

Megan Squire

Department of Computing Sciences

Elon University

Elon, NC, USA

msquire@elon.edu

*Abstract*— Stack Overflow is an enormously popular question-and-answer web site intended for software developers to help each other with programming issues. Some software projects aimed at developers (for example, application programming interfaces, application engines, cloud services, development frameworks, and the like) are closing their self-supported developer discussion forums and mailing lists and instead directing developers to use special-purpose tags on Stack Overflow. The goals of this paper are to document the main reasons given for moving developer support to Stack Overflow, and then to collect and analyze data from a group of software projects that have done this, in order to show whether the expected quality of support was actually achieved. The analysis shows that for all four software projects in this study, two of the desired quality indicators, developer participation and response time, did show improvements on Stack Overflow as compared to mailing lists and forums. However, we also found several projects that moved back from Stack Overflow, despite achieving these desired improvements. The results of this study are applicable to a wide variety of software projects that provide developer support using social media.

*Index Terms*—developer support, technical support, quality, Stack Overflow, mailing list, forums, metrics, social media.

## I. Introduction

Social media is changing the way work is done in many domains, and software development is no exception. Distributed teams of software developers, such as those working to build free, libre, and open source software (FLOSS), were especially quick to begin using digital communication media - such as email mailing lists and IRC chat - for getting things done. Some long-standing projects such as Apache *httpd* server and the Linux kernel have ongoing mailing lists for developer communication, with archives stretching back to the early 1990s. As social media expanded into social networks, blogs, microblogs, and question-and-answer (Q&A) formats, software developers have continued to use these tools to accomplish their community objectives [1].

This paper is motivated by our observation of two simultaneous and complimentary trends: the rise of web-based software support intended specifically for developers, and the introduction of highly successful user-driven Q&A web sites. As web sites have morphed into web services, the associated application programming interfaces (APIs) have become

indispensable part of the web programming landscape [2]. API producers need to offer support to skilled software developers in addition to providing traditional non-technical end-user support. As was the case with other end-user support, developer support was first offered digitally through email mailing lists and web-based discussion forums (or hybrids such as Google Groups). Mailing lists have the advantages of being very easy to understand, and they are available on many platforms. Many people prefer having the messages sent to them ("push"), rather than having to visit a special web site ("pull"). However, the list joining process can be awkward for a user who only has one or two questions. Mailing list archives can be difficult to search, and some common questions may get asked over and over again, which frustrates long-term members.

The web-based discussion forum is an improvement over standard mailing lists in that it typically provides a browse and search facility, and usually has no login or signup required to read the archive. This helps solve the problem of repetition in questions and answers. However, depending on the forum software, there are different degrees of quality control over questions and answers. (This is especially true if the developers are expected to help each other in addition to getting answers from the software provider.)

Q&A web sites (e.g. Quora, Stack Overflow, Yahoo Answers) provides the browse and search features of a web-based discussion forum, however they also include additional incentives designed to improve answer quality, such as badges or a voting system. Stack Overflow also allows questions to be marked as duplicates, and it encourages users to edit each others' questions and answers to improve readability and quality.

Stack Overflow was created in 2008, and by mid-2014 had nearly 3 million registered users asking 7.5 million programming-related questions. It is the 54th most-visited site in the world, with an average of more than five minutes and four pages per visit. [3] With the increasing popularity of Stack Overflow, some software project leaders began questioning whether their entire developer support infrastructure (mailing lists or web-based discussion forums) should be moved to Stack Overflow instead. In fact, this question appears in Stack Overflow's own community wiki as early as July 9, 2009 [4] when a user asks "Is it okay to use Stack Overflow as the support forum for a product or project?" After considerable back-and-forth discussion in the community and with Stack

Overflow leadership over the next five years [5][6], the answer seems to be that it is always encouraged to ask programming-related questions on Stack Overflow, even if those are about a commercial product, and this is true whether or not that corporation has its own support infrastructure in place. In addition, companies and individuals are encouraged to use tags on Stack Overflow (for example the tag is "facebook" for questions about the Facebook API), and that financially "sponsoring" one or more tags is also encouraged. However, there are also some warnings from Stack Overflow leadership that "outsourcing" [4] an entire support infrastructure to Stack Overflow is not acceptable. What is meant by "outsourcing" and "entire" is not clear.

Thus, many projects with online mailing lists or forums began debating the efficacy of moving some or all of their support infrastructure to Stack Overflow, instead of staying with their current infrastructure [7][8]. The debates typically centered around the perceived advantages of Stack Overflow, including the quality of the postings and larger user community, the modern interface, etc.

The research questions for this paper came from observing this ongoing debate and the significant lack of data to support either side. The questions this paper attempts to answer are as follows:

- Research Question 1: Which projects have moved to Stack Overflow, and what are the reasons given for closing down existing developer support channels and moving?
- Research Question 2: How does Stack Overflow compare to the previous support channels (mailing lists, forums) in terms of the quality indicators given in RQ1 as reasons to move?

By answering these two questions, we will be able to understand which projects have moved, what their motivations were for moving, and whether the move turned out to be validated – did the project leaders get what they were looking for? Did the resulting developer support actually get better?

In the next section, we provide an overview of the background literature on the existing quality indicators for technical support in general, for developer support (including documentation), and for crowdsourced Q&A sites.

Section III describes how we collected our data, including our survey of which projects have moved to Stack Overflow, and what their motivations were for moving their developer support. We also provide data for our questions about whether participation increased and whether response time decreased after the move.

In Section IV we discuss these results and show some areas where we had surprising results. In Section V we point out the limitations of our study. Finally, we conclude with a summary of what this analysis means for developer support more generally.

## II. Related Literature

In this section we review the literature on quality indicators for software support, and we provide background on how software is created using two online channels: (1) mailing lists

and forums, especially Google Groups, and (2) Stack Overflow.

Most of the literature on assessing the quality of technical support focuses on face-to-face or telephone support, and was written in the decades before online support was common. (In addition, none of that literature deals specifically with developer support.) For a good summary, we point to the empirical study in [9] which found that (with end-users at least) the most valued attributes of a positive technical support experience were, first, competence in the provider and, second, response time.

The larger body of literature regarding measurement of quality postings in online channels (forums, mailing lists, Q&A web sites) is also relevant to this study. Chai, *et al.* [10] use machine learning systems to classify user-generated social media postings as high or low quality. Zhu, *et al.* [11] develop a multi-dimensional model to determine whether answers on Q&A sites such as Yahoo! Answers are high quality. They discover 13 different dimensions of support quality (politeness, readability, informativeness, etc.) and outline a brief plan to study these dimensions using natural language processing techniques. Gamalielsson, *et al.* [12] look specifically at responsiveness (response time) on mailing lists as an indicator of the health of a FLOSS community, concluding that "[i]n healthy Open Source communities people are active and responsive to questions during the life cycle of a software system."

Other studies on resources for developers include studies of API documentation [13], which is relevant since documentation is often a first line of defense in providing high quality developer support. However, [13] shows that most web searches for help with APIs return not community-managed or corporate-managed documentation repositories, but socially constructed documentation, or what they call "crowd documentation".

Storey, *et al.* [1] confirm this trend within software development toward the crowdsourced and the social. Social media is changing the way work is done in software development communities of practice. Email lists and Stack Overflow are presented as book ends for the process of continued socialization of the development cycle: email lists are digital and were used very early as a development tool, Google Groups and forums offered slightly more social features, and Stack Overflow is both digital and socially-enabled, offering a way for knowledge to be shared as a community resource. Another work from Treude *et al.* [14] shows that Stack Overflow is an extremely important developer resource, providing updatable ("wiki-like") documentation and high quality answers with the ability for some back-and-forth dialogue between developers.

On measures of response time, one of the previously-mentioned measures of support effectiveness, [15] claims that Stack Overflow performs well: 92% of questions are answered by "the crowd", and of those, the median response time is 11 minutes. Answers judged to be satisfactory have only a slightly different profile: 21 minutes is their median response time. Vasilescu, *et al.* [16] study behavior of users as they shift from

mailing lists to Stack Overflow to discuss the programming language R. They compare the same users on both channels, and find that the social and gamification aspects of Stack Overflow are enjoyable for the participants and result in faster answers.

### III. Data

#### A. Which projects have moved?

To answer RQ1, we collected evidence of software projects moving their developer support to Stack Overflow. We performed various searches of mailing lists (using MarkMail, an aggregator of 8800 software project mailing lists) and Google Groups to determine which projects were moving. We used general web searches to locate projects that were moving and that were not located on either MarkMail or Google Groups. We also found several candidate projects mentioned in discussion threads on Stack Exchange Meta [4-6].

TABLE I. Projects that have moved their developer support to Stack Overflow, sorted by date of announcement

Ref.	Announced	New SO Tag	Moved From
A1	22-Apr-2011	Pex	Forum
A2	24-Aug-2011	Facebook	Forum
A3	14-Sep-2011	Flattr	Forum
A4	16-Nov-2011	Foursquare	GG
A5	06-Feb-2012	google-appengine	GG
A6	10-Feb-2012	google-maps-api-3	GG
A7	07-Mar-2012	soundcloud	GG
A8	01-May-2012	google-bigquery	GG
A9	06-Jun-2012	Shopify	GG
A10	14-Sep-2012	youtube-api	GG
A11	25-Sep-2012	Dwolla	Forum
A12	16-Oct-2012	google-plugin-eclipse	GG
A13	20-Nov-2012	breeze	Forum
A14	04-Jun-2013	docusignAPI	Forum
A15	27-Jul-2013	gracenote	Forum
A16	08-Oct-2013	sagepay	Forum
A17	31-Oct-2013	cosign-api	Forum
A18	04-Dec-2013	google-cloud-sql	GG
A19	06-Mar-2014	sony	Other
A20	23-Apr-2014	socrata	Forum

Table I summarizes the 20 projects we found that had moved their support from a mailing list or forum to Stack Overflow. We list a reference number for each project, so that we can match these to a list of that project's announcement, which are listed in the appendix to this paper. The references are in the form [An] where n is a project number, 1-20. (They are sorted in the order that they announced that decision to their mailing list or forum, earliest to latest.) This table does not include projects that were only discussing a move, or projects where a move had been proposed but never carried out, or projects which have moved and then moved back (more on

these in Section IV). Table I also lists what the new official tag is for the project on Stack Overflow, which is important since this is how questions are organized on that site. It also lists whether the support was previously a forum, Google Groups mailing list (GG), or other.

#### B. Why did they move?

Table II shows a count of the reasons given by projects identified as moving, from Table I. Table III shows that the most common reason for moving to Stack Overflow is that the developer community is larger: more people use the site. Higher quality of questions and answers is given as the second-most common reason (this includes de-duplication of questions), and faster response time is mentioned as the third most common reason.

TABLE II. Summary of stated reasons for moving

Ref.	New SO Tag	Reasons
A1	Pex	none
A2	Facebook	4, 5
A3	Flattr	1, 4, 5
A4	Foursquare	1, 2, 3
A5	google-appengine	1, 2, 6
A6	google-maps-api-3	1, 4
A7	soundcloud	1, 5
A8	google-bigquery	1, 2, 3
A9	Shopify	2, 3
A10	youtube-api	2
A11	Dwolla	1, 2, 3
A12	google-plugin-eclipse	1, 2, 6
A13	breeze	1, 4
A14	docusignAPI	1, 3
A15	gracenote	5
A16	sagepay	none
A17	cosign-api	6
A18	google-cloud-sql	none
A19	sony	none
A20	socrata	6, 7

TABLE III. Counts of stated reasons for moving

Reason	Reason Description	Count
1	Higher level of participation	10
2	Higher quality questions/answers	7
3	Faster response time	5
4	More/easier access to expertise	4
5	More convenient for company	4
6	More features (search, reputation)	4
7	Lower cost / free	1

### C. Which of these projects should we study?

To answer RQ2 (does Stack Overflow compare favorably to the mailing list or forum on these quality indicators), we chose to measure two of the seven factors shown in Table III: level of participation in the discussion (reason 1), and response time (reason 3). These factors were also mentioned by our background literature review (Section 2) as relevant to quality support, and they are straightforward to calculate.

Of the 20 projects shown in Table I, four projects explicitly identified both of these factors (number of participants and response time) as reasons to move: [A4] Foursquare API, [A8] Google BigQuery, [A11] Dwolla, and [A14] Docusign API. We will describe each project here, and describe their rationale(s) for moving.

Foursquare is a location-based social media application. For third-party development, it provides an API to their database of places and locations (such as restaurants, shopping, entertainment venues). In 2009, Foursquare launched a Google Groups-based mailing list and developer forum for API discussions and questions. In 2011 they announced that the Google Group would become announcement-only and that all new developer questions should use the 'foursquare' tag on Stack Overflow. Here is an excerpt of the message that announced the move (reasons are shown as underlined) [A4]:

*[A]s of late we've started really hitting the limits of using such a light-weight tool for managing a developer community well over 10,000 strong.*

*I've decided to move our developer forum off of this mailing list and on to StackOverflow, where I'm convinced all of you will get more timely and relevant help for your questions from not just each other, but the large number of developers who regularly participate on the site. I'll still be monitoring and answering questions (trying to get my reputation score up!) but hopefully the new format will encourage greater participation by all members of the community...*

Google BigQuery is a project that provides a REST-based querying interface to Google's Cloud Storage infrastructure. In 2010, Google launched a Google Groups-based mailing list and discussion forum ("BigQuery-Discuss") as a developer forum. In 2012, they announced that technical discussion on BigQuery-Discuss would be closed, and they would be moving to Stack Overflow using the 'google-bigquery' tag. Below is an excerpt from the "Why are we moving?" section of the goodbye message [A8]:

*We are moving technical discussion to Stack Overflow because we think this will improve developer support, increase the speed that questions get answered, and improve the quality of answers.*

*Many Google developer products already use Stack Overflow to provide technical support, and a large number of Google engineers are already heavily involved in the Stack Overflow community.*

Dwolla provides a developer API for their online payments system. In 2011, they launched a self-hosted developer forum. Dwolla has also offered developer support using a commercial solution (by GetSatisfaction) and in 2012 they announced a move to Stack Overflow, where they planned to use the 'dwolla' tag [A11a]:

*Stack Overflow is rich in content and community support. This move will open up our discussions and support to the entire development community as a whole, thus providing better and faster support to everyone.*

A second announcement (posted on the blog) also stressed cost [A11b]:

*[B]y having developers post their questions to Stack Overflow, we'll be opening up the discussion to the development community as a whole... Quicker, easier, more efficient. Oh yah, and it's free for everyone (including us).*

Docusign provides a developer API for their system for managing digital signatures. In 2010, they launched a self-hosted developer forum using commercial community discussion software (by Lithium). In June 2013 they announced that their transition to StackOverflow would begin and that posts should be tagged 'docusignapi', and the forums were made read-only in July of that year. We noticed 172 Stack Overflow questions that also used the tag 'docusign' (whereas 533 questions were tagged 'docusignapi'), so we included both tags in our analysis.

### D. Data about participation

There are several ways to measure the number of participants on a mailing list/forum and on Stack Overflow. One way is to simply count the number of unique users participating in an online community, both overall and for each thread. Of course, this only captures people who are involved in the actual discussion, and not those that are reading but not posting. To solve this, we can also search for view counts; this will answer how many times the thread was viewed or read. Another way to measure high activity in a forum or mailing list is to find out a ratio of answered to unanswered questions.

We collected data to measure all three of these factors, for the four projects mentioned, on both Stack Overflow and on the mailing list or forum used by that project previously. The source code and queries we used to collect all the data for this paper are kept on Github in our code repository [17]. All data for the tables as shown (Tables IV-VII) was collected on May 28, 2014.

Table IV (next page) shows the "number of participants" metrics for each of the four projects on their mailing list or forum. How we calculated each line (and why there is some missing data) is described in detail.

- Line 1: Total posts/threads. Foursquare and BigQuery thread totals were reported on the home page of the Google Group archive. Dwolla total threads were

reported on the home page of their forum. Docusign thread count is calculated by summing the post counts on the topic page of the forum.

- Line 2: Posts with no reply. Foursquare and BigQuery unanswered posts were retrieved by using the Google Group filter "Show only topics that match 'Are not replied to'" and counting the results. Dwolla unanswered posts were calculated by manually counting messages with zero replies in the categories "question", "problem" and "idea". Unanswered post count in Docusign was retrieved by using the advanced search facility in the forum software and setting it to show all posts with no replies.
- Lines 3a-c. Foursquare view counts and BigQuery view counts were derived from the home page of the forum on the Google Groups web interface. View counts were not available for Dwolla postings at all. View counts for Docusign were calculated from their post listing pages on the forum.
- Lines 4a-b. Foursquare and BigQuery unique users were calculated as unique question askers, as reported in the Google Groups interface. Dwolla total unique users were reported on the home page of their forum. The Docusign participant count was not available.

TABLE IV. "Level of participation" data for mailing lists / forums

		FS	BQ	DW	DS
1	Total threads	1850	222	475	5492
2	Unanswered threads	413 (22%)	65 (29%)	58 (12%)	123 (2%)
3a	View count: average	32	40	-	1067
3b	View count: highest	2132	782	-	8480
3c	View count: lowest	0	1	-	278
4a	Total Participants	1023	110	550	-
4b	Participants per Thread	.55	.490	1.15	-

Table V shows level of participation data for Stack Overflow. Our method for calculating the items in Table V can be explained as follows:

- Line 1. This is a count of the total number of questions for that tag on Stack Overflow.

- Lines 2a-c. The concept of an "unanswered question" is a little more complicated on Stack Overflow than it would be on a mailing list or forum. We divide the concept of unanswered questions into three Types. Type 1 are questions with no upvoted answers; they might have an answer but no one upvoted it, and the original asker did not select it as "correct" (so its score is 0 in the database). Type 2 are questions with no answers, but there may be comments on the question, indicating that someone needed more clarification or made a suggestion but did not want to classify their comment as an official answer. Type 3 are questions with no answers and no comments. Type 3 are most equivalent to the "unanswered threads" in Table III.
- Lines 3a-c. Question view counts are reported in the database, and we calculated the average, high, and low per tag.
- Lines 4a-b. Total unique participants is a count of how many unique users (participants) asked questions, answered questions, or commented on posts for that tag. 4b divides those unique participants by the number of questions (from Line 1).

TABLE V. "Level of participation" data for Stack Overflow

		FS	BQ	DW	DS
1	Total questions	1178	958	71	705
2a	Unanswered questions (Type 1)	453 (39%)	234 (24%)	28 (39%)	280 (40%)
2b	Unanswered questions (Type 2)	147 (12.5%)	63 (6.6%)	6 (8.5%)	54 (7.7%)
2c	Unanswered questions (Type 3)	51 (4.3%)	26 (2.7%)	2 (2.8%)	16 (2.3%)
3a	View count: average	523	213	172	77
3b	View count: highest	31838	8513	1199	951
3c	View count: lowest	6	13	10	9
4a	Total Unique participants	1833	837	97	392
4b	Participants per Thread	1.57	.87	1.37	.57

### E. Data about response time

We stated that a common reason for moving developer support to Stack Overflow was to improve the response time between when the question was asked and when it was answered. Table VI shows the results for forum/ mailing list response time, and Table VII shows the results for Stack Overflow response time. The SQL queries used on the Stack Overflow Data Explorer to return these results are given at [17]. We should note that Stack Overflow response times are for answers only, not comments. It may be that the response times would have actually been faster than the numbers shown in Table VII, if we had also included comments to questions and not just answers.

TABLE VI. Response Time for questions on mailing lists and forums, by project

	FS	BQ	DW	DS
Mean response time (in days)	-	7.45 days	-	28.96 days
Standard deviation	-	876 days	-	104 days
Median response time (in days and hours)	-	.75 days (18 hours)	-	.83 days (20 hours)

TABLE VII. Response Time for questions on Stack Overflow, by project

	FS	BQ	DW	DS
Mean response time (in days)	9.12 days	5.06 days	19.5 days	1.31 days
Standard Deviation	41.86 days	29.85 days	58.6 days	8.12 days
Median response time (in days and hours)	.56 days (13.44 hours)	.26 days (6.24 hours)	.63 days (15.12 hours)	.18 days (4.32 hours)

Google BigQuery average response time was calculated by manually downloading all the 222 Google Groups postings as plaintext email messages (raw list of URLs available at [17]), and subtracting the reply dates from the original dates for all messages that had a reply.

Docusign forum posts were downloaded automatically, and the timestamp difference for messages and replies were

calculated using the code available at the author's Github repository [17].

Foursquare averages are not included in Table VI since it was prohibitively difficult to download 1178 forum posts manually. (The Google Groups JavaScript-based interface has programmatically disallowed easy automatic collection of this data, so our manual collection of all those forum posts was limited to the smaller Google BigQuery forum, for reasons of time.)

Dwolla forum posts are not timestamped with enough specificity to be able to calculate response time. For example, the typical timestamp for a message will say it was posted "over 2 years ago" or had its "last reply 5 months ago", but no actual date and time stamp.

## IV. Discussion

In this section we apply the data we gathered and calculated in Section 3 to answer our research questions, which were as follows:

- Research Question 1: What are the reasons given for moving an existing developer support channel to Stack Overflow?
- Research Question 2: How does Stack Overflow compare to the previous communication channels in terms of the quality metrics given as reasons to move in RQ1?

To answer RQ1, we first established a list of projects that actually completed the move to Stack Overflow, closing or abandoning their previous developer support channel in the process. We read each announcement and categorized the reasons given for the move into seven categories. Of these seven reason categories, we selected to study "more participation" and "response time" as measures. Of the 20 projects, we selected four that had both of these as criteria for the move: Foursquare API, Google-BigQuery, Dwolla API, and Docusign API. Two of those projects were previously using Google Groups (a hybrid mailing list/discussion forum), and two others were using off-the-shelf discussion forum software. We collected data on participation levels and response time for each of the four projects, before and after the move.

### A. Result: Participation Levels

We find that there are many ways to measure participation levels. We started with "number of total questions asked" (Line 1 in Tables IV and V). The results for this were mixed. For Google-BigQuery, the number of threads on Stack Overflow was greater than it was on Google Groups. However, for the other three projects, the number of questions on Stack Overflow was lower than the number on their original forum. There are a few things to consider when evaluating this result. First, recall from Section 1 of this paper that Stack Overflow does allow to find and remove duplicate questions (and in fact this behavior is encouraged, through badges), and for many projects listed in Table II and Table III, this de-duplication feature was listed as an advantage. Second, Stack Overflow is

intended to be strictly about programming questions, and the users there are enthusiastic about closing off-topic questions. We did not calculate how many questions were closed in this manner for being duplicates (nor did we calculate how many forum posts were moved or closed for similar reasons). Third, Google has purchased sponsored tags on Stack Overflow (for BigQuery, among other projects), but Foursquare, Dwolla, and Docusign have not purchased sponsored tags. (Sponsored tags show the image of the company next to the tag name, as a form of advertising.) So this may be a factor that is driving the popularity of the "google-bigquery" tag.

We next gathered data on level of participation, specifically, whether the number of unanswered questions increased or decreased after the move. The argument is that putting a question on Stack Overflow will result in more people viewing the question, and more people viewing it means someone is more likely to be able to answer it. This line of reasoning is familiar; it is closely related to the famous "given enough eyeballs, all bugs are shallow" aphorism (sometimes called Linus' Law) about open source software [18]. This was a little more tricky to measure before and after the move, since Stack Overflow has at least three ways to measure whether a question was answered or not. We propose that the "Type 3" non-answers shown in Table V (line 2c) are the closest to the concept of "unanswered" found in mailing lists or forums. On this measure, three of four projects (Foursquare, BigQuery, and Dwolla) saw considerable improvement in reducing unanswered questions by moving to Stack Overflow. Docusign's answer rate was the roughly the same (2% vs. 2.3%). Whether this success is due to the "more eyeballs" theory or just a rejuvenated developer support effort on the part of the company is not known.

Next, we measured view counts, with the idea that increased participation also extends to how many people are reading the posts (not just writing or responding). On this measure (lines 3a-c on both Table IV and Table V), all projects but Docusign saw improvements after moving to Stack Overflow.

Finally, we calculated the number of active participants (askers and answerers), as well as how many participants per thread. For Foursquare-API and Google-BigQuery, the participants and thread numbers all improved on Stack Overflow. For Dwolla, the number of unique participants decreased, but the number of participants per thread increased. We did not have data for the number of Docusign participants on the forum, but on Stack Overflow their participants per thread number is .57 (meaning that each person is contributing to just under two threads on average).

## B. Result: Response Time

To answer the second half of RQ2, we calculated a few metrics to show whether the speed of responses was faster on Stack Overflow, compared to the original channel. Due to the way the forums were designed, we could only gather "pre" data for Google-BigQuery and Docusign. We found that the use of Stack Overflow improved Google-BigQuery response times by 12 hours, and Docusign developers got answers on Stack Overflow almost 16 hours faster (medians). For the

other two projects, we report the Stack Overflow response times, but they are less impressive, especially with no "pre" time to measure against. We observe that *none* of these median response times is close to the Stack Overflow average of 11 minutes that was reported earlier by [15], but at least for two of the projects, showing any response time improvement *was* a desired outcome. (And for those projects, we should note that no specific numeric goals were given at the time of the announcement). In short, on Stack Overflow, response times may be quicker, but are not likely to reach 11 minutes.

## C. Moving Back

Despite these gains in participation and response time, we also know that there were a few cases where the move to Stack Overflow was un-done, despite positive or mixed results in response time and participation. For example, in March 2014, Dwolla announced that even though they had moved to Stack Overflow only a year before, they were once again going to host their own discussion board (powered by Discourse and available at [discuss.dwolla.com](http://discuss.dwolla.com)) [A11c]:

*The purpose of the Discussion Board is exactly what the name suggests—for users to easily start a discussion amongst one another, whether they own a small boutique in Georgia, have a feature request they'd like to share with the development team, or are a developer working on an integration into their ecommerce site.*

That posting alludes to many kinds of users with differing needs, including - but not limited to - developers who write code.

The issue of what to do with non-programming questions was consistently mentioned in community discussions about Stack Overflow as a replacement for traditional social media channels (e.g. [4-6]). Since Stack Overflow is explicitly for programming questions, non-programming questions are routinely closed for being off-topic.

Thus, it may be that Dwolla's new discussion board is intended to be used for non-programming questions, whereas the Stack Overflow solution could still be used for programming questions. (The 'dwolla' tag is still in use on Stack Overflow as of this writing, although no mention of Stack Overflow is made on the Dwolla blog or discussion forum.)

Another community, OpenXava, initially closed its developer forums and moved everything to Stack Overflow with the following announcement [A21]:

*...[W]e want to promote StackOverflow for support over SourceForge forums.... If we put these questions in StackOverflow with the appropriate tags we have a better chance of getting answers, because the StackOverflow community is greater than the OpenXava one...*

But then re-opened the forums three months later with this update (reason underlined for emphasis):

*After three months using StackOverflow we come back to SourceForge forum. The main reason is that StackOverflow moderators close valid OpenXava questions, like this one (link), and we think this is unacceptable. Of course, if you like StackOverflow you can put OpenXava questions on it, and we'll answer you, but we'll promote SourceForge forums as main support source for OpenXava.*

A few other communities have figured out that Stack Overflow is not appropriate for all developer support, but is useful for specific things. They have thus proposed hybrid solutions. Gracenote [A15], Shopify [A9], and Facebook [A2] are three such examples. Facebook has attempted to delineate the various support channels with their announcement:

*Moving forward, Stack Overflow is where you'll go to have your Facebook Platform questions answered; Bugs is where you go to report an issue with a legitimate repro, and you should participate in the Group to discuss, chat, and talk with us directly.*

Shopify provides a longer list of pros and cons about moving to Stack Overflow. The announcement very clearly anticipates that they will still need other social media for communication with developers (reasons have been underlined for emphasis):

*Naturally there are downsides with the move. The most obvious one is that we lose the direct public communication channel between developrs [sic] and Shopify. Because of that we're not abandoning the mailing list entirely but rather repurposing it. We've renamed the list ...and relaunched it as a place for developers to discuss the business-oriented aspects of app design. How much should I charge for my app? What's the best way to publicize my new release to customers? How to I get feedback? All these topics still have a home in the new list.*

These project leaders understand that software development encompasses more activities than just programming, and thus Stack Overflow, despite its advances in response time and participation levels, should really be used as intended: for programming questions.

#### V. Limitations of the study

There are a few limitations to this data and methodology, which give us some ideas for future work. First, and most importantly, we were only able to calculate "pre" response times for two of four projects. This makes it difficult to tell whether moving to Stack Overflow represents an improvement in this area. The first reason for this limitation is that Google Groups uses a JavaScript-based interface that makes it tricky to collect the postings in an automated way, and thus we were only able to collect data from that site for the smaller of the two Google Groups projects (we collected Google-BigQuery, but not Foursquare). We were unable to find another reliable archive of the Foursquare posts to calculate this number.

Similarly, Dwolla's forum software inhibited our collection of the granular time data necessary for us to be able to perform these calculations.

A second limitation of this work is that we did not measure the second-most-common reason for moving to Stack Overflow: the expected higher quality of postings, especially answers. The background literature review did point out that quality of answers and level of expertise were both very important to an end-user's feeling of satisfaction with support. Stack Overflow does have the concept of upvotes and downvotes for both questions and answers, so this would have been straightforward to measure there. However, mailing list postings and forums do not typically include any way to judge the user's perception of quality, making it difficult to gather similar rankings on the "pre" side. (Dwolla and Docusign's forum software does have the concept of "highlight" and "thumbs up" on a posting, but it is not clear how many people actually used these features.) Perhaps in future work it would be possible to devise some ways to measure perceived quality of questions and answers on the mailing lists and forums, and then compare that to the quality metrics given by votes on Stack Overflow.

#### VI. Conclusion

The goals of this paper were to identify projects that were moving their developer support to Stack Overflow, to classify the reasons for the move, and then to measure whether the move was successful in accomplishing those objectives. We first reviewed the literature on expectations for developer support on social media. We then gathered data on 20 projects that have moved their developer support to Stack Overflow. We classified their reasons for the move into seven different categories. We chose two of the categories to measure, both before and after the move. The first factor, level of participation, we found to be higher in most cases after the move to Stack Overflow. The second factor, response time, we found to be much more favorable on Stack Overflow for two of the projects, although we were missing some data for two other projects. We also found some evidence of unintended consequences of moving to Stack Overflow, namely that certain types of questions are disallowed, thus undercutting the effectiveness of the support channel. This resulted in a subsequent "move back" or hybrid solution for a few projects. The impact of the work is that it is the first research to attempt an assessment of the quality of developer support provided by Q&A sites such as Stack Overflow, in comparison with previous tools used for developer support. We provide some guidance for software project leaders who are trying to determine whether moving to Stack Overflow is worthwhile to achieve their intended developer support objectives.

#### Acknowledgment

We would like to thank Jordan Tigani and Christian Funkhouser from Google, Inc. and Nick Olano from Credit Suisse for their valuable contributions with drafts of this paper, including feedback about providing developer support via



Stack Overflow, and helping with the Stack Overflow database dumps.

#### Appendix: Moving Announcements

A1. Microsoft Pex and Moles (later Fakes) Link to announcement: <http://goo.gl/6pyG8n> (shortened link)

A2. Facebook API. Link to announcement: <https://developers.facebook.com/blog/post/545/>

A3. Flattr API. Link to announcement: <http://blog.flattr.net/2011/09/forum-moving-over-to-stack-overflow/>

A4. Foursquare API. Link to announcement: [https://groups.google.com/d/msg/foursquare-api/rmS0DbKKOHo/CAM\\_KUxuMgMJ](https://groups.google.com/d/msg/foursquare-api/rmS0DbKKOHo/CAM_KUxuMgMJ)

A5. Google App Engine. Link to announcement: [https://groups.google.com/d/msg/google-appengine/Z6XN\\_64cA7w/Jf1OcUK4cF0J](https://groups.google.com/d/msg/google-appengine/Z6XN_64cA7w/Jf1OcUK4cF0J)

A6. Google Maps API. Link to announcement: <https://groups.google.com/d/msg/google-places-api/MppSAw8qs5Y/YtedMy1x32YJ>

A7. Soundcloud API. Link to announcement: [https://groups.google.com/d/msg/soundcloudapi/Ri6ddd3Hj\\_8/7LA1ZlW-W8AJ](https://groups.google.com/d/msg/soundcloudapi/Ri6ddd3Hj_8/7LA1ZlW-W8AJ)

A8. Google BigQuery. Link to announcement: <https://groups.google.com/d/msg/bigquery-discuss/z5Z-aRS9z8I/StEdfXojyg8J>

A9. Shopify API. Link to announcement: <https://groups.google.com/d/msg/shopify-app-discuss/G7eyzQIbX6w/R3AbsQsSFTwJ>

A10. Youtube API. Link to announcement: <http://apiblog.youtube.com/2012/09/the-youtube-api-on-stack-overflow.html>

A11a. Dwolla API. Link to forum announcement of the move: [https://getsatisfaction.com/dwolla/topics/dev\\_support\\_moving\\_to\\_stack\\_overflow](https://getsatisfaction.com/dwolla/topics/dev_support_moving_to_stack_overflow)

A11b. Dwolla API. Link to blog announcement of the move: <http://blog.dwolla.com/moving-to-stack-overflow/>

A11c. Dwolla API. Link to subsequent blog announcement of the re-creation of the discussion board: <http://blog.dwolla.com/the-evolving-dwolla-community/>

A12. Google Plugin for Eclipse. Link to announcement: [https://groups.google.com/d/msg/google-plugin-eclipse/K2p9Hz3OF88/k\\_X7rgnoBhgJ](https://groups.google.com/d/msg/google-plugin-eclipse/K2p9Hz3OF88/k_X7rgnoBhgJ)

A13. Breeze API

Link to announcement: [http://www.ideablade.com/forum/forum\\_posts.asp?TID=3814&title=breeze-moved-to-stack-overflow](http://www.ideablade.com/forum/forum_posts.asp?TID=3814&title=breeze-moved-to-stack-overflow)

A14. Docusign API. Link to announcement: <http://community.docusign.com/t5/Announcements/DEV-ZONE-MOVING-TO-STACK-OVERFLOW/td-p/20909>

A15. Gracenote API. Link to announcement: <https://developer.gracenote.com/sticky-moving-stack-overflow-3>

A16. Sagepay API. Link to announcement: <http://www.sagepay.co.uk/support/support-forum>

A17. Cosign API. Link to announcement: <https://www.arx.com/forum/showthread.php?328-Moving-to-Stack-Overflow>

A18. Google Cloud SQL. Link to announcement: [https://groups.google.com/d/msg/google-cloud-sql-discuss/IpAz85uZaBs/Rhx\\_Hr\\_wAN1wJ](https://groups.google.com/d/msg/google-cloud-sql-discuss/IpAz85uZaBs/Rhx_Hr_wAN1wJ)

A19. Sony. Link to announcement: <http://developer.sonymobile.com/2014/03/06/developers-get-support-from-sony-engineers-on-stack-overflow/>

A20. Socrata. Link to announcement: <http://support.socrata.com/entries/55829467-Technical-question-Post-it-on-StackOverflow->

A21. OpenXava. Link to announcement of move, and update about the move back: <http://www.openxava.org/en/blog/moving-support-to-stackoverflow>

#### References

- [1] Storey, M.A., L. Singer, B. Cleary, F.F. Filho, & A. Zagalsky. The (R)Evolution of social media in software engineering. In *Proceedings of Future of Software Engineering (FOSE14)*. May 31-June 7, 2014. Hyderabad, India. pp. 100-116.
- [2] Benslimane, D., S. Dustdar, & A. Sheth. Services Mashups: The New Generation of Web Applications. *IEEE Internet Computing*, 12(5). 2008. pp. 13–15.
- [3] Alexa.com. 2014. Stackoverflow.com Site Overview. Available at: <http://www.alexa.com/siteinfo/stackoverflow.com>. Last accessed June 12, 2014.
- [4] M4N. Is it okay to use Stack Overflow as the support forum for a product or project? In Stack Exchange Meta. June 9, 2009. Available at: <http://meta.stackexchange.com/questions/3966/is-it-okay-to-use-stack-overflow-as-the-support-forum-for-a-product-or-project>. Last accessed June 2, 2014.
- [5] RIAstar. 2012. Encyclopedia Stack Exchange vs. commercial products. In Stack Exchange Meta. May 24.

- Available at: <http://meta.stackexchange.com/questions/133522/encyclopedia-stack-exchange-vs-commercial-products>. Last accessed June 2, 2014.
- [6] Paul. 2013. Product support redirects to Stack Overflow. In Stack Exchange Meta. Sept. 3. Available at: <http://meta.stackexchange.com/questions/195558/product-support-redirects-to-stack-overflow>. Last accessed June 2, 2014.
- [7] Zoritchak, G. 2011. Community tools. On Apache Wicket mailing list. Oct. 6. Available at: <http://markmail.org/message/57r4oeqo4n2dvill> Last accessed June 13, 2014.
- [8] Narendran. 2010. StackExchange for web2py community. On Web2Py Google Group. Sept 16. Available at: <http://markmail.org/message/jc5g3bfarl3hsg53> Last accessed June 13, 2014.
- [9] Shaw, N.C., DeLone, W.H., & F. Niederman. Sources of dissatisfaction in end-user support: An empirical study. *The Database for Advances in Information Systems*. 33(2). 2002. pp. 41-56.
- [10] Chai, K., Wu, C., Potdar, V., & P. Hayati. 2011. Automatically measuring the quality of user generated content in forums. In *Proceedings of AI 2011*. LNAI 7106. pp. 51-60.
- [11] Zhu, Z., Bernhard, D., & I. Gurevych. 2009. A multi-dimensional model for assessing the quality of answers in social Q&A sites. *Technical Report TUD-CS-2009-0158*. Technische Universitat Darmstadt. 14 pages.
- [12] Gamalielsson, J., B. Lundell, & B. Lings. 2010. Responsiveness as a measure for assessing the health of OSS ecosystems. In *Proceedings of the 2nd International Workshop on Building Sustainable Open Source Communities (OSCOMM 2010)*. 8 pages.
- [13] Parnin, C., & C. Treude. 2011. Measuring API documentation on the web. In *Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering*. ACM. pp. 25-30.
- [14] Treude, C., O. Barzilay, MA. Storey. 2011. How do programmers ask and answer questions on the web? (NIER track) In *Proceedings of the International Conference on Software Engineering (ICSE'11)*. ACM. May 21-28. Honolulu, HI, USA. pp. 804-807.
- [15] Mamykina, L., B. Manoim, M. Mittal, G. Hripcsak, & B. Hartmann. 2011. Design lessons from the fastest Q&A site in the west. In *Proceedings of CHI 2011*. ACM. May 7-12. 9 pages.
- [16] Vasilescu, B., A. Serebrenik, P. Devanbu, V. Filkov. 2014. How social Q&A sites are changing knowledge sharing in open source software communities. In *Proceedings of Computer Supported Cooperative Work (CSCW'14)*. Feb 15-19. Baltimore, MD, USA. pp. 342-354
- [17] Squire, M. 2015. Docusign Scraper. Available at: <https://github.com/megansquire/stackpaper2015>
- [18] Raymond, E. 1999. *The Cathedral and the Bazaar*. O'Reilly & Associates.