

# Building Reputation in StackOverflow: An Empirical Investigation

Amiangshu Bosu, Christopher S. Corley, Dustin Heaton, Debarshi Chatterji, Jeffrey C. Carver, Nicholas A. Kraft  
Department of Computer Science  
The University of Alabama  
Tuscaloosa, AL 35487-0290, USA  
{asbosu, cscorley, dwheaton, dchatterji}@ua.edu, {carver,nkraft}@cs.ua.edu

**Abstract**—StackOverflow (SO) contributors are recognized by reputation scores. Earning a high reputation score requires technical expertise and sustained effort. We analyzed the SO data from four perspectives to understand the dynamics of reputation building on SO. The results of our analysis provide guidance to new SO contributors who want to earn high reputation scores quickly. In particular, the results indicate that the following activities can help to build reputation quickly: answering questions related to tags with lower expertise density, answering questions promptly, being the first one to answer a question, being active during off peak hours, and contributing to diverse areas.

**Index Terms**—Mining repositories, StackOverflow, reputation

## I. INTRODUCTION

StackOverflow (SO) is the most popular community for obtaining answers to software development questions and is a rapidly growing base of information about topics ranging from algorithms to languages to tools. Building reputation within the community is a key motivator for contributing to SO, and contributors' reputations are quantified by scores based on their answers to questions posed on SO. A high reputation score recognizes a contributor's expertise and earns the contributor added privileges, and affords the contributor more trust from other community members.

A contributor builds his/her reputation score by providing answers that are accepted by the SO community. To earn a high score, a contributor must share technical expertise via a sustained effort. Moreover, to have answers accepted by the community, a contributor must compete with other reputation seekers and with highly reputed contributors whose answers may be trusted more by the community. Thus, earning a high reputation score requires a contributor to quickly provide high-quality answers. As of August 2012, 443K of the 1.3M registered SO users had answered at least one question. Of those 443K contributors, only a small number (5,932) had earned a reputation score greater than 5000.

To identify potential paths to a high reputation score, we empirically evaluated the SO data and identified the:

- 1) Strongest topic areas — help a reputation seeker understand the level of effort required in his interest area(s)
- 2) Most reputed contributors and their impacts in different topic areas — help a reputation seeker understand the level of competition in his interest area(s)

- 3) Times of day/week with fewer active contributors — help a reputation seeker target less competitive times
- 4) Contribution styles of the 10 fastest contributors to earn reputation scores of at least 20,000 — help a reputation seeker to understand how other contributors quickly earned high reputation scores

The paper is organized as follows. §II defines the study metrics and describes the analysis method, §III describes the analysis results, and §IV discusses implications of the results.

## II. RESEARCH METHOD

We used the SO data provided by the MSR 2013 challenge [1]. We defined the following metrics to measure the efficiency of the SO community.

**Accepted Ratio:** *Percentage of questions with an answer accepted by the question submitter.*

**Unanswered Ratio:** *Percentage of questions with at least one answer, but no answers up-voted up at least once.* We use this definition rather than *questions with no accepted answers* because the original question submitter may simply forget to accept an answer.

**No-Response Ratio:** *Percentage of questions with no answers.*

**First Answer Interval:** *Time elapsed between the postings of a question and its first answer.*

**Accepted Answer Interval:** *Time elapsed between the postings of a question and its accepted answer.*

Table I summarizes the metric values for the SO data.

TABLE I  
STACKOVERFLOW METRICS

Metric Name	Value
Accepted Ratio	62.21%
Unanswered Ratio	21.18%
No-Response Ratio	8.69%
Median First Answer Interval	14.98 minutes
Median Accepted Answer Interval	23.57 minutes

For the remainder of the paper, when analyzing the First Answer Interval and the Accepted Answer Interval over a series of questions or users, we use *median* as the central tendency due to the skewness of the data.

## III. RESULTS

The following subsections describe our analysis method and results based on the four perspectives identified in §I.

TABLE II  
TOP CATEGORIES OF STACKOVERFLOW

Category	% of Ques.	Accept. Ratio	Unans. Ratio	Median 1st Ans.	Top Areas
.NET	18.5%	65.0%	19.02%	12.97	c#, asp.net, .net, vb.net, wcf
Java	16.1%	58.7%	23.54%	15.77	android, java, eclipse
Web	15.2%	64.3%	20.27%	9.13	javascript, jquery, html, css
LAMP	13.2%	62.1%	21.02%	9.42	php, mysql, arrays, apache
C/C++	9.5%	66.5%	13.16%	9.55	c, c++, windows, qt
OOP	6.5%	67.1%	15.46%	13.90	oop, image, performance, delphi
iOS	5.9%	61.6%	24.15%	21.75	iphone, ios, objective-c
Databases	5.5%	67.6%	14.81%	10.08	sql, sql-server, database
Python	4.6%	67.9%	13.88%	15.63	python, django, list
Ruby	3.5%	65.9%	20.35%	29.20	ruby, ruby-on-rails
Strings	3.2%	72.0%	10.92%	7.90	regex, string, perl
MVC	2.0%	68.2%	17.97%	17.70	asp.net-mvc, mvc
Adobe	1.2%	57.6%	27.32%	52.30	flex, flash, actionscript
SCM	0.8%	68.9%	13.31%	13.62	git, svn

### A. Areas of Expertise

A contributor whose expertise is related to topics about which a large number of questions are asked has ample opportunity to earn reputation points quickly. There are 122 tags that have over 10,000 associated questions. These tags cover 86% of all questions. We used the Gephi [2] implementation of Blondel et. al.’s community detection algorithm [3] to cluster these tags into related categories. For this social network analysis we constructed a weighted undirected graph in which nodes represent tags and edge weights are based on the numbers of questions shared between tags. Using a resolution value of 0.35 [4] resulted in 14 tag categories. We labeled each category using the tag represented by the node with the most edges. Table II summarizes these results.

Qualitative analysis of the tags reveals some interesting patterns. Although five categories (.NET, Java, C/C++, Python, Ruby) relate to object-oriented languages, a distinct category is dedicated to object-oriented programming (OOP). In addition, two categories (Web and LAMP) relate to web development, and with the exception of C/C++, each language-specific category includes tags for the language’s web framework(s). These observations suggest that OOP and web development are the most prominent topics on SO.

### B. Levels of Expertise Available in Different Areas

The top contributors for each tag, based upon their score relative to that tag, are recognized by bronze, silver or gold badges<sup>1</sup>. A gold badge requires a total score of at least 1,000 in at least 200 non-community wiki answers<sup>2</sup>, a silver badge requires a total score of 400 on 80 answers, and a bronze badge requires a total score of 200 on 20 answers. We define an expert contributor as a contributor who has earned at least one gold or silver badge.

As of August 2012, 806 contributors had earned at least one gold badge, and 2,040 contributors had earned at least one gold or silver badge. These 2,040 contributors total about 0.5% of all contributors who have answered at least one question. This small group of experts is effective; they have contributed approximately 29% of the posted answers and approximately 32% of the accepted answers. These experts are also efficient;

their median answer time is 12.23 minutes (versus a median answer time of 24.45 minutes for all contributors). Further analysis showed that these experts were even quicker (median answer time 11.72 minutes) answering questions related to their areas of expertise (based upon silver and gold badges).

The c# tag has the highest number of experts: 153 contributors with gold badges and 337 with silver badges. However, c# also has the highest number of questions, so the total number of expert users alone may not be a reliable indicator of the available expertise in an area. Therefore, we computed the ratio of expert users to posted questions (*Experts-to-Questions* ratio) for 121 of the tags mentioned in §III-A (we excluded *homework*). To analyze the effect of the expert users, we calculated a series of Pearson’s correlations between Experts-to-Questions ratio and the efficiency metrics defined in §II. For Pearson’s,  $|r| < 0.3$  indicates small correlation,  $0.3 \leq |r| < 0.5$  indicates medium correlation, and  $|r| \geq 0.5$  indicates strong correlation.

First, the Unanswered Ratio ( $r = -0.571$ ,  $p < 0.001$ ) and the No-Response Ratio ( $r = -0.487$ ,  $p < 0.001$ ) are negatively correlated with the Experts-to-Questions ratio, indicating that the availability of more expertise reduces number of unanswered questions and no-response questions. Second, the Accepted Ratio ( $r = 0.529$ ,  $p < 0.001$ ) and answers per question ( $r = 0.340$ ,  $p < 0.001$ ) are positively correlated with the Experts-to-Questions ratio, indicating that the availability of more expertise increases both the quality and the quantity of answers. Third, the median of the First Answer Interval ( $r = -0.129$ ,  $p = 0.157$ ) and the Accepted Answer Interval ( $r = -0.117$ ,  $p = 0.2$ ) are negatively correlated with the Experts-to-Questions ratio, suggesting that the availability of expertise may cause (a significantly insignificant) reduction of median first answer interval and median accepted answer interval.

Consider two tags that have no badged users: *sharepoint* and *wordpress*. These tags have two of the highest Unanswered Ratio values (32.76% and 39.53%, respectively) and high No-Response Ratio values (12.38% and 16.47%). These tags also have few answers per question (1.63 and 1.4), low acceptance percentages (50.28% and 47.98%), high median first answer intervals (122 and 15 minutes), and high median accepted intervals (210 and 25 minutes). Conversely, *scala* has the highest Experts-to-Questions ratio, a low No-Response Ratio (2.73%), and a low Unanswered Ratio (6.41%). It also has a high acceptance rate (77.66%) and many answers per question (2.16). Yet, median first answer interval (35 minutes) and median accepted answer interval (64 minutes) are high.

Notably, *android* has the 5<sup>th</sup> highest number of questions posted, yet has a high unanswered ratio (32%), a high median first answer interval (20.4 minutes), a high median accepted answer interval (30.22 minutes), and a low accepted ratio (52.5%). Among the 15 tags with the largest numbers of questions posted, SO is least efficient for *android*. SO also has low efficiency for *ios* (median first answer interval = 22 minutes, unanswered ratio = 28%, and accepted ratio = 58%), which suggests that SO might lack expert contributors in mobile development.

<sup>1</sup> <http://stackoverflow.com/badges?tab=tags>

<sup>2</sup> <http://meta.stackoverflow.com/questions/11740/what-are-community-wiki-posts>

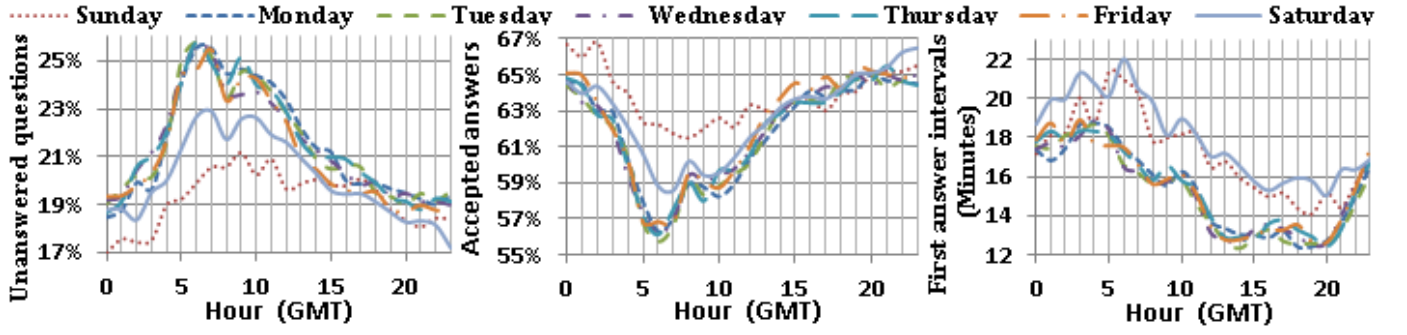


Fig. 1. Distribution of a) Unanswered Ratios, b) Accepted Ratios, and c) First Answer Intervals For Each Hour of the Week

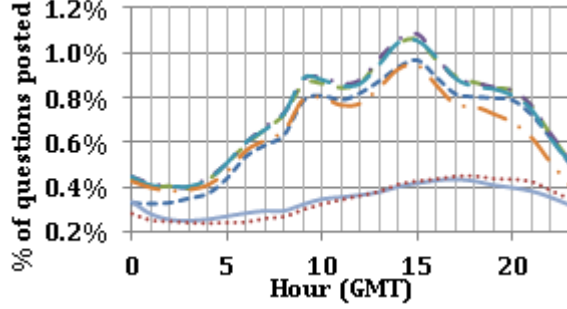


Fig. 2. Distribution of Percentage of Questions Posted Each Hour

Contributors interested in gaining reputation may be interested in the following areas, all of which have many posted questions and few experts (i.e., 25<sup>th</sup> percentile Experts-to-Questions ratio): flash, facebook, ipad, apache, excel, silverlight, eclipse, web-service, osx, xcode, and visual-studio-2010. Conversely, reputation seekers may want to avoid these areas, all of which have high Experts-to-Questions ratios: scala, r, delphi, c#, perl, php, c, c++, python, java, tsq, .net, javascript, jquery, git, and regex.

### C. Temporal Efficiency

To understand the best and worst times to obtain answers to questions, we analyzed the efficiency of SO for each of the 168 hours in a week. This analysis focused on variations in the Accepted Ratio, Unanswered Ratio, median First Answer Interval, and median Accepted Answer Interval. We can observe three time-based patterns from the results in Figure 1 (Note times are GMT — 5 hours ahead of the US East Coast).

First, while questions posted on the weekend are more likely to be answered than those posted during the week, it takes between two and five minutes longer to get that answer. Similarly, questions posted on the weekend are more likely to get an acceptable answer, but it may take up to ten minutes longer to get that answer.

Second, between 23:00 and 5:00 GMT the First Answer Interval is above 17 minutes and the Accepted Answer Interval is above 27 minutes. At the same time, users post the smallest number of questions during this time (see Figure 2). Taken together, these observations indicate that these hours are the *SO off-peak hours*. The largest answer intervals occurs around 4:00 GMT. But in the hours after 11:00 GMT, the answer

intervals rapidly decrease. The length of the answer interval for each hour trends closely with the number of questions posted during that hour. The Pearson Correlation is -0.832 for the median First Answer Intervals and -0.814 for the median Accepted Answer Intervals. Both of these correlations are significant at the 0.01 level and indicate that there is a strong relationship between the time intervals during which most people are asking questions and the time intervals during which contributors are answering questions most quickly. This relationship also might explain why the answer intervals are higher on weekends despite the higher Accepted Ratio, because there are fewer active users during the weekends.

Third, between 4:00 and 8:00 GMT is the worst time to submit a question. On each day other than Sunday, the Accepted Ratio drop below 59% during these times. Furthermore, on the weekdays the Unanswered Ratio increases above 22% during these times. We call this time interval the *low efficiency hours*.

It is interesting to note that while the *SO off-peak hours* we defined earlier do not entirely overlap with the *low efficiency hours*. The *low efficiency hours* correspond to 23:00 to 3:00 East Coast US time<sup>3</sup>, 20:00 to 00:00 West Coast US time, and 5:00 to 9:00 Central European Time. The country distribution of the expert contributors, as defined in §III-B, may explain the reason behind the *low efficiency hours*. Most experts reside in the US (40%), Central Europe (15%), the UK (14%), or Canada (5%). Experts residing in those countries are likely unavailable during this time period, and the quality of answers posted during these hours is generally lower.

Another observations is that a portion of the *SO off-peak hours* (i.e., 23:00 to 2:00) have the lowest Unanswered Ratios and the highest Acceptance Ratios. This result may be due to few questions being posted while US-based experts (40% of all experts) are available. Note that these times correspond to 15:00 and 18:00 West Coast US time.

Finally, 4:00 GMT to 5:00 GMT is the worst time interval to post questions to SO. This time interval falls in both the *off peak hours* and the *low efficiency hours*.

### D. Proposed Strategies for Increasing Reputation Score

Contributors gain additional privileges along with reputation. *Trusted users*<sup>4</sup>, those with a reputation score of at least

<sup>3</sup> Ignoring daylight saving time

<sup>4</sup> <http://stackoverflow.com/privileges/trusted-user>

TABLE III  
10 FASTEST CONTRIBUTORS TO EARN TRUSTED USER LEVEL

UID	First Score	Days	Ans.	Ans./day	Median	1st Ans.	Accepted	Top	Post Hours	Top Tags
938089	2011-10-09	64	489	7.64	5.08	70.8%	63.4%	70.1%	09:00-23:00	javascript, jquery, html, css
616700	2011-02-14	73	1004	13.75	4.38	70.7%	46.3%	61.1%	08:00-01:00	c, c++, java, linux
22656	2008-09-26	77	1184	15.38	7.32	47.4%	43.1%	53.4%	06:00-00:00	c#, .net, java, linq, asp.net
573261	2011-01-12	77	1085	14.09	14.35	45.8%	50.1%	50.4%	18:00-12:00	sql, mysql, sql-server, query
224671	2010-01-07	77	895	11.62	8.33	64.6%	42.9%	54.0%	06:00-20:00	iphone, c++, objective-c, javascript
335858	2011-11-20	84	926	11.02	6.65	44.8%	41.1%	45.1%	11:00-04:00	c#, java, c, objective-c, c++
157882	2009-11-01	85	1245	14.65	22.15	37.8%	34.5%	43.4%	11:00-04:00	java, jsp, html, servlet, jsf
95810	2009-04-25	85	1143	13.45	12.23	46.0%	34.2%	45.4%	14:00-06:00	python, sql, c++, mysql
922184	2011-08-31	91	563	6.19	2.88	72.3%	46.0%	62.0%	16:00-08:00	c, c++, java, algorithm
61974	2009-11-02	95	856	9.01	6.52	43.9%	39.3%	48.1%	08:00-03:00	c#, sql, regex, python, mysql

20,000, have the most privileges. To propose strategies for quickly gaining reputation, we evaluated the working style of those who became *trusted users* most quickly. The data identified 1,024 *trusted users* but did not provide information about when the user became trusted. We wrote a script to calculate the daily reputation score for each user, based on the SO reputation rules<sup>5</sup>. In our calculation, we could not identify the number of down votes cast (-1 score) by a user to answers. Therefore, reputation score calculated by our script is slightly different than reputation score in the user's profile. This difference does not affect our results because the difference between our reputation score calculation and the score provided by SO is less than 1%. Table III presents the results for each of the 10 users who became trusted most quickly relative to: number of posts, comments, acceptance rate, answer interval, answer areas, distribution of posting times, and up votes received for answers. We analyzed the working style of these 10 users to identify patterns they may have led to their success.

First, these users were highly active. Their activity spanned 14 hours per day and averaged more than 10 answers per day.

Second, these users answered quickly with median answer intervals (sixth column of Table III) much lower than the overall population median of 24.45 minutes.

Third, four of the users had high percentages (over 60%) of posting the first answer to a question. The four users also had the highest answer acceptance rates. This observation is not surprising, because there is a high probability (0.44) of users accepting the first answer (termed the 'Fastest Gun In the West' problem among SO users<sup>6</sup>).

Fourth, the accepted answer is not always the most voted answer. All of these users had more most-voted answers than accepted answers. SO community member liked their answers resulting in a higher score based on *up votes*. SO only allows a user to score 200 points per day from *up votes*, and these users earned the daily maximum at least 40 times while earning *trusted user* status. Indeed, absent a daily maximum, the users would have reached *trusted user* status more quickly.

Fifth, among all 10 users, only the 1st, 4th, and 9th focused on questions related to a small number of tags. The other seven

users focused primarily on a set of core tags but also posted many answers/comments unrelated to their core tags.

Finally, in §III-C we identified SO efficiency as being low between 4:00 to 8:00 due to the unavailability of many expert contributors. We found that the hours of activity for 2 of the 10 users overlapped completely with these low efficiency hours, while the hours for 3 of the other 8 users overlapped partially. However, the 10 users did not take advantage of expertise shortages (see §III-B). Only 1 of the 10 contributors had one low-expertise area among his top 10 focus areas.

#### IV. DISCUSSION AND CONCLUSION

We analyzed the SO dataset from four perspectives to provide suggestions to potential reputation seekers. We found that a large number of questions are related to .NET technologies, OOP languages, and web development. Therefore, contributors with expertise in those topics will have a greater chance building reputation quickly. However, these topics also have lower answer intervals and higher experts-to-question ratios, indicating that contributors need to be prompt because of more competition. Our analysis showed that being prompt and being the first respondent helps quickly build reputation. Another option is for a contributor to focus on topics with small numbers of experts (e.g., facebook or xcode) or on mobile development (e.g., android or ios). Besides offering less competition, topics with few experts often have higher median answer intervals. Therefore, a contributor has more time for answer preparation. In addition, a contributor can be active when most experts are not (i.e., between 4:00 to 8:00 GMT). Finally, a contributor should participate regularly and answer as many questions as possible. These actions will improve the contributor's influence and chances of getting up-votes. We believe that these strategies will not only help a contributor quickly build reputation, but also improve the efficiency of the entire SO community.

#### REFERENCES

- [1] A. Bacchelli, "Mining challenge 2013: Stack overflow," in *The 10th Working Conf. on Mining Soft. Repositories*, 2013, p. to appear.
- [2] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," 2009.
- [3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.
- [4] R. Lambiotte, J. c. Delvenne, and M. Barahona, "Laplacian dynamics and multiscale modular structure in networks," *ArXiv*, 2009.

<sup>5</sup> <http://stackoverflow.com/faq#reputation>

<sup>6</sup> <http://meta.stackoverflow.com/questions/9731/fastest-gun-in-the-west-problem>