Mining Testing Questions on Stack Overflow

Pavneet Singh Kochhar
School of Information Systems
Singapore Management University
kochharps.2012@smu.edu.sg

ABSTRACT
During software maintenance, testing is a crucial activity to ensure the quality of code as it evolves over time. With the increasing size and complexity of software, adequate software testing has become increasingly important. Developers often ask problems they face during testing on Community Question Answering (CQA) websites such as Stack Overflow. These websites can serve as good repositories to understand the common topics of discussions and challenges faced by developers during testing.

In this paper, we present a study of common challenges and important topics of discussion, by mining testing related questions asked on Stack Overflow. We use unsupervised learning to categorize the questions and rank all the Stack Overflow questions based on their importance. Our results show that topics such as test framework, database and client server are more often discussed compared to other topics. Also, there has been an uptrend for mobile development questions in testing related discussions.

CCS Concepts
• Software and its engineering → Software testing and debugging; Empirical software validation;

Keywords
Stack Overflow, Topic Modeling, Software Testing

1. INTRODUCTION
Software testing is an important part of software development life-cycle. Despite the availability of various tools to ensure quality of software through testing, most software products suffer from insufficient testing. The impact of inadequate testing can consist of a substantial number of unhandled failures, which leads to poor quality of software, higher software development costs and delays in time to market the product. A study conducted by the National Institute of Standards and Technology reported that inadequate software testing costs the U.S economy $59.5 billions annually, i.e., about 0.6% of its GDP [14]. The number of bugs uncovered after the code has been shipped can overwhelm projects developers when software is not thoroughly tested. Thus, a triager from Mozilla project admitted that they receive almost 300 bugs everyday that need triaging [2]. These figures reinforces the fact that software testing is paramount for developing bug free software.

Software developers1 often post problems they face during testing on question answering websites such as Stack Overflow2, which can serve as good repository for other developers to find similar questions asked by their counterparts. However, what are the different types of questions asked by software developers? What are the hot topics of discussion among software developers? What are the common challenges faced by developers? These are some of the important questions whose answers can help software developers to overcome the challenges they face everyday while testing their applications.

To address this problem, in this study we mine questions from Stack Overflow. Our goal is to bridge the gap between the developer and research communities, and help in developing tools which can help developers in efficiently testing their applications. We conduct quantitative as well as qualitative study of questions related to software testing. Stack Overflow, being one of the active QA sites, offers huge amount of data which can be leveraged to get insights into the issues faced by testers. Our work is inspired by the study of Bajaj et al. [3], that analyzes questions asked by web developers on Stack Overflow.

The contributions of this paper are as follows:

1. We categorize testing related discussions into multiple categories based on the topics of the interactions among developers. We highlight the common topics of discussion.

2. We identify temporal trends in the discussions to understand the current and future trends in software testing.

3. We discuss important challenges faced by developers and solutions to those problems as posted by other developers.

1In this study, we refer software developers to people which do development and/or testing.
2http://stackoverflow.com/
2. BACKGROUND AND MOTIVATION

In this section, we give a brief description about software testing, Stack Overflow and Latent Dirichlet Allocation (LDA).

2.1 Software Testing

Software development produces programs that are often buggy or incomplete with respect to some features. Software complexity, and therefore that of bugs, grows to the limits of our ability to manage that complexity [5]. To verify that software is compliant with its requirements, developers often resort to software testing.

Testing is an activity which is managed through the building of a set of test cases. A test case consists of a set of inputs, preconditions, as well as the associated expected results to be compared with the outputs of the program under testing. Together with test scripts, test cases are constructed to check whether a given software performs in conformance with its requirements. Test cases can further provide the following information such as bug count, help managers in decision making, examine quality of product, how to get these bugs fixed etc. Tests are useful when they produce significant results, provide insight into a product or application, easier to evaluate and reveal valuable information.

2.2 Stack Overflow

StackOverflow is one of the biggest question answering websites where users can share knowledge, seek expert advice on a wide range of topics in computer programming. Users on StackOverflow have the ability to ask and answer questions, to vote questions up and down and several other features. StackOverflow employs gamification techniques to reward users for performing various set of actions.

With more than 2.7 million users and over 7,000,000 questions, StackOverflow has become a huge knowledge repository. Most of the questions are generally related to a specific programming problem, a software algorithm or software tools. Each question is assigned tags according to the topic which question belongs to. For example tag “unit-testing” specifies that the question is related to unit testing. Stack Overflow has a set of predefined tags and only users with reputation above certain level can create new tags.

2.3 Latent Dirichlet Allocation

LDA model [6] has been widely used in information retrieval. LDA has the ability to model topics in large corpus i.e., questions/answers in our experiment. LDA model is represented as graphical model.

The topic mixture in LDA is drawn from a conjugate Dirichlet prior, which remains same for all the questions/answers. The process of generating question profile \( \theta_u \) for a specific question \( u \) is shown in Figure 1 and described as follows:

1) \( \phi_z \) is the multinomial distribution for each topic \( z \) from a Dirichlet distribution with parameter \( \beta \), \( \phi_z \) gives the word distribution within topic. 2) Pick a multinomial distribution \( \theta_u \) for each question from Dirichlet distribution with parameter \( \alpha \). 3) Select a topic \( z \in \{1, \ldots, K\} \) from the multinomial distribution \( \theta_u \) for each word token \( w \) in question profile \( \theta_w \). 4) pick word \( w \) from the multinomial distribution \( \phi_z \). To generate the question profile, above procedure is repeated for \( N_u \) times where \( N_u \) is number of words in \( \theta_w \). Further, the above procedure is repeated \( N \) times for \( N \) questions. The likelihood for the question profile collection is given as

\[
P(u_1, \ldots, u_N|\alpha, \beta) = \prod_{i=1}^{N} P(\phi_z|\beta) \prod_{u=1}^{N} P(\theta_u|\alpha)
\]

(1)

![Figure 1: Question profile corpus generated by Latent Dirichlet Allocation](image)

LDA presents a different representation compared to the language model for generating question profile based on topics of the question. LDA gives multiple topics for each question which shows that each question may define a problem belonging to different but related topics. After we estimate \( \theta \) and \( \phi \), the probability of generating a word from a question profile is given as

\[
P_{LDA}(w|\theta, \phi, \theta_w) = \sum_{z=1}^{K} P(w|z, \phi)P(z|\theta, \theta_w)
\]

(2)

where \( \theta \) and \( \phi \) are the posterior estimates of \( \theta \) and \( \phi \) respectively.

3. METHODOLOGY

In this section, we describe our dataset, the research questions we investigate and methodology to process our data.

3.1 Dataset

We collect questions from Stack Overflow from Jan 2009 to Dec 2014, which contain word “test” in their tags such as unit-testing, automated-tests etc. The questions in our dataset may span across different programming languages and different platforms. Table 1 shows the dataset we analyse for this study, which consists of over 38,000 questions asked by 25,292 distinct users.

<table>
<thead>
<tr>
<th># of Questions</th>
<th># of Askers</th>
</tr>
</thead>
<tbody>
<tr>
<td>38,289</td>
<td>25,292</td>
</tr>
</tbody>
</table>

3.2 Research Questions

In this study, we investigate the following research questions:

RQ1: What are the categories of topics of testing related discussion?

RQ2: What are the hot topics related to software testing in terms of importance?
**RQ3**: Are there temporal trends present in discussions related to software testing?

**RQ4**: How prevalent are testing-related topics in discussions related to mobile web development?

**RQ5**: What are the main technical challenges related to testing?

### 3.3 Data Processing

In this section, we describe how we process data specific to each research question:

- **Figure 2** shows our overall methodology for analyzing the research questions. We follow the similar methodology to Bajaj et al. [3]. The rest of this section describes how we extract data to answer our questions.

- **RQ1**: **Categorization of topics of discussion.** To answer RQ1, i.e., listing the categories of discussions, we used LDA to categorize the discussions on Stack Overflow. Categories discovered are major topics related to testing. We first extract the text of questions and accepted answers. After this step, we remove stop words, filter out HTML tags and use the Porter Stemming Algorithm\(^3\) to convert all the words to their root form. We run LDA on this filtered data set to find topics. We use the list of generated topics to find the categories of discussion and topics share to find the proportion of discussions belonging to each category.

- **RQ2**: **Finding hot topics of discussion.** For this question, we extract top 2000 most viewed questions. Many developers use Stack Overflow to read the questions which are resolved. Such users do not actively participate. Thus, questions with high view count will be of more interest to developers.

- **RQ3**: **Analyzing temporal trends over time.** To answer this question, we divide our dataset into subsets of

---

\(^3\)https://tartarus.org/martin/PorterStemmer/
6 months data each. The choice of 6 months was to have enough questions to be analysed by LDA. Our dataset ranges from Jan’09-Jun’09 to July’14-Dec’14. Although our dataset contains questions from July’08 to Dec’08, we exclude these questions as Stack Overflow was launched at that time and there were not many questions posted.

**RQ4: Prevalence of testing questions in mobile development.** For RQ4, we first analyze the trend of testing related questions within the subset of questions related to mobile development. To filter out mobile related questions, we use tags such as android, bada, blackberry, iphone, ios, java-me, phonegap, symbian, tizen, webo, and windows-phone, which have been used in the previous study [3]. After identifying questions related to mobile developments, we use LDA on these questions to find the main topics of discussion.

**RQ5: Technical challenges faced by developers.** To answer RQ5, we first select important questions and qualitatively analyze them in depth. We use the metric to identify top 50 questions as proposed by Bajaj et al. [3], which the authors call as Accumulated Post Score (AMS). The calculation of score is as follows:

\[
AMS_i = 3U_i - 25D_i + 10C_i + A_i + F_i
\]  
(3)

where \(U_i\) is the number of users who upvoted the question, \(D_i\) is the number of users who downvoted the question, \(C_i\) is the number of comments for a particular question, \(A_i\) is the number of answers provided to a question and \(F_i\) is the number of users who marked this question as favorite. After calculating the score, we filter top 50 questions to be analysed manually.

### 4. RESULTS

Here, we describe the results of research questions we investigate in this study.

**4.1 Discussion Categories**

To answer this question, we use LDA on our collection of questions related to testing. In total, we have 38,289 questions which have an accepted answer. Table 2 shows different topics and their corresponding words as returned by LDA. We manually give topic names after reading the words in each category. Test framework, database, client server, threads and login are some of the topics which are commonly discussed by developers on Stack Overflow.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Framework</td>
<td>test unit run suit integr</td>
</tr>
<tr>
<td>Database</td>
<td>database db creat db delet</td>
</tr>
<tr>
<td>Client Server</td>
<td>request server respons client http</td>
</tr>
<tr>
<td>Login</td>
<td>user password login usernam</td>
</tr>
<tr>
<td>XML Build</td>
<td>project run build xml includ</td>
</tr>
<tr>
<td>Threads</td>
<td>run start thread process call</td>
</tr>
<tr>
<td>Forms</td>
<td>button window form click element</td>
</tr>
<tr>
<td>Image Processing</td>
<td>imag png imgur path</td>
</tr>
</tbody>
</table>

Developers commonly discuss issues related to test framework, database, threads, forms etc.

### 4.2 Hot Topics

To answer this question, we identify topics that have been viewed by most number of developers. We refer these topics as hot since many developers view these questions to resolve their problems. Since many users do not log in to Stack Overflow to view the question, view count can give us an idea of how often the question is viewed.

Table 3 shows the hot topics, namely test framework, database and client server, that are discussed by users on Stack Overflow.

<table>
<thead>
<tr>
<th>Hot Topics with the highest view counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Framework, Database, Client Server</td>
</tr>
</tbody>
</table>

Test framework, database and client server are hot topics of discussion among developers.

### 4.3 Temporal Trends

To answer RQ3, we divide the six year time period of data into six month intervals, and use LDA to find important topics of discussion. Figure 3 shows the temporal trend of testing related discussions. We depict the trend of hot topics of discussion i.e., test framework, database and client server. We can observe that all the three hot topics follow a similar trend over the years. Lately, there have been more discussions about test framework and client server issues.

**Figure 3: Temporal Trends**

Hot topics have been consistently discussed among the developers from January 2009 to December 2014.

### 4.4 Mobile Development

To understand the prevalence of testing related discussions in mobile development, we first study what percentage of the testing discussions are related to mobile. In total, we have 2,434 mobile related questions. Figure 4 shows the trend of testing discussions in mobile related questions. We can observe that the testing discussions have increased over time in questions related to mobile development.

**Figure 4: Trend of testing discussions in mobile development**

Hot topics have been consistently discussed among the developers from January 2009 to December 2014.
4.5 Technical Challenges

To gain insights into the challenges faced by developers during testing, we manually analyse top 50 questions posted on Stack Overflow. We extract dominant categories by reading the question and we discuss some of the common challenges.

Basics of Testing. Developers often ask questions related to basics of testing. This question is commonly asked by novice developers or developers learning a new language. For example, a user posted:

“What is Unit test, Integration Test, Smoke test, Regression Test and what are the differences between them? And Which tools can I use for each of them?”

App Testing. As we observed in Section 4.4, the questions related to mobile development have increased over the years. During card sort, we found several questions related to testing mobile applications. A user posted:

“How to develop and test an app that sends emails (without filling someone’s mailbox with test data)?”

The accepted answer for this question suggests creating a dummy SMTP server as “Windows 7/Vista/XP/2003/2010 compatible dummy SMTP server. Sits in the system tray and does not deliver the received messages. The received messages can be quickly viewed, saved and the source/structure inspected. Useful for testing/debugging software that generates email.”

It is often infeasible for developers to test their applications on all the different types of mobiles. Thus, they often make use of emulators, which provides a virtual environment to expand testing coverage to many devices. We find that emulator related questions are also commonly asked by developers:

“How to emulate GPS location in the Android Emulator? I want to get longitude and latitude in Android emulator for testing.”

The accepted answer suggests connecting to the emulator via Telnet, which lets user enter data like geo fixes, network etc. The answerer also gives example on how to connect to console and command to be run:

telnet localhost 5554
geo fix <longitude value> <latitude value>

Best Practices. Stack Overflow platform provides an opportunity for novice developers to ask questions to their experienced counterparts, who might have faced similar challenges before. Experienced developers often share best practices which can make the job of other developers easier. A developer asked the following question:

“I was wondering what the best practice is for unit testing abstract classes and classes that extend abstract classes.”

The answerer suggests writing “a Mock object and use them just for testing” and using unit test to call the abstract method that the developer wants to test.

Test Framework. Frameworks provide an execution environment to help developers run test cases automatically. The framework integrates the function libraries, test data sources and various reusable modules to simplify the automation effort. Developers ask questions on which framework to use and usage guidelines. One of the question asked by a developer:

“ NUnit vs. MbUnit vs. MSTest vs. xUnit.net... There are quite a lot of unit testing frameworks out there for .NET. ... Now I am to choose the best one for us. But how? Does it matter? Which one is most future proof and has a decent momentum behind it? Should I care about the features?”

Database. As we find in RQ2, database is one of the hot topics as discussed by developers on Stack Overflow. During manual analysis, we find that developers ask questions on how to test database related applications. A question posted by a user:

“testing the ORM and database itself has always been fraught with problems and compromises... What strategies have you used for testing database-driven applications, if any? What has worked the best for you?”

The accepted answer suggests several ways to overcome this problem. First, to keep the schema and the scripts that generate it in the same source control that lets other users create a schema after checking out. Another solution is to use Continuous Integration (CI) to build the schema and run tests, which helps find bugs that will usually be found later in the lifecycle.

Web Testing. Our previous questions (RQ1, RQ2 and RQ3) show that developers often ask questions related to web applications. We found similar such questions during manual analysis which deal with testing of web applications. Below is one of the examples:

http://stackoverflow.com/questions/243274
http://stackoverflow.com/questions/261139
http://stackoverflow.com/questions/261139
http://stackoverflow.com/questions/340564
The user accepted answer suggests using tools such as Selenium\(^\text{11}\), which is a testing framework for web applications. Selenium allows writing tests for many popular languages such as Java, C\#, Python and can be run against most modern browsers.

| Users often ask questions related to app testing, best practices, test framework and database testing. |

5. Threats to Validity

In this section, we describe several threats to validity for our empirical study.

Threats to external validity relate to the generalizability of our results. We have investigated testing related questions from Stack Overflow. It is difficult to generalize our results for other CQA websites. To the best of our knowledge, Stack Overflow is one of the biggest repositories for programming related questions.

Threats to internal validity relate to the conditions under which experiments are performed. We only mine questions which contain word “test” in their tag. We might have missed out on some questions which are related to testing but their tag does not contain word test.

A threat to construct validity is that we use metric define by Bajaj et al. \(^\text{3}\) to rank questions based on importance for qualitatively analyzing these questions (RQ5). However, this metric uses all the available information related to questions from Stack Overflow such as upvotes, downvotes, answer count etc.

6. Related Work

In this section, we describe studies that analyze Stack Overflow and studies related to software testing.

Researchers in the past have studies Stack Overflow for various reasons. The closest to our work is study done by Bajaj et al. \(^\text{3}\), where they analyse questions asked by web developers, in particular questions related to JavaScript, HTML and CSS. They use topic modeling to categorize questions into different topics. Their results show that web related discussions have increased among developers and some discusisons such as browser related and form validation are more prevalent. Barua et al. analyse questions on Stack Overflow and use LDA to find topics of discussions among developers \(^\text{4}\). They define various metrics for these topics and analyse their trends over time. Their results show that some topics have garnered more interest such as mobile development, use of Git, web development etc. Treude et al. analyse questions from Stack Overflow to categorize what type of questions are asked and which questions are answered \(^\text{16}\). Their results show that Stack Overflow is effective at code reviews and for discussing conceptual questions. Mamyrkina et al. analyse the success of Stack Overflow and found several reasons behind this success such as highly responsive and iterative approach, tight interaction with the community and incentives which motivate users to contribute \(^\text{10}\). Allamanis et al. use topic modeling that combines question concepts, types and code to understand programming concepts with particular types of questions \(^\text{1}\). They find that the types of questions do not vary across programming languages. Pal et al. study the evolution of experts in Community Question Answering (CQA) sites and differentiate expert users from ordinary users \(^\text{12}\). They find that machine learning models using temporal data can be used to identify experts. Tian et al. analyze questions and answers on Stack Overflow and use topic modeling to predict the best answerer for a new question posted on the website \(^\text{15}\). They find user’s interests by analysing questions answered by the user and use collaborating voting mechanism to find users expertise. The results show that their approach performs better than the TF-IDF based approach.

There have been several studies on testing in the past. Pham et al. discuss several strategies to understand the testing culture on social coding sites such as GitHub. They also present some guidelines which can be used by developers and managers to influence the testing behavior in their projects \(^\text{13}\). Greiler et al. conduct a qualitative study of test practices followed by a community of people working on plug-in based applications \(^\text{7}\). Memon et al. present their analysis to improve the current testing techniques and strategies to create new collaborative development and testing processes where developers can share tools and information repositories \(^\text{11}\). Kochhar et al. investigated the correlation between the presence of test cases and various project development characteristics, including the lines of code and the size of development teams \(^\text{8}\). They extended their study to include characteristics such as number of bugs, number of bug reporters and the programming languages \(^\text{9}\).

7. Conclusion and Future Work

Software testing is used to ensure that the software produced is complete, correct, secure and of higher quality. Developers often post questions/queries on Community Question Answering (CQA) websites like Stack Overflow, which provide a good repository to analyse common topics of discussions and challenges faced by them. In this study, we analyse testing related questions and use topic modeling to find topics of discussions, temporal trends and challenges faced by developers.

Our analysis shows the following results:

1. Test framework, database, client server, threads, forms are some of the discussion categories on Stack Overflow.
2. Among common discussions test framework, database, client server are hot topics.
3. Questions related to test framework and database have been regularly discussed over time.
4. Mobile related discussions have increased in testing questions.
5. Users often post questions related to basics of testing, app testing, test framework, best practices and testing database-driven applications.

In this work, we only consider questions from Stack Overflow. We plan to increase the number of questions and

\(^\text{11}\)http://www.seleniumhq.org/
answers by analysing other community questions answering websites. Also, we plan to survey developers to analyse if the common challenges faced by them are the same as found in our study.

8. REFERENCES


