 RQ3: Relationships

Our third research question explored how Twitter can help nurture relationships between developers. We found that it can help them discover interesting developers, and achieve trust and rapport with distant colleagues—whether those are co-workers in the same company or just peers working on related open source projects in their spare time. Some developers consciously manage their own image on Twitter as it can provide them with validation and feedback of their work and might even let them access job opportunities. Some developers use Twitter to build communities around technologies they care about.

In our validation survey, we included statements for eight themes related to this research question (cf. Fig. 4):

(8) 52% of survey respondents (485) agreed or strongly agreed that Twitter helps them manage their image as a developer.

(9) 41% (381) agreed or strongly agreed that it helps them build community around projects they contribute to.

(10) 30% (283) agreed or strongly agreed that Twitter is helpful for business networking.

(11) 67% (627) agreed or strongly agreed that it helps them discover interesting software developers.

(12) 49% (452) agreed or strongly agreed that Twitter is helpful in building trust and rapport with other developers. 23% disagreed.

For most of these results, agreement is not that high, with the exception of statement (11). Statement (11) refers to a rather passive, consuming activity. The other statements, such as the one about business networking, all relate to activities that would require developers to interact with others over Twitter. Using the Twitter usernames of survey respondents, we intend to investigate this phenomenon in a future analysis: it's possible that agreement to these statements is correlated with a developer's conversation volume or style on Twitter, or their activities on GitHub.

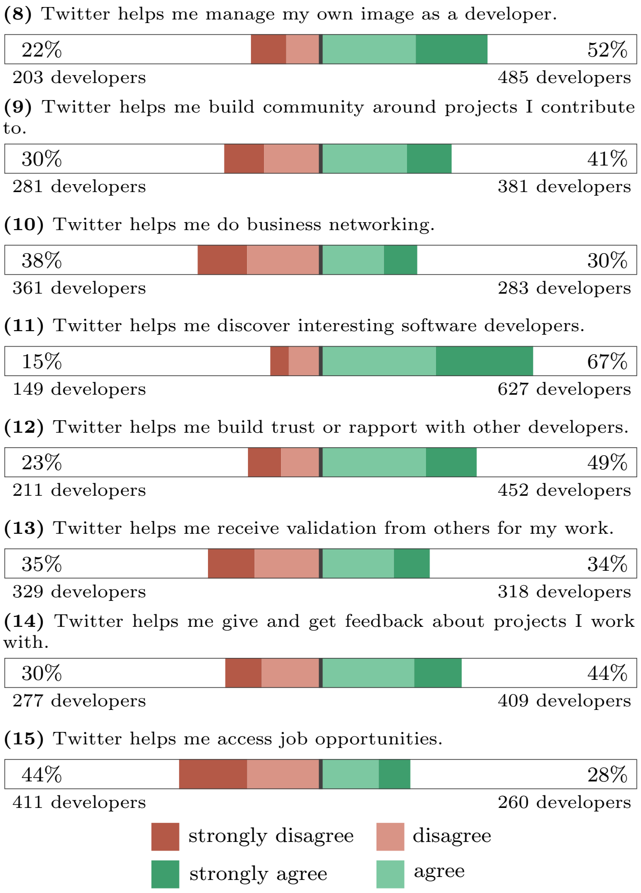

Figure 4: Results from our validation survey (RQ3) for developers who use Twitter at least once a week.

Similar results can be seen in the other statements related to this research question:

(13) 34% of survey respondents (318) agreed or strongly agreed that Twitter provides them with validation of their work. 35% (329) disagreed.

(14) 44% (409) agreed or strongly agreed that it helps them give and get feedback on their projects. 30% (277) disagreed.

(15) 28% (260) agreed or strongly agreed that Twitter is helpful in finding job opportunities. But 44% (411) disagreed.

The activities implied by statements (13) and (14) would again require a developer experienced enough to have produced work that he or she would like to share. Also, many developers might not be able or allowed to share their work openly. An analysis of the experience and job situations of survey respondents might be able to shed light into this issue.

Statement (15) again is probably only relevant to a subset of developers: those who are interested in job opportunities, are experienced enough to be attractive for potential employers, and are able to publish their work openly as a portfolio. Other developers may have to resort to finding work through more traditional channels. Again, a future analysis might provide interesting insights into these results.

It is worth noting, that in terms of absolute numbers, there are still hundreds of developers agreeing with the statements provided in the survey. Consequently, Twitter appears to be useful for managing relationships for at least a subset of the population we surveyed.

5.4 RQ4: Challenges

Our fourth research question investigated the challenges faced by software developers using Twitter and how they cope with them. We found that a central challenge is to maintain a relevant network. To do so, developers may first choose to follow a few thought leaders from their technological niche. They would then discover new developers based on whom the thought leaders retweet and mention. However, each additional user will increase the volume of content in one's timeline—thus, developers would carefully consider whether to follow someone. Following someone was sometimes mentioned as being on a trial basis: some developers routinely remove irrelevant or high-volume users from their timelines.

Another challenge lies in consuming the content. Developers reported a number of different strategies they used to stay on top of their timelines, such as filtering some keywords, using others' profile pictures to quickly skim their timelines, or skimming regularly and often. Others reported that they did not even try to read every tweet but just looked at what was at the top of their timeline.

In our validation survey, we included statements for six themes related to this research question (cf. Fig. 5):

(16) 72% of survey respondents (672) agreed or strongly agreed that they are careful about who they follow on Twitter to avoid information overload.

(17) 67% (628) agreed or strongly agreed that they followed others merely on a trial basis first and then unfollowed users with irrelevant or too much content.

(18) 41% of respondents (368) agreed that it can be hard to cope with the information load on Twitter. 34% disagreed.

Statements (16) and (17) show a relatively strong agreement and low disagreement. This indicates that being careful of whom to follow and following only on a trial basis are both strategies used by many developers who are active Twitter users—at least in the population we surveyed.

We were surprised that only 41% of respondents thought that information overload was a problem on Twitter—we expected more developers to be affected. It's possible that the strategies used to keep the volume low in one's timeline are effective, or developers simply stop using Twitter when they feel overwhelmed.

Statements that related to the advantages and disadvantages of Twitter's constraints show relatively strong agreement:

(19) 68% of survey respondents (632) appreciated the succinctness that the 140 character limit enforces.

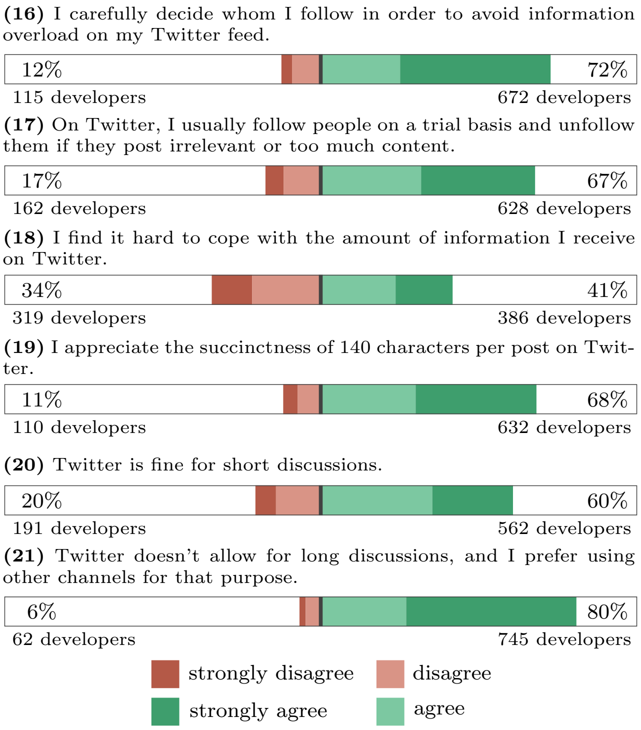

Figure 5: Results from our validation survey (RQ4) for developers who use Twitter at least once a week.

(20) 60% (562) agreed or strongly agreed that Twitter can be used for short discussions.

(21) 80% (745) agreed or strongly agreed that they prefer channels other than Twitter for long discussions.

Contrary to some complaints we heard about Twitter's character limit, 68% of survey respondents (682) agreed that the enforced succinctness helps them. In some interviews, developers told us that this helps them manage the volume of posts they have to skim. A few developers said that on Google+, which has no such limit, skimming was much harder. Therefore, they were more likely to feel overwhelmed.

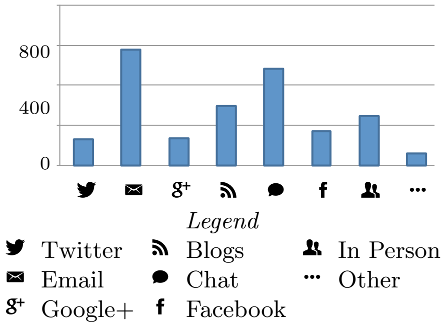

Figure 6: The channels Twitter adopters said they use for longer discussions.

Statements (20) and (21) both addressed the issue of discussions on Twitter. Short discussions seem to be tolerable through Twitter, but the vast majority of respondents said they prefer switching to other channels for longer discussions. Fig. 6 shows a summary of the channels respondents said they would use for longer discussions. Most preferred email or chat, blogs were more popular than personal contact, and only a few used Twitter for longer discussions.

The validation survey responses, in connection with our fourth research question, confirmed most of the findings we reported from the exploratory survey and interviews. However, at the start of this study we anticipated that Twitter would overwhelm the majority of adopters, but this was not the case.

5.5 Adopters and Non-Adopters: Channel Use

Our fifth research question inquired why software developers might choose not to use Twitter. Study participants were concerned about being exposed to too much noise, some disliked the 140 character constraint, and some criticized the poor support for conversations. Some non-adopters were unsure of Twitter's benefits and a few simply had no choice other than to adopt an alternative channel due to a lack of peer involvement on Twitter.

To complement these findings, and to understand how non-adopter channel preferences differ from those of active Twitter users, our validation survey investigated which channels adopters and non-adopters use. We asked respondents which channels they use at least once a week for *staying aware* and *spreading information* regarding software development technologies and practices, and also for *managing relationships* with other developers. We included one closed question per theme and presented channels that were often reported by our interview participants in the second phase of the study.

Fig. 7 shows a comparison of the validation survey answers for each of these three themes. Results from those who actively use Twitter—i.e., *adopters*—are displayed on top, *non-adopters'* results are displayed at the bottom. The vertical bars indicate how many respondents reported to use each channel.

Despite the striking difference in Twitter usage between adopters and non-adopters, both groups reported similar preferences regarding other channels. *Blogs* are widely used both by Twitter adopters and non-adopters for *staying aware* (739 out of 940 adopters and 310 out of 473 non-adopters), followed by email and news aggregators.

For *spreading information*, both groups value in-person communication, email, blogs, and chat, but opinions vary slightly in the order of their preferences. Many non-adopters reported to use email (175 out of 473) for spreading information, followed by in-person communication, chat, and blogs. Twitter is by far the preferred choice among adopters (642 out of 940), followed by in-person communication, blogs, chat, and email. Seemingly, some Twitter adopters refrain from using traditional communication channels such as email and prefer to use social media channels such as Twitter and blogs for spreading information.

For *managing relationships*, 585 out of 940 adopters indicated they use Twitter. Next were email (528), chat, and in-person communication. Non-adopters showed the same ordering of channels—therefore, apart from Twitter, both groups reported similar behaviors.

Channel use was surprisingly similar between Twitter adopters and non-adopters. We had expected stronger differences between usage patterns, e.g., a higher adoption of other channels by non-adopters. Similarly, one could expect Twitter adopters to be more sympathetic to social media channels such as Google+ and blogs than non-adopters. However, our results show similar usage patterns for both groups.

Twitter seems to be a controversial tool among software developers as we could not observe this level of divergence in the adoption of any other channel. Even though the Twitter-centric nature of our validation survey could be the reason for this phenomenon, we believe future research should investigate it.

6. DISCUSSION

Our study uncovered how some software developers use Twitter to support their development activities. Twitter helps them stay aware of software engineering tools and practices, supports learning, and plays a role in maintaining relationships between developers. The specific constraints and mechanisms of Twitter lend themselves to certain usage patterns that seem to help developers extract value from the service.

Our validation survey provides strong support for several of the themes we found, especially those that require only passive consumption of content on Twitter, such as following leaders to stay current in a niche or serendipitous learning. Other themes were less agreed upon in our validation study, and most of those seemed to be most relevant for experienced developers in situations that would allow them to publish open source software. For example, we found the lowest agreement (28%) for the statement that "Twitter helps me access job opportunities." Yet, even those 28% refer to 260 software developers whose professional lives may have been changed because of their Twitter use.

6.1 Usage Strategies

Software developers who use Twitter actively—that is, probably successfully—agreed that they use a certain set of strategies to derive several benefits from the service. A developer just starting to use social media could consider adopting these strategies to increase their chances of staying aware, learning, and building relationships with others. Even though our results are not generalizable (cf. section 6.4), adopting these strategies could make sense as they seem to carry a low enough cost to be worth experimenting with. In particular, we would consider the following strategies:

- Follow leaders in your technological niche.
- Build up your network organically: follow people that those leaders mention or retweet. Extend your network using this strategy.
- Following someone incurs a cost on attention, therefore, carefully decide whom you follow.
- Follow others on a trial basis and reevaluate whom you follow regularly.
- Share what you learn about the technology and related practices.
- Use Twitter for short conversations and switch to email or chat for longer discussions.

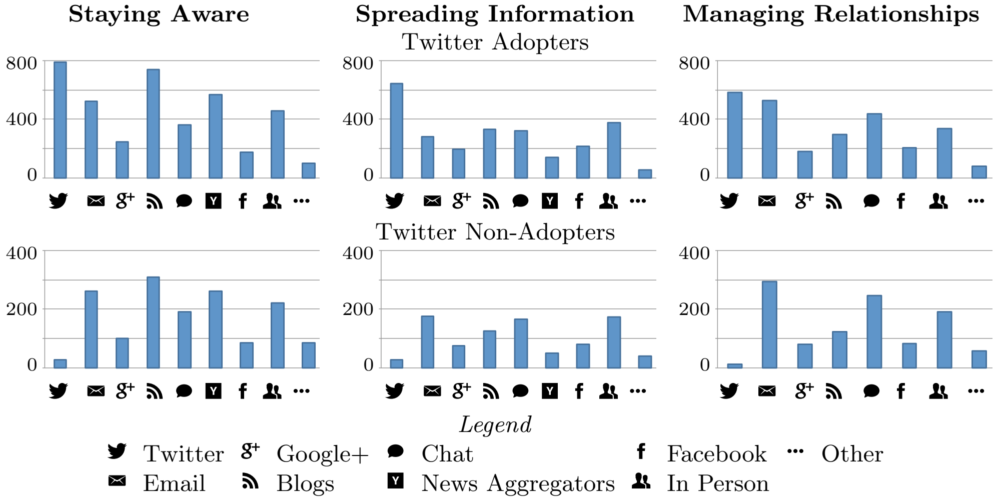

Figure 7: The channels Twitter adopters and non-adopters use for different activities.

These are just some of the strategies that could be derived from our findings and only include those themes with the most support from our validation survey respondents. We discovered other usage strategies in our study (cf. section 4) that might provide additional guidance for developers in specific contexts.

6.2 Understanding Non-Adoption

The insights provided by non-adopters and infrequent users are valuable for understanding the limitations of Twitter in software engineering. Several 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 some 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. Adopters as well as 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.

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

6.3 Contributions

Previous work investigated the role of Twitter in conversations [5, 4, 2] and information dissemination [5]. Others investigated the role of microblogging in increasing awareness and connectedness among co-workers in organizational settings [17]. To the best of our knowledge, our study is the first to conduct an in-depth investigation with a diverse population of software developers. We contrast the viewpoints from developers who have adopted Twitter with those who have not. We also discuss the interplay between Twitter and related services.

Software developers work in a rapidly evolving field where staying current is both required and a challenge. Developers use social media to form and maintain relationships with coworkers within the same organizational context, but also to connect with and stay aware of other developers around the globe. These distinctions are important when providing recommendations for software developers themselves, but also for organizations willing to improve their knowledge management strategies to better adapt in evolving domains.

6.4 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 to everyone. Concepts and themes that emerged from our analysis cannot be generalized.

We invited active yet random GitHub users 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.

However, the majority of our study participants were software developers—many of them professional ones. Our validation survey found high agreement for several of our findings. Therefore, we believe that we have uncovered valuable insights regarding software developers' use of Twitter.

6.5 Future Work

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 an increasing 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 could 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.

7. 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 those challenges, 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 in a rapidly evolving field.

8. 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.

9. REFERENCES

[1] 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 *Proc. of the 2nd Int. Workshop on Web 2.0 for Software Engineering*, Web2SE '11, pages 31–36, New York, NY, USA, 2011. ACM.

[2] D. boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *System Sciences (HICSS), 2010 43rd Hawaii Int. Conf. on*, pages 1–10, 2010.

[3] J. Corbin and A. Strauss. *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage Publications, 3rd edition, 2008.

[4] C. Honey and S. Herring. Beyond microblogging: Conversation and collaboration via twitter. In *System Sciences, 2009. HICSS '09. 42nd Hawaii Int. Conf. on*, pages 1–10, 2009.

[5] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: Understanding microblogging usage and communities. In *Proc. 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.

[6] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proc. of the 19th Int. Conf. on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[7] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann. Design lessons from the fastest q&a site in the west. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI '11, pages 2857–2866, New York, NY, USA, 2011. ACM.

[8] A. E. Marwick and d. boyd. I tweet honestly, i tweet passionately: Twitter users, context collapse, and the imagined audience. *New Media & Society*, 2010.

[9] P. McFedries. Technically speaking: All a-twitter. *IEEE Spectrum*, 44(10):84–84, 2007.

[10] D. Pagano and W. Maalej. How do developers blog?: an exploratory study. In *Proc. of the 8th Working Conf. on Mining Software Repositories*, MSR '11, pages 123–132, New York, NY, USA, 2011. ACM.

[11] 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 *Proc. of the 2013 Int. Conf. on Software Engineering*, ICSE '13, pages 112–121, Piscataway, NJ, USA, 2013. IEEE.

[12] 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 Int. Conf. on*, pages 596–599, 2012.

[13] P. C. Rigby and M.-A. Storey. Understanding broadcast based peer review on open source software projects. In *Proc. of the 33rd Int. Conf. on Software Engineering*, ICSE '11, pages 541–550, New York, NY, USA, 2011. ACM.

[14] 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 *Proc. of the ACM 2013 Conf. on Computer Supported Cooperative Work and Social Computing*, CSCW '13, New York, NY, USA, 2013. ACM.

[15] 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 Conf. on*, pages 247–250, 2012.

[16] X. Wang, I. Kuzmickaja, K. Stol, P. Abrahamsson, and B. Fitzgerald. Microblogging in open source software development: The case of drupal using twitter, 2013.

[17] J. Zhang, Y. Qu, J. Cody, and Y. Wu. A case study of micro-blogging in the enterprise: Use, value, and related issues. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI '10, pages 123–132, New York, NY, USA, 2010. ACM.

[18] D. Zhao and M. B. Rosson. How and why people twitter: the role that micro-blogging plays in informal communication at work. In *Proc. of the ACM 2009 Int. Conf. on Supporting Group Work*, GROUP '09, pages 243–252, New York, NY, USA, 2009. ACM.