

# Measuring API Documentation on the Web

Chris Parnin  
Georgia Institute of Technology  
College of Computing  
Atlanta, GA USA  
chris.parnin@gatech.edu

Christoph Treude  
University of Victoria  
Dept. of Computer Science  
Victoria, Canada  
ctreude@uvic.ca

## ABSTRACT

Software development blogs, developer forums and Q&A websites are changing the way software is documented. With these tools, developers can create and communicate knowledge and experiences without relying on a central authority to provide official documentation. Instead, any content created by a developer is just a web search away. To understand whether documentation via social media can replace or augment more traditional forms of documentation, we study the extent to which the methods of one particular API — jQuery — are documented on the Web. We analyze 1,730 search results and show that software development blogs in particular cover 87.9% of the API methods, mainly featuring tutorials and personal experiences about using the methods. Further, this effort is shared by a large group of developers contributing just a few blog posts. Our findings indicate that social media is more than a niche in software documentation, that it can provide high levels of coverage and that it gives readers a chance to engage with authors.

## Categories and Subject Descriptors

D.2.7 [Software Engineering]: Distribution, Maintenance, Enhancement—*Documentation*

## General Terms

Measurement

## Keywords

api documentation, social media, crowd documentation

## 1. INTRODUCTION

For better or worse, many software developers increasingly depend on web search to reference code documentation, seek code examples, or learn about programming. When any answer is a just a web search away, for those seeking, just exactly who or where that answer came from matters less and less.

But it was not always this easy. Not long ago, the information that developers could access was limited to paper manuals, massive documentation files, or other busy colleagues. There are many efforts to promote the creation and maintenance of good documentation, however, all too often documentation is absent or incomplete. When documentation is written, it quickly becomes stale. This stagnation is often the root of mistrust, which can lead to documentation being rarely consulted in practice [9]. Surveys of many software companies [8] have found documentation not only to be ignored, but to be inconsistent in quality and coverage.

Even as documentation produced by companies and API developers comes online, developers are starting to look elsewhere. Indeed, according to a recent survey of over 3000 MSDN developers [18], developers indicate they learn about new APIs primarily through web search; further, developers are more likely to find an answer on a blog than from another colleague. Born out of widely accessible infrastructure and social media technology — wikis, blogs, and web forums have blossomed with rich content and social interaction.

Unlike *community documentation* where a person may contribute to documentation of an open source project [3], with *crowd documentation* through social media, the individual contribution does not matter as much as the aggregate result. Because a web search can find any online resource, the process of learning from crowd documentation is indiscriminate on where or who that information comes from. For instance, on stackoverflow.com, a programmer can ask a question about almost any diverse technical area, and receive a detailed response within 10 minutes median [10]. Not only is the original programmer satisfied, but any future programmer can view a community voted and curated set of answers when they have the same question. For developers who contribute in this manner, there is almost no barrier to entry, no community vetting, or formal processes that would be associated with community documentation.

To examine whether crowd documentation has the potential of replacing more traditional forms of documentation, in this paper we report on a study aimed at measuring the effectiveness and completeness of API documentation via social media. We selected one particular API — jQuery — and performed web searches for each of the API methods. We then examined the first 10 results for each API method and analyzed the different information sources available to developers. We found that 87.9% of the API methods are covered by software development blogs, and that these blog posts mainly consist of tutorials and experiences using those API methods.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Web2SE'11, May 24, 2011, Waikiki, Honolulu, HI, USA  
Copyright 2011 ACM 978-1-4503-0595-2/11/05 ...\$10.00

## 2. BACKGROUND AND RELATED WORK

Effectively documenting and using APIs is not trivial. Robillard [16] observed insufficient or inadequate examples and issues with the API's structural design as obstacles for developers trying to learn an API. He also identified several other task, format and design related obstacles.

Several tools have been proposed to help with API documentation. Jungloids [11] are based on the idea that programmers often know what type of object they need, but not how to get to it. Jungloids support this process by automatically discovering how to convert from a set of initial types to a desired type. Jadeite [20] takes an alternative approach of displaying the most common way to construct a desired type by letting users add new “pretend” classes or methods that are displayed in the actual API documentation and can be annotated with the appropriate APIs to use.

Researchers have also considered improving API documentation by adding information on API usage. Holmes and Walker [6] used data on the popularity of API methods to recommend certain methods. Mica [19] augments standard search results to help programmers find API methods and examples of their usage by analyzing the content of web pages and classifying the results. Assieme [5] is a web search interface that supports programming search tasks by combining information from .jar files, API documentation and pages with sample code and explanations to help programmers reduce the number of queries they have to run.

Research on social media to support software development processes is not yet far advanced. Following the success of Web 2.0 [13] and its lightweight collaboration and communication mechanisms, researchers have started to ask how the “architecture of participation” [14] can support the social aspects of software development. In a web-based survey for software developers, Black *et al.* [2] found that social media can enable better communication throughout the software development process.

Websites such as delicious<sup>1</sup>, facebook<sup>2</sup> and wikipedia<sup>3</sup> now have software development counterparts. Dogear [12] introduces social bookmarking for large enterprises. Using Codebook [1], developers can become friends with work artifacts, keep track of dependencies and discover connections using a web interface. Annoki [21] supports collaboration in software development through enhanced wikis.

Research on the use of blogs by software developers is still limited. A few researchers have started to investigate the role of blogs in requirements engineering. Park and Maurer [15] investigate the role that blogging can play in generating a product vision and they discuss four types of strategies for expressing requirements in blogs. As observed by Seyff *et al.* [17], end-user involvement in software engineering is an ambivalent topic. They present a tool to enable end-users to blog their needs.

In many companies, blogging has found its way into corporate culture. In a study from 2007, Efimova and Grudin [4] describe emergent blogging practices in a corporate setting. They found blogging in the enterprise to be an experimental, rapidly evolving terrain in which balancing personal and corporate incentives and issues is one of the challenges that bloggers face. Huh *et al.* [7] conducted a similar study and

found that the corporate blogging community allows access to tacit knowledge and that it contributes to new forms of collaboration within the enterprise.

To date, there has been no study on API documentation via social media. In this paper, we examine the coverage of one particular API on the Web.

## 3. RESEARCH QUESTIONS

Our research questions focus on the coverage of API methods on the Web and on the characteristics of the corresponding resources. When our initial analysis revealed that blog posts played a prominent role in the search results, we focused on blog posts for our more detailed research questions.

- Where are API methods documented on the Internet and how high is the API coverage of different kinds of resources?
- How are API methods covered using blog posts?
- Who writes blog posts about API methods?
- How much interaction do blog posts about API methods attract?
- How frequently are source code snippets used in API related blog posts?

## 4. METHODOLOGY

This sections describes the methodology we used to answer our research questions.

### 4.1 Data Collection

To retrieve the resources that developers have at their disposal when doing a web search for API documentation, we performed web searches for all methods of the jQuery API. At the time of our study, there were 173 API methods available. We performed the web searches using Google, but made sure to be signed out of our personal accounts to avoid personalization of the search results. All queries were prefaced by “jQuery”, e.g. to retrieve the results for jQuery's `.add()` method, we searched for “jQuery add”.

We then extracted the links to the first 10 search results for each method, yielding a total of 1,730 links. For each search result, we recorded the query that found it, the rank it had on the result page, the link and the domain.

The data is available for other researchers. The data includes: an xml snapshot of the jQuery documentation, cached html search results, and a database of annotated search results.

### 4.2 Data Analysis

We analyzed all 1,730 websites in our data using a mix of quantitative and qualitative methods. In a first step, we classified them using criteria such as the URL (e.g., for links starting with `stackoverflow.com`), or the content of the website. Both authors of this paper coded a subset of the links to check for consistency in our coding, and then divided the rest of the data set up for individual coding.

In a second step, we did additional qualitative coding of all 376 unique blog posts in our data set. We used the same coding process as in the first step, but focused on the content and style of the blog post. In addition to a label for the type of blog post, we also recorded the number of code snippets used and the number of comments for each post.

<sup>1</sup><http://www.delicious.com/>

<sup>2</sup><http://www.facebook.com/>

<sup>3</sup><http://www.wikipedia.org/>

## 5. FINDINGS

### 5.1 Coverage by Different Kinds of Resources

SEARCH RESULT TYPE	COVERAGE	MEAN RANK
code snippet site	8.7%	9
q&a	9.8%	9
forum	20.2%	8
official bug tracker	21.4%	3
mailing list entry	25.4%	7
official documentation	30.1%	3
official forum	37.0%	3
unofficial documentation	63.6%	6
stackoverflow	84.4%	6
<b>blog post</b>	<b>87.9%</b>	<b>5</b>
official API	99.4%	1

Table 1: API coverage by different web resources.

The classification of the top 10 search results for all API methods revealed several different categories of web resources. The most frequent categories are given in Table 1 along with the extent to which they cover the different methods in the API and their mean page rank. The following paragraphs describe the different types of resources.

**Official API.** Part of the official API documentation, either documenting one of the API methods or listing several API methods belonging to the same category. For 99.4% of the API calls, the official API documentation was part of the first 10 search results and usually appeared on top.

**Blog Post.** A blog post, belonging to a blog with regular posts and a commenting feature. We found that there were blog posts among the top 10 search results for 87.9% of the API methods. Blog posts usually appeared right after the official documentation, and before Q&A websites, mailing list entries, and unofficial documentation. Details on the characteristics of the blog posts are given in the next section.

**Stackoverflow.** A Q&A exchange between developers on stackoverflow.com appeared on the first page of the search results for 84.4% of the calls.

**Unofficial Documentation.** Unofficial method documentation found on websites other than the official website of the API came up for 63.6% of all API calls (often cloned).

**Official Forum.** For 37.0% of the API calls, threads in the developer forum hosted on the API website were part of the top 10 search results.

**Official Documentation.** Official tutorials and other documentation material hosted on the API website only appeared for 30.1% of the API calls.

**Mailing List Entry.** Websites serving as entry point to the archive of a mailing list were part of the search results for 25.4% of the API calls.

**Official Bug Tracker.** For 21.4% of the API calls, a bug in the official bug tracking system came up in results.

**Forum.** A thread in a developer forum not hosted on the official API website appeared for 20.2% of the API calls.

**Q&A.** A Q&A exchange on a website other than stackoverflow.com appeared for 9.8% of the API calls.

**Code Snippet Site.** Code snippet sharing websites such as <http://www.codesniper.com/> had a coverage of 8.7%.

Other notable resources that appeared were the web-based hosting site GitHub and documentation in languages other

than English. We were surprised how infrequently official documentation appeared in comparison to other resources.

### 5.2 Different Kinds of Blog Posts

POST TYPE	FREQUENCY
design idea	5
repost	5
workaround	5
announcement	6
research	6
opinion	12
link / referral	24
new functionality	27
code snippet	35
experience	99
tutorial	217

Table 2: Types of blog posts.

Our following research questions focus on the nature of the blog posts used for API documentation. The qualitative coding of all blog posts in our data revealed the categories shown in Table 2. The most common blog post types were tutorials and experience reports by the blog authors. The following paragraphs describe the different post types.

**Tutorial.** The post contextualizes a problem or task and then proceeds to describe how to achieve a solution in a step by step manner.

**Experience.** The post documents development knowledge drawn from a recent experience<sup>4</sup>:

I spent over a half hour looking for the best solution to this. Personally, I blame the jQuery documentation. When reading over the jQuery core description it states, starting in version 1.4, that jQuery returns an empty set but offers no method to detect it. Ultimately, I found that `.length` is the way to go but I wanted to expound on all three methods I discovered.

**Code Snippet.** The post provides a succinct piece of code easily copied and extractable.

**New Functionality.** The post discusses new features in the latest release of a software.

**Link / Referral.** The post shares interesting news by linking to at least one other online resource.

**Opinion.** The post presents a technical opinion on an issue, design, or technology stack.

**Research.** The post describes an empirical investigation such as performance between different functions, or compatibility across different browsers for code.

**Announcement.** The post announces a new release or updates to a release of a software.

**Workaround.** The post presents a solution to an encountered problem.

**Repost.** The post duplicates content from another source (often stackoverflow) without adding additional content.

**Design Idea.** The post describes a novel technique or the merits of a particular approach for a problem.

### 5.3 Small Contributions from Many Authors

Many developers chipped in with documentation, even if they could only spare a few posts. Over half the posts (214)

<sup>4</sup><http://b-knox.com/181/detect-an-empty-set-in-jquery/>

came from authors contributing only 1 or 2 blog entries. In some sense, the documentation was truly coming from the *crowd*, with 210 authors accounting for a total of 376 blog posts.

Other authors gave more than their fair share, with the top five publishing between 10 and 26 posts. In Figure 1, we show the total distribution of posts by authors.

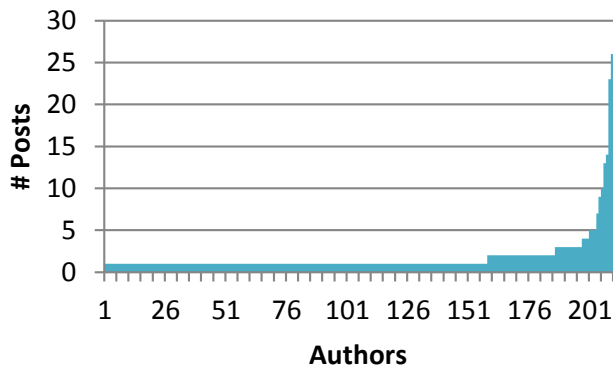


Figure 1: Number of posts contributed by authors.

#### 5.4 Readers have Conversations with Authors

Not all readers were passive, but actively engage in conversation with the authors in our data. 81% of the posts had at least one comment, with a median of 8 comments per post. In Figure 2, a clearer picture of the distribution can be seen (we removed one outlier with 882 comments).

Although we did not record the number of times authors replied, as we counted the total comments in a blog post, we did observe that authors frequently engaged with reader comments. Often readers responded with constructive criticism, and included their own improved code snippet, which the author might then incorporate into an updated version of the post. Other times readers solicited help for a similar problem; as a result, usually the author or another reader offered assistance. The conversations may not be as structured or focused as other social media sites such as stackoverflow, where research has found that discussion is mostly constrained to questions centered around code reviews, conceptual questions from novices, how-to questions, and questions about unexpected behaviors [22].

This result may not generalize to all posts or readers. The posts we examined might receive more traffic because they are highly ranked in web search results. It may be that other posts written by the same author may not receive as much traffic and thus fewer comments. Future studies can confirm this result by examining the comments received on all posts written by an author regardless of web ranking. Only a small percentage of readers may respond to posts. Without traffic logs from the blog sites, we cannot estimate the total size of readership that do engage in conversation.

#### 5.5 Blogs Talk about Code

As developers primarily write code, it is not too surprising that developers also talk about code in their blog posts. 90% of posts had code snippets in the post, a median of 3 code

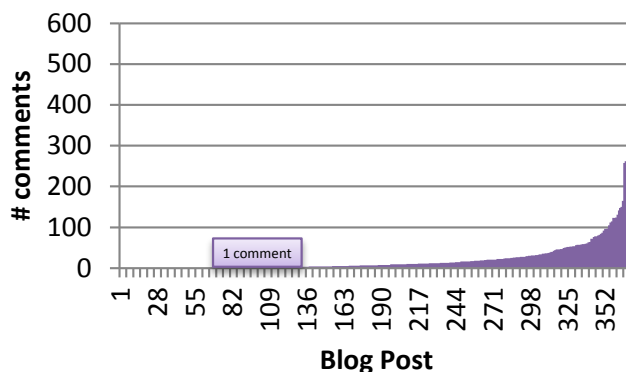


Figure 2: Number of comments per blog post.

snippets per post. That is 336 posts with a total of 1322 snippets of code. In Figure 3 the distribution is shown.

The presence of code snippets helps reinforce that posts are not rants or raves about the hardships of programming, but technical discussions sprinkled with code examples and buttressed with insights, motivations, and design trade-offs. This is interesting to contrast with other web results such as code snippet sharing sites. For the 15 code snippets we observed from [www.codesniper.com](http://www.codesniper.com), none had any explanatory text to explain the behavior or purpose of the code.

Now, we realize our methodology certainly biased the blogs we found to have code samples. We do not claim that the percentage of posts we found is representative of all types of blog posts written, just that discussions about technology are often illustrated with code examples and snippets.

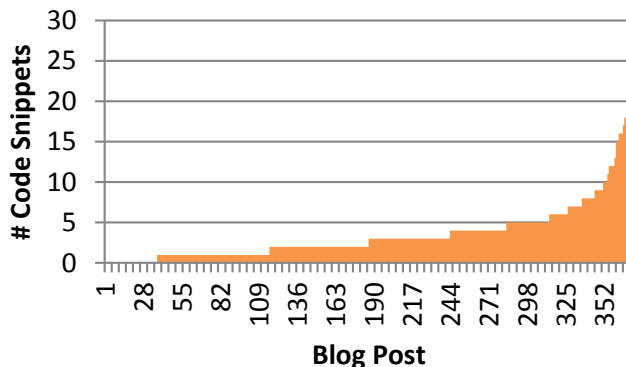


Figure 3: Number of code snippets per blog post.

## 6. DISCUSSION

### 6.1 Blogs as a Medium

What is conveyed by blog posts? After examining hundreds of blog posts about jQuery, the collective voice is something that would seemingly be discordant but is unexpectedly resonant and clear. Many purposes emerge that would

seem difficult for traditional documentation to emulate. We enumerate some different ways blog posts were used and the issues encountered.

**Philosophy.** Some posts did more than describe technical intricacies of coding, but delved into the philosophy of a particular design approach or problem. In one post, “The philosophy of handling ajax errors”, the author describes how they came to adopt a certain philosophy for handling ajax errors. This sort of writing provides real guidance for other less experienced developers in how to design their application to more gracefully handle a certain class of errors particular to web programming.

**Niche Communities.** Blogs also provided a way for niche communities to build their own support. In traditional API documentation, the scope is limited to describing one API in isolation. In practice, no one responsible party documents combinations that span more than one language or API. Here, blogs thrive at providing this sort of content. For example, we observed several examples of blog posts that included examples combining jQuery with ASP.NET or sharepoint sites (both Microsoft technologies).

**Personal Repository.** Some developers used their blog as a personal repository for information. For example, one developer describes how he found useful information comparing the speed efficiency of different API calls and wanted to save it for himself.

**Updates.** Finally, we found a couple instances where a post was marked as deprecated or updated. This raises an interesting issue: Should technical information on the web have expiration dates? With a centralized source distributing official documentation, more care can be taken in controlling if that information remains available. But with blog posts, out of date information can still persist. In one example, a post described an issue with using a jQuery function. Because the post lacked a publication date, the readers complained about not knowing if the information was still relevant or if it was addressed with the latest version of jQuery. One reader claimed the issue was still relevant, but in general, how can a reader trust this information?

## 6.2 Authors and Motivations

Why do authors make these public contributions? Who are they and what are their motivations? These are questions that arose when reading through hundreds of blog posts, but that we cannot answer from our data. We can give some impressions from reading through hundreds of posts.

There were a few bloggers that may have deliberately sought ad revenue but did offer legitimate content. These sites were plastered with various ads that interrupted the flow of the blog posts. The content was not copied from any apparent source, such as the jQuery documentation, but the explanation was a little terse. One such blogger had 9 posts comprising of several little tutorials, but did not go into much depth. Still his posts frequently appeared in search results and received some comments from readers.

In direct contrast with the previously described bloggers, were *powerhouse bloggers*. These bloggers wrote extensively, featured a clean and professional website, and had a large following from readers and subscribers. One such blogger also wrote many little tutorials; however, they were grounded from recent experiences, illustrated with real-world problems, and often were accompanied with video demonstrations that walked through and described the code.

The blogger’s 23 posts received many more comments (731 comments!) and actively participated in the discussion with his readers. As this blogger runs an independent consulting firm, his motivations may be different from that of simple ad revenue — building a community reputation or personal branding may be much more valuable.

Finally, there were 158 authors who each contributed a single post. It is difficult to explore the various motivations that comes from such a diverse crowd. We picked one such author at random for illustration. For this author, his contribution describes an experience in resolving a conflict with other javascript libraries, and how one of the jquery API options (noconflict mode) did not resolve the problem as advertised. The author posts his workaround. We do not know what benefit the author gained from sharing this experience, but one reader certainly shared his appreciation:

I ran into a conflict of 2 libraries, did a few hours of researching and this was the only solution that worked for me. I don’t quite understand it being new to jQuery, but it works! I’ll drop a line in my code comments back to your site. Much appreciated.

## 7. LIMITATIONS

In addition to the limitations noted in our findings, there are several general limitations with our methodology and study. We examine social media from the perspective of only one API and technology area. Although this provides a good starting point, the conclusions we draw may not hold in other technology areas. As we studied web resources discussing web programming (jQuery), we may have also attracted developers interested in using web technologies. Perhaps we would not find as extensive resources for embedded systems programming. Finally, established and extensive communities, such as developers using the Java JDK API, may involve different challenges or have different forces at play than developers in emerging or niche communities.

Web search can be limited by a mismatch of semantic intention between the search terms and results. For example the search term, “jquery first”, brings up blogs illustrating the function `first`, but also “my first jquery plugin”, which may be interesting or irrelevant to the searcher. We do not know if the search results are relevant for every hypothetical search, only that of their potential to answer a question.

Finally, we began our study with a limited set of search terms, namely the names of API functions from jQuery. Developers may use a variety of search terms to target the questions or problems they have about programming. This may explain why we did not encounter some social media sites such as reddit where programming discussions often occur. Future studies might also explore larger sets of search terms seeded by web suggestions or logging search terms used by developers in practice. Further, this data can be correlated with popularity of API methods.

## 8. CONCLUSION AND FUTURE WORK

Our findings indicate that social media plays an important role in software documentation, and that media such as blog posts can reach a high level of coverage featuring tutorials and personal experiences. Looking at coverage may validate the concept of crowd documentation, but it may also oversimplify its value. Many searches may try to find specific use cases not covered by any traditional form of documentation.

The combinatorial nature of programming may span multiple APIs that cannot be covered by one authority. Bloggers may be able to publish many forms of mix-and-match examples that bridge these boundaries and offer a personalized and task-relevant form of documentation.

The high occurrence of code snippets gives several implications: For those that *create* code snippets, little tool support exists for extracting and publishing code snippets. For those that *use* the code snippets; what happens if thousands of developers copied a code sample that is later updated to fix an security error? Do we need tool support for web provenance?

Future work will have to investigate the role of documentation found on the Web in more detail. We need to understand what can be done to help developers find documentation more effectively, and how tool support can help those creating documentation using social media. The high coverage of API methods by blog posts observed in this study is a first indicator that social media is more than a niche in software documentation and that it can reshape the way software knowledge is communicated.

## 9. REFERENCES

- [1] A. Begel, Y. P. Khoo, and T. Zimmermann. Codebook: discovering and exploiting relationships in software repositories. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 1, ICSE '10*, pages 125–134, New York, NY, USA, 2010. ACM.
- [2] S. Black, R. Harrison, and M. Baldwin. A survey of social media use in software systems development. In *Web2SE '10: Proceedings of the 1st Workshop on Web 2.0 for Software Engineering*, pages 1–5, New York, NY, USA, 2010. ACM.
- [3] B. Dagenais and M. P. Robillard. Creating and evolving developer documentation: Understanding the decisions of open source contributors. In *Proceedings of the 18th ACM SIGSOFT International Symposium on the Foundations of Software Engineering*, pages 127–136, November 2010.
- [4] L. Efimova and J. Grudin. Crossing boundaries: A case study of employee blogging. In *HICSS '07: Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, page 86, Washington, DC, USA, 2007. IEEE Computer Society.
- [5] R. Hoffmann, J. Fogarty, and D. S. Weld. Assieme: finding and leveraging implicit references in a web search interface for programmers. In *Proceedings of the 20th annual ACM symposium on User interface software and technology, UIST '07*, pages 13–22, New York, NY, USA, 2007. ACM.
- [6] R. Holmes and R. J. Walker. A newbie's guide to eclipse apis. In *Proceedings of the 2008 international working conference on Mining software repositories, MSR '08*, pages 149–152. ACM, 2008.
- [7] J. Huh, L. Jones, T. Erickson, W. A. Kellogg, R. K. E. Bellamy, and J. C. Thomas. Blogcentral: the role of internal blogs at work. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 2447–2452, New York, NY, USA, 2007. ACM.
- [8] M. Kajko-Mattsson. A survey of documentation practice within corrective maintenance. *Empirical Softw. Engg.*, 10(1):31–55, 2005.
- [9] T. C. Lethbridge, J. Singer, and A. Forward. How software engineers use documentation: The state of the practice. *IEEE Software*, 20:35–39, 2003.
- [10] L. Mamykina, B. Hartmann, B. Manoin, and M. Mittal. Design lessons from the fastest q&a site in the west. In *CHI '11: Proceeding of the 29th conf. on Human factors in computing systems*, 2011.
- [11] D. Mandelin, L. Xu, R. Bodík, and D. Kimelman. Jungloid mining: helping to navigate the api jungle. In *Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation, PLDI '05*, pages 48–61. ACM, 2005.
- [12] D. R. Millen, J. Feinberg, and B. Kerr. Dogear: Social bookmarking in the enterprise. In *Proceedings of the SIGCHI conference on Human Factors in computing systems, CHI '06*, pages 111–120, New York, NY, USA, 2006. ACM.
- [13] S. Murugesan. Understanding web 2.0. *IT Professional*, 9:34–41, July 2007.
- [14] T. O'Reilly. What is Web 2.0: Design patterns and business models for the next generation of software, 2005. <http://oreilly.com/web2/archive/what-is-web-20.html>.
- [15] S. Park and F. Maurer. The role of blogging in generating a software product vision. In *CHASE '09: Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, pages 74–77, Washington, DC, USA, 2009. IEEE Computer Society.
- [16] M. P. Robillard. What makes apis hard to learn? answers from developers. *IEEE Softw.*, 26:27–34, November 2009.
- [17] N. Seyff, F. Graf, and N. Maiden. End-user requirements blogging with irequire. In *ICSE '10: Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering*, pages 285–288, New York, NY, USA, 2010. ACM.
- [18] J. Stylos. Msdn programming help resources survey. Personal Communication.
- [19] J. Stylos and B. A. Myers. Mica: A web-search tool for finding api components and examples. In *Proceedings of the Visual Languages and Human-Centric Computing*, pages 195–202, Washington, DC, USA, 2006. IEEE Computer Society.
- [20] J. Stylos, B. A. Myers, and Z. Yang. Jadeite: improving api documentation using usage information. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems, CHI '09*, pages 4429–4434, New York, NY, USA, 2009. ACM.
- [21] B. Tansey and E. Stroulia. Annoki: a mediawiki-based collaboration platform. In *Proceedings of the 1st Workshop on Web 2.0 for Software Engineering, Web2SE '10*, pages 31–36, New York, NY, USA, 2010. ACM.
- [22] C. Treude, O. Barzilay, and M.-A. Storey. How do programmers ask and answer questions on the web? (nier track). In *Proceedings of the 33rd ACM/IEEE International Conference on Software Engineering - Volume 2*, 2011.