

Design Lessons from the Fastest Q&A Site in the West

Lena Mamykina¹, Bella Manoim², Manas Mittal³, George Hripcsak¹, Björn Hartmann³

1—Columbia University Medical Center
Department of Biomedical Informatics
lena.mamykina@dbmi.columbia.edu
hripcsak@columbia.edu

2—Bard College
bm458@bard.edu

3—University of California, Berkeley
Computer Science Division
{mittal,bjoern}@cs.berkeley.edu

ABSTRACT

This paper analyzes a Question & Answer site for programmers, Stack Overflow, that dramatically improves on the utility and performance of Q&A systems for technical domains. Over 92% of Stack Overflow questions about expert topics are answered — in a median time of 11 minutes. Using a mixed methods approach that combines statistical data analysis with user interviews, we seek to understand this success. We argue that it is not primarily due to an a priori superior technical design, but also to the high visibility and daily involvement of the design team within the community they serve. This model of continued community leadership presents challenges to both CSCW systems research as well as to attempts to apply the Stack Overflow model to other specialized knowledge domains.

Author Keywords: Q&A, mixed methods analysis

ACM Classification Keywords: H5.3. Group and Organization Interfaces: Web-based Interaction.

General Terms: Design, Human Factors

INTRODUCTION

Individuals increasingly rely on their distributed peer communities for information, advice, and expertise. Millions of individuals learn from each other on public discussion forums (e.g., Usenet), community-built encyclopedias (e.g., Wikipedia), social networks (e.g., Aardvark), and online question and answer sites (e.g., Yahoo! Answers). Recently, several large Q&A sites have attracted the attention of researchers [2,4,8,10,11,15,21]. In aggregate, these studies suggest that general-purpose Q&A sites have answer rates between 66% and 90%; often attract non-factual, conversational exchanges of limited archival value; and may be poorly suited to provide high quality technical answers.

This paper describes a popular Q&A site for programmers and software engineers, Stack Overflow (SO) and analyzes factors in the site's design and evolution that contributed to

its success. Within two years, SO has become one of the most visible venues for expert knowledge sharing around software development. With approximately 300,000 registered users and >7 million monthly visits (as of August 2010), SO has an answer rate above 90% and a median answer time of only 11 minutes. The site has captured significant mindshare among software developers: anecdotally, users report that the site has replaced web search and forums as their primary resource for programming problems; others now consider their portfolio of SO answers a valuable component of their professional resumes. This community “buzz” about SO's success prompted our investigation.

How might we understand the factors behind this success?

We first conducted a statistical data analysis of the entire SO corpus to understand usage patterns. We investigated answer time, user types, suitability for different question types, and possible extensions of the SO model to other domains. To ground this aggregate view in concrete user experiences, we also conducted a qualitative interview study with users and the design team. The authors are not affiliated with the site. This mixed method approach is shared with prior work [21]; interviews with site designers are, to our knowledge, novel in studies of Q&A sites.

Consistent with prior work, we found that certain features of the SO design were critical to its effective functioning as a Q&A service: fast answer times and high answer quality arise from a carefully crafted reputation system and a strict set of community guidelines that favor factual, informational answers. *However, our analysis also demonstrated that these features were a consequence of a particular design philosophy and organization espoused by its founders.* In short, the design team is strongly and publicly involved in both control of and debate within the community. This involvement is made possible by the site's focus on a single domain in which the design team had prior standing as community leaders. In contrast, many large-scale self-organizing Q&A sites are broad in reach—the site operators supply a general platform for question answering but are not directly involved in either content creation or moderation (i.e., the developers are external to the communities they support). This tight engagement with the community led to three factors that we believe were critical to the success of SO.

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

Copyright 2011 ACM 978-1-4503-0267-8/11/05...\$10.00.

1) *Making competition productive*: Since the founders and the design team of SO were active members of the software development community, they had a clear view of both needs and driving forces among its members. Tight focus on technical answers enabled by the Q&A format and a voting system created a strong alternative to the more conversational software forums. Adding game mechanics through a reputation system harvested the competitive energy of the community and led to intense short participation for some users, and long sustained participation for others.

2) *Credibility in the community*: Thought-leader status and visibility within their community allowed the founders to gather a critical mass of dedicated users even before the site was introduced, thus ensuring its success early on. It also helped them to get their community on board with the proposed design and editorial practices. While many in the community lament the lack of possibilities for discussion and debate, they continue to uphold the founders' vision because they acknowledge the resulting benefits.

3) *Evolutionary approach to design*: Finally, the design team established a continuous feedback loop with their users. A forum for discussion about the site, but external to it (meta.stackoverflow.com) helped the founders understand challenges and concerns of their users, and prioritize feature requests. (Users on meta vote for feature requests the same way they vote for questions). These requests were addressed through rapid design iterations with new releases of the site introduced almost daily.

Other similarly successful knowledge sharing communities, such as Slash(dot) (a news site covering advances in technology with user-contributed content) or TuDiabetes (a site for individuals with diabetes), also had founders who not only provided the tools, but also actively shaped their communities. This pattern raises a challenge for HCI researchers. The SO approach is predicated on ongoing, deep community involvement and, simultaneously, continuous technical adaptation of their software platform. This model conflicts with the canonical process of human-centered design, in which it is more typical for technology developers to have intense, short periods of interaction with perspective users early on but then step down once the tools are introduced. The questions for HCI researchers are twofold: *Is it possible for outsiders of a community to foster knowledge sharing with the right set of tools? And what impact can CSCW systems research make without long-term community involvement?*

In the remainder of this paper, we position our contribution with respect to related work; we then support our assertion of SO's success through a data analysis and comparison to other Q&A sites. Next, we distill themes that emerged from our interviews with site designers and users; and conclude with a discussion of implications for research in social computing systems.

RELATED WORK

Prior work has investigated popular question answering sites (including Stack Overflow) as well as complementary approaches to expertise sharing. We review each in turn.

Analyses of Popular Q&A Sites

The blueprint for Q&A sites was established by Ackerman's Answer Garden [1], which focused on expert answers in a single domain. Recent Q&A platforms operate at Internet scale and often strive for generality. Their size has led to the creation of new analysis methods — some driven by data, others by qualitative studies.

Methodology: Structural Analyses Capture Aggregate Use

One class of research relies predominantly on analyses of Q&A data sets. For example, network analysis algorithms *e.g.*, HITS [16], have been used to characterize user activity and identify users with high expertise for the Java Forum [27] and Yahoo! Answers [4,10]. Network analysis has also been used to discriminate sub-communities on Yahoo! Answers [2]. Prior work that uses the Stack Overflow data set conducted quasi-experiments about impacts of design decisions through post-hoc database analyses [22]; and considered Stack Overflow as an example of a two-sided market [18]. This paper also applies data analysis to describe the performance of Stack Overflow; this analysis then guides our qualitative study of SO's design process.

Qualitative and Mixed Method Studies Focus on Individuals

To better understand individual users' experiences, Nam et al. used interviews and data analyses to investigate usage patterns and user motivations of a popular Korean Q&A site, KiN [21]. Dearman surveyed users to find out why they *don't* answer certain questions [8]. Torrey exclusively used interviews to find patterns of seekers of craft knowledge online [26]. We also rely on user interviews but additionally report on interviews with founders. We next summarize trends highlighted by Q&A studies.

Q&A On Other Sites Is Often Not About Factual Knowledge

Several distinct types of questions on Q&A sites can be distinguished: *factual* (seeking objective data); *advice*, (seeking recommendations); *opinion*, (seeking others' viewpoints), and *non-questions* (spam) [11,15,21]. Significant parts of general Q&A sites are *conversational*; the sites perform poorly on focused technical questions [21]. Algorithms to distinguish between *informational* and *conversational* threads have been proposed [11,25]. We do not investigate such distinctions as SO explicitly (and successfully) discourages conversational contributions.

Sites Leverage Both Intrinsic and Extrinsic Motivations

Motivations of individuals who contribute answers to Q&A sites can be categorized as either intrinsic (altruism, the desire to learn) or extrinsic (gaining status, monetary rewards) [14,21,23]. Point systems and other game mechanics are frequently used extrinsic motivators. Adding monetary rewards can transform the user's sense of the system from a social interaction space to a more formal transaction space [14]. Stack Overflow has several highly

effective extrinsic motivating factors: a reputation and badge system that rewards activity; and public profiles, which demonstrate a user's expert knowledge to her community of peers [24].

Moderation Policies Shape Use

Researchers have studied patterns of community moderation on knowledge sharing sites. Lampe and Resnik [19] looked at distributed moderation practices on Slash(dot). Active members of this site earn privileges that include the ability to cast votes that either increase or decrease a post's prominence and visibility. Stack Overflow relies heavily on community moderation. Active participants can vote questions and answers of others up and down and recommend closing inappropriate questions.

Alternatives: Forums, Social Search and Specific Tools

Questions about programming are also posed and answered in other online media such as mailing lists, newsgroups, and internet relay chat. When contributions are ordered temporally in active forums, it may become difficult to locate relevant posts. Research has proposed ways to find relevant subsets of posts, *e.g.*, through collaborative filtering [17]; and to build improved models of discussion structure, *e.g.*, by tracking quotations [3].

In *social Q&A*, questions are directed to the asker's online social network; this choice trades off size of the answer pool for social proximity. Answers are normally not aggregated in a common knowledge base. Social Q&A may leverage existing platforms—for example, people ask questions through Facebook or Twitter status messages [20]. Aardvark, a social Q&A service [13] routes questions to the most relevant users who are online at the moment a question is asked; neither questions nor answers are shared with the community. Researchers have hypothesized that services like Aardvark are especially suited for eliciting opinions and subjective answers from trusted sources.

Most Q&A and social search systems (including Stack Overflow) rely on plain text messages. For questions about programming, researchers have also integrated techniques for finding help, examples, and debugging strategies directly into programming environments [5,12]. Such systems have not yet seen widespread adoption, making it premature to compare their benefits to text-based Q&A.

HOW STACK OVERFLOW WORKS

Stack Overflow follows a common model of Q&A site design: users post questions, answer questions, comment, and vote on posts. Posts can be tagged. Users can edit their own prior submissions. Figure 1 shows the main page of questions, with the most recently asked questions at the top; Figure 2 shows a question page with answers and comments underneath.

Users may register on the site to gain reputation points and badges, as determined by user activity and votes by others on a user's posts. With higher reputation scores users are awarded editing rights on the site, including editing other users' answers. The site is dedicated to factual answers and



Figure 1: Stack Overflow's list of recent questions.

When to prefer JSON over XML?

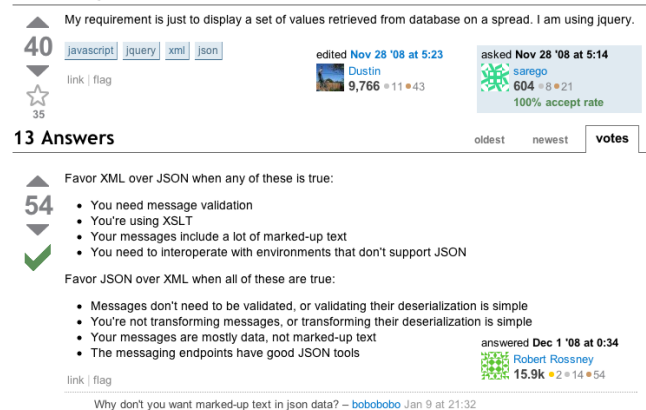


Figure 2: An individual SO question page (layout compressed).

explicitly discourages subjective, conversational topics or discussions. To enable conversations about the site itself without interfering with the main Q&A function, Stack Overflow has a “meta” site, comparable to discussion pages on Wikis.

Moderation is achieved through voting (which orders contributions) user edits (to avoid stale information), and through actions of official moderators. Moderators can modify, close and delete posts and user profiles. Moderators are elected by SO users in a formal, annual vote.

HOW WELL DOES STACK OVERFLOW PERFORM?

This section summarizes the results of a data analysis we conducted on two years of user activity on Stack Overflow — from July 31, 2008 to July 31, 2010. We used the publicly available Stack Overflow database export files¹.

Hundreds of Thousands, but not Millions of Users

As of early August 2010, Stack Overflow has a total of 300k registered users who asked 833k questions, provided 2,2M answers, and posted 2,9M comments. In August 2010 the site served 7.8 million monthly visitors. This makes Stack Overflow smaller than general Q&A sites, but larger than social Q&A or programming forums (Table 1).

¹ <http://blog.stackoverflow.com/category/cc-wiki-dump/>

Site	Users	Total Posts	Posts/day (last month)	Source
Stack Overflow	300,534	833,427 Q 2,225,456 A	2226 Q 4573 A	This paper
Aardvark	90,361	225,047 Q 386,702 A	3167 Q	Horowitz [12]
Java Dev. Forum	13,379	333,314 messages	?	Zhang [25]
K-iN	?	60 million total	44,000 Q 110,000 A	Nam [19]
Yahoo! Answers	?	23 million resolved A	39,299 Q 281,745 A	Adamic [2]
Live QnA	290,000	600,000 Q 1,800,000 A	?	Hsieh [13]

Table 1: Comparison of multiple Q&A sites.

92.6% of Questions are Answered; Most Multiple Times

Most questions are answered: 92.6% of questions receive at least one answer. This rate exceeds rates reported for Yahoo! Answers (88.2% [11]) and KiN (~66% [21]). More importantly, the answers are predominantly technical. 63.4% of questions receive strictly more than one answer. Since SO focuses on *informational* questions, we expect fewer answers than in sites that permit *conversational* questions. This intuition is supported: Harper reported 5.71 answers per question for Yahoo! Answers [11]; SO has 2.9.

However, each post can also have a list of comments associated with it. Comments are used for follow-up and statements of agreement or disagreement. We analyze the *thread length* of a question by summing answers and all comments. The complete distribution of answers and thread lengths is shown in Figure 3. Accounting for comments, the average thread length is similar to Yahoo! (mean=6.7).

Answers are Fast: First Answers in 11 Minutes, Accepted Answers in 21 Minutes (Medians)

How long do users have to wait until they receive answers? We considered three different definitions of answer times, based on the time elapsed until: 1) the first answer is posted; 2) an answer is posted that eventually receives a positive vote (“upvoted”); 3) the answer is posted that is eventually accepted by the questioner. Fewer questions have accepted answers as accepting is not required.

The median time for a first answer is only 11 minutes (Figure 4): half of all questions that eventually receive answers are answered within 11 minutes. The median time for upvoted answers is 10:52 minutes (questions with upvoted answers are a subset of questions with answers; hence a faster time is possible); for accepted answers it is

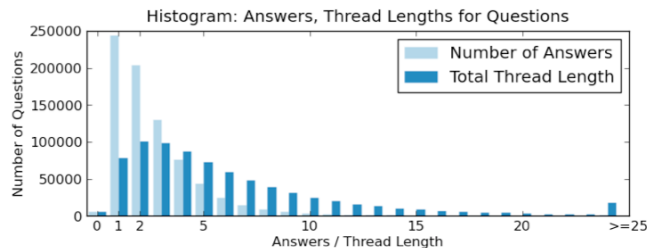


Figure 3: Answers and thread lengths for all questions.

21:10 minutes. Expanding our analysis to all questions, including those that are never answered, yields that 50% of all questions receive a first answer within ~12 minutes; an upvoted answer within 25 minutes, and the accepted answer within approximately 6 hours. This is an astonishing result— one interviewee remarked:

“If you complained about a lack of response on a newsgroup after 24 hours you were labeled impatient; now you can realistically expect an answer within 20 minutes.” (User 4)

Average reported times for other Q&A sites are: 2h 52min for first answers on Live QnA; 1h 14min for mimir [14] (both for small subsets of questions). Medians were not reported. Social Q&A site Aardvark is faster (median answer time: 6 min 37 sec [13]), due to the fact that Aardvark routes questions to users known to be online.

Most Answer Activity Takes Place in the First Hours

While first answers are fast in the median, relatively few additional questions are answered after the first hours (see Figure 4, Figure 5). There is also a long tail of questions that remain unanswered: the mean time for first answers, which is heavily skewed by these long-latency responses, is 2 days and 10 hours.

Answers Have Been Fast Since Early On

How did this rapid answer time emerge over the site’s history? Median first answer times have been essentially flat since the site’s inception in fall 2008; median times for all answers and for accepted answers have been around 20

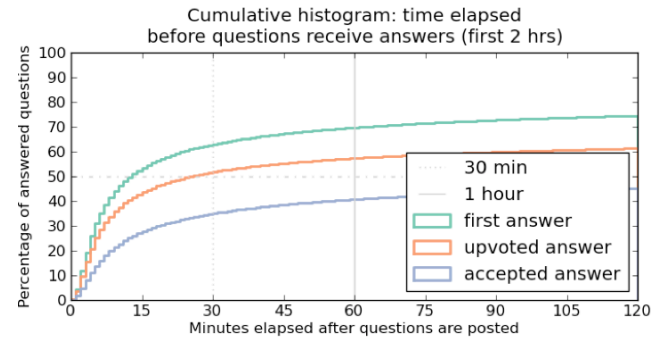


Figure 4: Answers in the 2 hours after questions are posted.

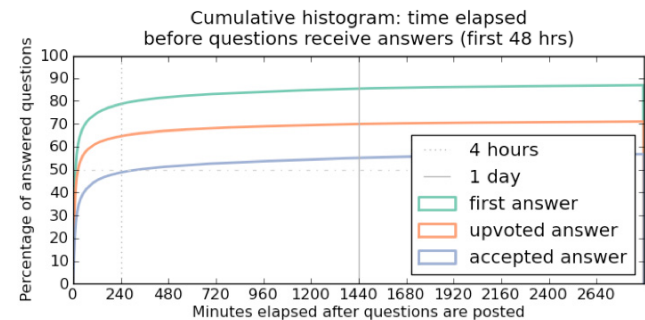


Figure 5: Answers in the first 2 days: little activity after 4 hrs.

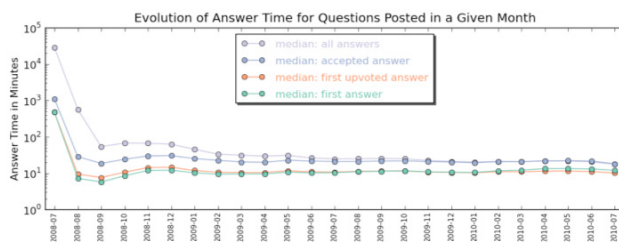


Figure 6: Answer time has been consistent for many months.

minutes since summer 2009, even as traffic on the site increased over time (Figure 6).

This trend suggests that motivating users to frequently return and participate is more important than the total number of users. It also suggests that *Stack Overflow has been operating at a response time minimum and that further improvements in response time are unlikely*. 10 minutes appears to be the minimum time for a knowledgeable programmer to find a question, read and think about it, formulate a reply, and publish that reply.

Askers and Answerers Overlap

There are 300,534 registered users. Of these, the largest group of roughly one quarter have not yet asked, answered, or voted on any questions. The second most frequent group consists of users who only ask, but do not answer or vote on questions (23.7%), followed by users who only answer, but never ask or vote (20.4%). Overall, nearly half (48.5%) of registered users have answered questions. The overlap of users who ask and answer is significantly larger ($16.4 + 7.7 = 24.1\%$) than in Nam's analysis of KiN (5.4%), and approximates Gyöngyi's analysis of Yahoo! Answers.

Frequent Users Post More Answers Than Questions

Figure 7 shows that user activity follows a power law: most users have very little activity, and the number of users with higher activity falls off exponentially (*i.e.*, as a linear relationship in a log-log plot). *Infrequent users* (low x values in Figure 8) *post more questions than answers*. For these users, the median ratio of answers to all posts is at or below 0.5. In contrast, *frequent users overwhelmingly tend to have high answer ratios*, *i.e.*, they answer more questions than they ask, with the exception of a few outliers.

Four Answer Behaviors: Community Activists, Shooting Stars, Low-Profile Users, and Visitors

The unique success of Stack Overflow can be understood in terms of the ecology of different user behaviors it enables. We identified four distinct groups of users, based on the frequency with which they provide *answers* in the system. We distinguish between *low-activity* users (< 20 answers in a given month) and *high-activity* users (≥ 20 answers). Each user has an *activity signature* that describes their activity month-to-month (e.g., signature *LLLH* denotes three months of low activity, followed by one month of high activity). We found four types of signatures:

Community Activists: Registered users who are highly active on the site for multiple months.

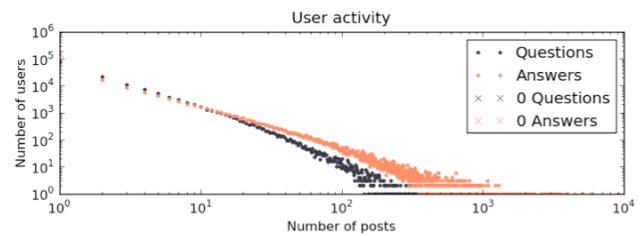


Figure 7: User activity conforms to a power law.

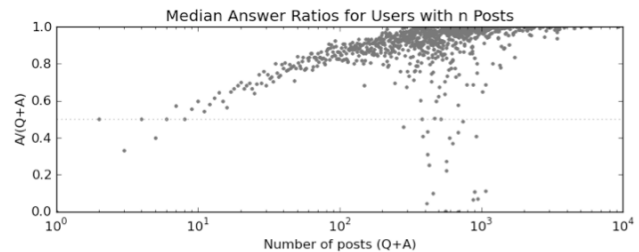


Figure 8: Frequent users (right) mostly post answers ($Y > 0.5$).

Shooting Stars: Registered users who have a single, short period of high activity followed by low activity.

Low-Profile Users: Registered users who have intermittent activity, but who never become highly active.

Lurkers and Visitors: Users who have not been asking or answering questions; visitors without user accounts.

We calculate the percentage of users in each class, and the percentage of answers that are supplied by those users, using regular expression matching on activity signature strings. 94.4% of users are never highly active; they supply 34.4% of answers; *shooting stars* make up 4.2% of the user base and supply 21.9% of answers; *community activists* make up 1% of users but supply 27.8% of answers (Figure 9). The remaining 15.9% of answers are provided by non-registered users or users that do not fit these profiles.

We hypothesize that the game mechanics of the site draw in both community activists and shooting stars, but convert only the first group into highly active contributors. The second group moves on after a short infatuation period.

Questions Receive Dozens of Views—Mostly Early on

Users who visit SO without ever creating an account are largely invisible to our analysis. Anecdotally, many of these individuals find answers to previously asked questions through search engines. Future work could quantify the size of this user group through web server log analysis; we note that the site receives 7 million monthly visitors, but has only 300,534 registered users. Figure 10 shows the distribution of views for questions, which includes both logged-in users and visitors. Most questions receive dozens to hundreds of views (mode: 34); few questions receive thousands to tens-of-thousands of views. For most questions, the majority of views occur early after a question has been posted (highest points in curves roughly align); however, a smaller number of questions continues to collect

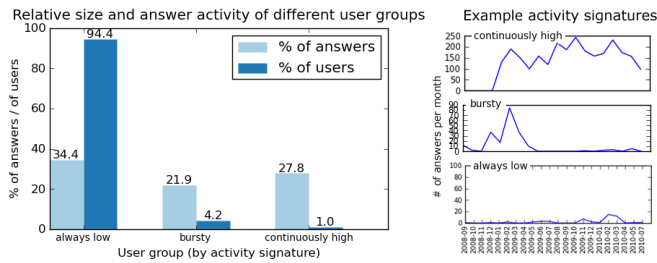


Figure 9: Left: Most answers are provided by 3 distinct user groups: low-profile users, shooting stars, and community activists. Right: Example activity signatures for these groups.

views over many months — these are the few outlier questions that are most interesting to visitors over time.

Some Questions are not Supported Well by SO

The design choices made by Stack Overflow bring tradeoffs with them: certain types of questions are better suited to be asked and answered on SO than others. Our interviewees hypothesized about several classes of questions that remain unanswered or are answered slowly:

1. Questions about relatively *obscure technologies* for which there are few users.
2. Questions that are *tedious* to answer.
3. Problems that *cannot be easily reproduced* with a small, self-contained code fragment.
4. Questions that do not have a clear best answer and thus invite *discussion*, even if that discussion is technical.

Thus far, our data analyses have only been able to confirm the first hypothesized reason. Figure 11 shows answer times for a selection of 30 tags that occur frequently for fast and slow questions, respectively. Fast tags on the left tend to cover widely used technologies (c#, php, .net); while slow tags on the right are more obscure (ireport, amqp, wif). Attempts to characterize questions as slow or fast by analyzing question topic, question type, or term frequencies have been inconclusive.

Meta Site has a Small, Opinionated, Active Base

The SO meta site only has 6% of the users of the main site, suggesting that a vocal minority engages in discussion. Meta questions attract about twice as many comments per thread and the median time to first answer is only 8:29

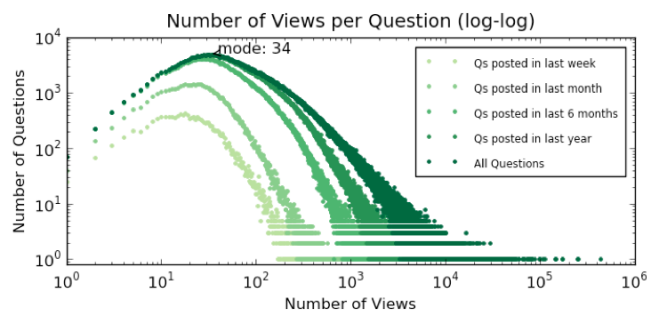


Figure 10: Questions receive dozens of views from visitors.

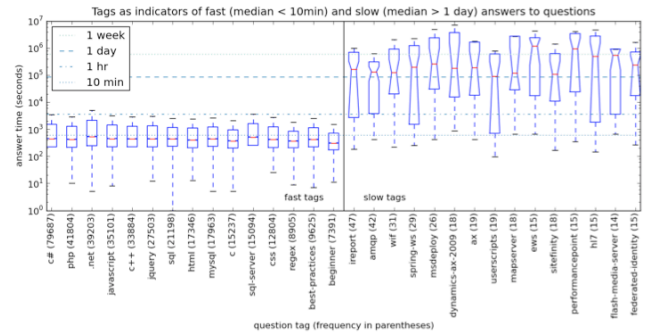


Figure 11 Left: 15 Tags with fast median response times (<10min); Right: 15 tags with slow response times (>1 day).

minutes. These effects are mainly attributable to the discussion-focused nature of meta questions: stating an opinion may take less time than formulating a technical answer; it may also to elicit more responses in turn.

The SO Model Might Extended to Other Domains

In addition to the main SO site, there are two sibling sites that utilize the same platform, Server Fault (for system administrators) and Super User (for computer enthusiasts). These sites receive approximately 1/10th of the traffic of SO, but have similar answer times and ratios. In recent months, an additional crop of sites has been created. Detailed data about these offshoots was not available; we report summary data in Figure 12. Even the most active of these sites are a factor of 100 less active than Stack Overflow. Interestingly, a site dedicated to *subjective* questions about programming is now among the most active offshoots. This suggests that it is more important to draw community boundaries narrowly, with precise definitions what is “in bounds” and “out of bounds” for a given site. The meaning of those boundaries may matter less.

QUALITATIVE STUDY

Methods

In the previous section we discussed the patterns of questions and answers that emerged on SO over time. This analysis showed SO to be largely successful in accomplishing its primary goal: giving software developers fast, informative answers to their questions. To better understand the driving factors behind these patterns we conducted a qualitative study of the community. Participants of the study included SO founders (n=2), members of the site design team (n=4) and users (n=6).

	Users	Questions	Answers	A/Q	% answered
Stack Overflow	300,534	833,427	2,225,456	2.7	92.6%
Web Apps	4,600	1,200	4,700	3.9	90%
Gaming	3,600	2,300	5,100	2.2	96%
Programmers	3,200	642	5,000	7.8	98%
Cooking	2,500	1,600	5,600	3.5	100%
Ubuntu	2,400	1,200	3,200	2.7	94%
Game Developers	2,200	765	3,100	4.1	98%
Mathematics	2,000	1,300	3,500	2.7	93%
Webmasters	2,000	758	2,300	3.0	97%
Unix and Linux	1,900	567	1,600	2.8	94%

Figure 12: Summary statistics for the top 10 Stack Exchange sites (collected 9/20/2010; A/Q = mean answers per question).

Participants among the design team were recruited using leads from site founders. Users were recruited based on their reputation level on SO, creating a mix of top users and moderately active users. The study included interviews conducted over the phone, on Skype, and in person, and lasted for about 1 hour. We used a semi-structured interview format following general themes, but exploring emergent topics in conversations. The themes for users included: 1) Users' motivation, adoption and usage patterns, 2) types of questions asked and answered, 3) interactions with members of the SO community, 4) comparison to alternative Q&A formats and forums. These questions elicited both strengths and limitations. The themes for founders and designers included: 1) early design choices, 2) evolution of the SO platform over time, 3) current state and future directions. The interviews were recorded and transcribed verbatim. We used inductive iterative coding [6] to allow common themes to emerge from the data.

Findings

Our analysis yielded a number of design choices our interviewees perceived as critical to the success of SO, and several design strategies adopted by the site's founders that led to this success. These strategies include: 1) making competition productive, 2) building on existing credibility within the community, and 3) adopting a continuous evolutionary approach to design. In the next section we first discuss positive findings that we believe contributed to the success of the site; we then turn to challenges and barriers the site continues to experience.

Improving on Forums through Productive Competition

Stack Overflow is the result of a collaboration between two individuals well known within the software development community for their heavily-read blogs, Joel Spolsky and Jeff Atwood. Their main goal was to create a sustainable resource where anybody with a question pertaining to software development could quickly find "the right answer". Their design approach specifically *prioritized information over conversation* through a Q&A format and a voting system, and encouraged participation through a system of game incentives.

Information Instead of Conversation

Both founders were active participants in software development discussion forums. They witnessed many of the forum threads devolve into conversational spaces ridden with rants, spam and anti-social behavior, thus complicating search for valuable information [17]. To address this ongoing challenge, SO was conceived as a Q&A site rather than a discussion forum.

To help valuable pieces of concrete technical knowledge to become more visible, SO designers introduced a voting system, in which users earn rights to vote on posts of others through active participation. These community moderation mechanisms were previously explored in discussion forums, however they produced sub-optimal results:

"What it [voting system] was doing on those sites wasn't very successful because when you have comments and there's a conversation going on, if you say, 'Well, these are valuable comments and these are not valuable comments,' then the only way to get a valuable thing to read is to take everything that's highly voted. Then you're skipping interim conversation." (Founder 1)

In contrast to these systems, in SO answers are treated as discrete, independent pieces of information that can be reordered to express relevance. This choice had important consequences for the design of the system of external incentives and its impact on user engagement.

System of External Incentives

Consistent with previous research, our interviewees demonstrated a combination of intrinsic motivational factors, including a desire to help their community and learn, and extrinsic ones, for example a wish to enrich their professional portfolios or simply collect reputation points. Many of the early users of SO had extensive track records of educating their community through blogs, technical books, and active participation in software forums. However, all individuals we interviewed that actively participated, even the most established educators, described SO's system of external incentives as one of the main factors that "got them hooked" and kept them coming back:

"I am very competitive and you give me indication that a high number is good and I will try to get a high number. I don't think it's to do with reputation so much as, 'This is a game.'" (User 4)

Reputation or point systems are commonly used in social computing applications to encourage more proactive participation. In SO this strategy was highly effective. Many users set their goals on reaching the reputation cap—the maximum number of points one could earn in one day, and developed multiple tools and strategies to maximize their gains. Several interviewees compared their experience explicitly to games, where cleverly designed reward systems also produce dramatic effects:

"Stack Overflow — it's like World of Warcraft, only more productive." (User 5)

The drive to maximize reputation points had an unintended consequence described as the "Fastest Gun in the West" problem on the SO meta site. Providing faster brief answers gave users more reputation points than providing more detailed answers that took longer to write. As a consequence, the community's focus drifted somewhat away from optimizing the quality of information.

One aspect of the reputation system remains a contention point among site experts we interviewed. After reaching 10,000 points, an individual has all the moderation and editing privileges the site offers and can no longer benefit from increase in reputation. Based on our quantitative analysis of user participation patterns, we suggest that the reputation plateau is in fact the defining point in

establishing a user's activity signature. For users who relied primarily on external motivators, reaching the plateau led to a subsequent reduction in participation, creating the shooting star pattern. Users who additionally had strong internal motivation continued to actively participate and contribute, for the sake of the community.

Credibility in the Community

In addition to being active participants on discussion forums, both founders were active and prolific bloggers. In the summer of 2008, when SO was introduced, their respective blogs, *Joel on Software* (Joel Spolsky) and *Coding Horror* (Jeff Atwood) had a combined readership of approximately 140,000 people. This prominence gave the founders two unique advantages: the ability to gather a critical mass of dedicated users, and a high initial level of trust for their vision.

Achieving Critical Mass

All social software systems have to address the problem of building an initial critical mass of users [8]. The SO design team addressed this challenge *before* launching by discussing SO on their blogs and holding a series of weekly podcasts describing their vision, inviting readers to share their thoughts and provide feedback. As a result, when Stack Overflow was introduced, thousands of people were asking and answering questions within the first day:

"So on the first day, the first question I could come up with had already been asked and answered, and there were three or four answers, and some voting had happened. The best answer had already been voted to the top. So on the first day when I saw everything working, I knew that we were in really good shape." (Founder 2)

Acceptance and Negotiation of Founders' Vision

While discussion forums and news groups were popular with software developers, a strictly informational Q&A site was novel. Many early users of SO were skeptical of this approach.

"In the early days, there were a lot of people coming and saying, 'We want threading and we want instant messaging and we want user profiles like Facebook.'" (User 3)

However, since many of them respected the founders from their blogs, users were willing to give the SO model a try.

"You know, when you hear that these guys came up with something you wanna go check it out." (User 6)

Some early adopters and enthusiasts of the founders' vision became active advocates and community educators.

"And I would come in and start saying, 'Listen, this is why this works the way it does and why the other stuff doesn't work.' So I think that was my involvement, was sort of evangelizing sort of that new paradigm." (User 3)

However, some points of contention remain to this day. Many current users lament the lack of possibilities to engage in a debate over more controversial issues related to software development. This tension is also evident on the

site, where some discussion-oriented questions continue to be highly popular and visible (Figure 13).

Votes	Question Title
1109	What is the single most influential book every programmer should read?
908	What should a developer know before building a public website?
840	What is your favorite programmer cartoon?
802	Strangest language feature
797	I'm graduating with a Computer Science degree but I don't feel like I know how to program

Figure 13: Discussion-oriented questions among the 10 highest ranked questions on Stack Overflow (as of Sep. 21, 2010).

Evolutionary Approach to Design

Tight Feedback Loop with Users

While experienced in traditional software engineering processes, the Stack Overflow team took a different approach to design that is becoming more popular among software startups:

"We pretty much had to forget all the software engineering processes we learned." (Designer 2)

Instead of investing time in requirements analysis or user testing behind closed doors, the design team adopted a rapid prototyping approach driven by direct and immediate user feedback. Even before SO was designed and deployed, its prospective users became a significant guiding force, providing comments and often challenging the designers' vision. After introduction of the first version of the site, the feedback loop was formalized in a user forum, User Voice, later replaced by the SO Meta site. Meta used the same Q&A engine as Stack Overflow, but was meant to engage users in the discussion about the site, its features, editorial policies, and community values. Introduction of Meta gave the designers an opportunity to keep an ongoing discussion with their users. In addition, it moved conversational topics away from the main site, to preserving the high "signal-to-noise" ratio for technical information. The five most popular questions from Meta (Figure 14) are indicative of the variety of topics this site covers:

Votes	Post Title
771	The official FAQ for Stack Overflow, Server Fault, and Super User
626	Could we please be a bit nicer to the noobs
401	Using what I've learned from stackoverflow. (HTML Scraper)
365	Jon Skeet Facts? [Jon Skeet is the most active SO user]
356	Why aren't people voting for questions?

Figure 14: Highly ranked questions from the SO Meta site.

Rapid Design Iterations

The second factor that contributed to the success of the site was a particular design approach adopted by the founders and designers of the site. Specifically, they adopted a practice of constantly adjusting the design of the site and immediately releasing the new modifications to the community:

"We pretty much release new versions every day. Sometimes they are really small changes; the bigger ones often get announced on Meta." (Designer 1)

These rapid iterations allow for experimentation; the team can tweak the design and observe resulting changes in community behavior. For example, the “Fastest Gun in the West” problem we discussed earlier, often led to longer, more informative answers being buried under shorter but quicker ones. Changing the design so that answers with equal numbers of votes were presented in random order partially alleviated this challenge.

“When we saw that, we decided to change the sorting so that if two questions are tied, instead of just listing them in order of appearance (oldest first), we now sort in random order if they’re tied for the number of points. This radically affected how the community worked because it meant that the secondary questions that were really good had a much higher probability of getting voted up.” (Founder 2)

To this day, all main features of SO, including voting and reputation systems, continue to be adjusted by the design team and frequently the founders, who remain actively involved in its day to day operations.

Challenges

Despite site’s success, there remain many challenges hotly debated by the site’s designers and users. Some of them relate to the nature of the community the site generated. Our interviewees with high reputation levels were aware of the challenges experienced by the new members of the community who find it difficult to gain visibility and status. Disparities between novice and expert users are exemplified by the thread “Could we please be a bit nicer to the noobs”, which continues to be highly active on Meta (see Figure 14). Other concerns relate to the potential longevity and future success of the site. An ongoing discussion about “Stack Overflow Fatigue” among members reflects these threads. Yet another set of concerns is related to the practices that the design of the site enabled and encouraged. Earlier we mentioned “The Fastest Gun in the West” problem; similar challenge is related to the difference in point allocation between providing a new answer and correcting an earlier answer. Because of this difference, users feel less compelled to correct previous answers; as a consequence, suboptimal answers persist on the site if they were accepted as correct ones. These and other challenges continue to demand further refinement of the site and improvements to its design.

DISCUSSION

Design Matters

The study of Stack Overflow demonstrated that careful consideration of design decisions significantly contributed to the success of SO over prior and competing platforms. While few of SO’s features are truly novel, the details of their design effectively promoted desirable behaviors and discouraged undesirable ones.

For example, voting systems are common in many online forums. However, subtle adjustments such as randomization of answer order within sets of equal votes allowed the founders to reinforce their focus on technical answers over

discussions. Point systems are equally popular in social software. But careful tuning of the reputation allocation mechanisms within the point system reinforced the general design philosophy. For example, designating “community” questions that do not gain reputation points reduced incentives to ask overly broad questions.

Community Organization Matters More

At the same time, our study suggests that deep engagement and prominent membership in the community gave the SO founders important advantages. The nuanced insider’s perspective on needs and driving motivations of the community; the ability to attract a critical mass of users (in this case before launch); and the ability to continue this engagement throughout the system’s deployment have all been recognized as important contributors to the success of groupware applications [8]. While user-centered design methodology aims to increase the chance of attaining these goals when designing for others, SO demonstrates the power of designing from *within* a community. However, SO is also not a naive case of designing for oneself, as the continuous refinement based on user feedback attests to.

Cloning Success?

The overarching theme that emerged from our work is that tight engagement with the community was critical to the success of knowledge sharing within the SO community. To what extent then can the success of SO be replicated — even by the same design team — in other domains of specialized knowledge? An early experiment to license the SO Q&A engine for profit to third parties largely failed. Many of the communities started “with a credit card” but without appropriate leadership languished; the program was ultimately abandoned.

Currently, a second effort to broaden the reach of SO beyond programming is underway that attempts to improve on these prior missteps. New sites attempt to leverage the existing user base by largely focusing on leisure activities the current users already engage in (e.g., gaming, mathematics, cooking). The new approach includes introduction of a formal community proposal system in which communities must *define* clear topic boundaries; *demonstrate commitment* to the project; and show *viability* through sufficient user activity in a test phase. To our knowledge, this formalization of online community formation is novel; the outcome is still uncertain.

CONCLUSIONS

In this paper we discussed lessons learned from a popular Q&A site for software developers, Stack Overflow. Analysis of the patterns of user interactions with the site helped to highlight some of the more common user groups and behaviors. A qualitative study helped to explain these behaviors and shed some light on the reasons for the site’s success. These reasons include 1) founders’ tight involvement with the community, 2) highly responsive and iterative approach to design, and 3) a system of incentives that promoted desirable user behavior.

These findings present a number of questions to the HCI and CSCW communities in regards to the importance of socio-cultural context for the adoption of social computing platforms. While these questions are not new, new technology platforms for sites like SO bring unique challenges. If the success of social computing applications depends on both deep community engagement and, simultaneously, on continuous modification of those applications in response to shifting community behaviors, this presents a significant barrier to systems research in social computing. Some researchers already argue that the most innovative work in this area is not produced in academia [5]. Instead, startup companies, dedicated to their cause, and willing to maintain engagement with their user communities for extended periods of time appear to have a stronger position to iterate towards novel, successful models. Given these concerns, what role should (and can) systems research play in social computing? Academic research should not be relegated to studying exciting new communities created by others. But what is the right vector for introducing new tools in this space from research?

REFERENCES

1. Ackerman, M.S. and Malone, T.W. Answer Garden: a tool for growing organizational memory. *Proceedings of the ACM SIGOIS and IEEE CS TC-OA conference on Office information systems*, ACM (1990), 31-39.
2. Adamic, L.A., Zhang, J., Bakshy, E., and Ackerman, M.S. Knowledge sharing and yahoo answers: everyone knows something. *Proceeding of WWW 2008*, ACM (2008), 665-674.
3. Barcellini, F., D tienne, F., Burkhardt, J., and Sack, W. Thematic coherence and quotation practices in OSS design-oriented online discussions. *Proceedings of GROUP 2005*, ACM (2005), 177-186.
4. Bouguessa, M., Dumoulin, B., and Wang, S. Identifying authoritative actors in question-answering forums: the case of Yahoo! answers. *Proceeding of ACM SIGKDD 2008*, ACM (2008), 866-874.
5. Brandt, J., Dontcheva, M., Weskamp, M., and Klemmer, S.R. Example-centric programming: integrating web search into the development environment. *Proceedings of CHI 2010*, ACM (2010), 513-522.
6. Brooks, F.P. *The Design of Design: Essays from a Computer Scientist*. Addison-Wesley Professional, 2010.
7. Corbin, J. and Strauss, A.C. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, Inc, 2007.
8. Dearman, D. and Truong, K.N. Why users of Yahoo! Answers do not answer questions. *Proceedings of CHI 2010*, ACM (2010), 329-332.
9. Grudin, J. Groupware and social dynamics: eight challenges for developers. *Communications of the ACM* 37, 1 (1994), 92-105.
10. Gy ngyi, Z., Koutrika, G., Pedersen, J., and Garcia-Molina, H. Questioning Yahoo! Answers. *Proceedings of the First Workshop on Question Answering on the Web (at WWW 2008)*, (2008).
11. Harper, F.M., Moy, D., and Konstan, J.A. Facts or friends?: distinguishing informational and conversational questions in social Q&A sites. *Proceedings of CHI 2009*, ACM (2009), 759-768.
12. Hartmann, B., MacDougall, D., Brandt, J., and Klemmer, S.R. What would other programmers do: suggesting solutions to error messages. *Proceedings of CHI 2010*, ACM (2010), 1019-1028.
13. Horowitz, D. and Kamvar, S.D. The anatomy of a large-scale social search engine. *Proceedings of WWW 2010*, ACM (2010), 431-440.
14. Hsieh, G. and Counts, S. mimir: a market-based real-time question and answer service. *Proceedings of CHI 2009*, ACM (2009), 769-778.
15. Hsieh, G., Kraut, R.E., and Hudson, S.E. Why pay? Exploring how financial incentives are used for question & answer. *Proceedings of CHI 2010*, ACM (2010), 305-314.
16. Kleinberg, J.M. Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46, 5 (1999), 604-632.
17. Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., and Riedl, J. GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM* 40, 1997, 77-87.
18. Kumar, R., Lifshits, Y., and Tomkins, A. Evolution of two-sided markets. *Proceedings of the third ACM international conference on Web search and data mining*, ACM (2010), 311-320.
19. Lampe, C. and Resnick, P. Slash(dot) and burn: distributed moderation in a large online conversation space. *Proceedings of CHI 2004*, ACM (2004), 543-550.
20. Morris, M.R., Teevan, J., and Panovich, K. What do people ask their social networks, and why?: a survey study of status message q&a behavior. *Proceedings of CHI 2010*, ACM (2010), 1739-1748.
21. Nam, K.K., Ackerman, M.S., and Adamic, L.A. Questions in, knowledge in? A study of Naver's question answering community. *Proceedings of CHI 2009*, ACM (2009), 779-788.
22. Oktay, H., Taylor, B.J., and Jensen, D.D. Causal Discovery in Social Media Using Quasi-Experimental Designs. *Proceedings of the ACM/SIGKDD Workshop on Social Media Analytics*, (2010).
23. Raban, D.R. and Harper, F.M. Motivations for Answering Questions Online Abstract. *New Media and Innovative Technologies*, (2008).
24. Raymond, E.S. Homesteading the Noosphere. In *The Cathedral and the Bazaar*. O'Reilly, 1999.
25. Rodrigues, E.M. and Milic-Frayling, N. Socializing or knowledge sharing? Characterizing social intent in community question answering. *Proceedings of CHI 2009*, ACM (2009), 1127-1136.
26. Torrey, C., Churchill, E.F., and McDonald, D.W. Learning how: the search for craft knowledge on the internet. *Proceedings of CHI 2009*, ACM (2009), 1371-1380.
27. Zhang, J., Ackerman, M.S., and Adamic, L. Expertise networks in online communities: structure and algorithms. *Proceedings of WWW 2007*, ACM (2007), 221-230.