Mining Developer Questions about Major Web Frameworks

Zakaria Mehrab, Raquib Bin Yousuf, Ibrahim Asadullah Tahmid and Rifat Shahriyar
Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh

Keywords: Mining, Web Framework, Stack Overflow.

Abstract: Web frameworks are the de facto way to build web-enabled applications. Stack Overflow, being one of the leading question answering sites available, has become a helpful resource in numerous software engineering research. In this paper, we present a study of common challenges and issues among developers of two major web frameworks namely Laravel and Django by mining questions asked on Stack Overflow. We extracted the issues that the developers are most concerned about. We sorted these issues by popularity and difficulty metrics and observed the contrasting nature of difficulty and popularity. We also noted an exception that installation is a popular issue over both the frameworks and simultaneously it is also difficult to resolve. Besides, we found that about 50% issues are common over both the frameworks. Our findings would help the framework developers to understand better the need of the framework users by focusing most difficult and the most popular issues.

1 INTRODUCTION

Web development is a term comprised of developing websites, web services, and web application. In recent years, web applications require aesthetic representation as well as an efficient interaction between users and websites, leading to more complexity. According to Hevner et al. (Hevner et al., 2007), currently web engineers face three intractable problems: domain/system complexity, increased development time and cost. Web developers feel the need for modularization of concerns to accommodate big and complex applications (Ginige, 1998). Web frameworks help developers in this task by bringing a standard way to develop and implement web applications. These frameworks often help to decouple the logic portion from the view portion, namely implementing the Model-View-Controller(MVC) pattern. Moreover, frameworks enable developers to reuse design and implementation by combining the procedures of necessary tasks. According to J. Carlos et al. (Fernández-Conde and González-Calero, 2002) these frameworks have a positive effect on the project with shortened development time, reduced complexity, increased productivity, extensibility, and reliability.

Since the first appearance in the late 1990s, more than 5,000 frameworks have been released (Github, 2018d). While using these frameworks for creating web applications, developers often find themselves grounded with various problems. Developers often post questions in Stack Overflow (Exchange, 2017a), part of the Stack Exchange platform, to seek help and guidance. According to Mamykina et al. (Mamykina et al., 2011) Stack Overflow is larger than any other social Q&A forum or programming forum. Therefore, a thorough analysis of its posts can help us understand the problems faced by the web framework developers.

Created in 2008 by Jeff Atwood and Joel Spolsky, Stack Overflow is the flagship site of the Stack Exchange Network. It features questions and answers on a wide range of topics regarding computer programming and related technologies. The huge amount of web framework related posts in Stack Overflow shows the importance of studies related to the problems faced by web framework developers. In Figure 1, we have presented a graph with the percentage of posts related to eleven popular frameworks. Together these eleven frameworks measure up to more than 15% of the total questions on Stack Overflow each month in 2017.

We used Stack Overflow data dump (Exchange, 2017a) to explore the issues of web frameworks that developers face.

We picked two web frameworks, Laravel, and Django for our analysis. The reason behind choosing Laravel and Django is that both are very popular

Mehrab, Z., Yousuf, R., Tahmid, I. and Shahriyar, R.
Mining Developer Questions about Major Web Frameworks.
DOI: 10.5220/0006929501910198
In *Proceedings of the 14th International Conference on Web Information Systems and Technologies (WEBIST 2018)*, pages 191-198
ISBN: 978-989-758-324-7

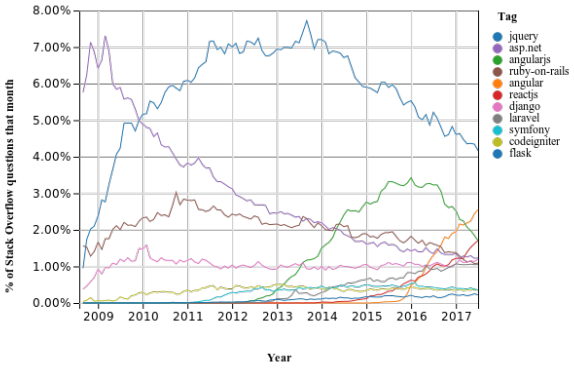

Figure 1: Stack Overflow trends of framework related post.

but not well studied, unlike Javascript frameworks. Whereas Laravel and Django together have a little over 45,000 stars on their GitHub repository (Github, 2018b) (Github, 2018a), one Javascript framework, vuejs, alone has bagged over 95,000 stars on its GitHub repository (Github, 2018c). There is also the fact that searching with keyword ”javascript” in (DBLP, 1993) shows 784 matches whereas ”django” keyword produces 34 results (24 of which belongs to author ”Django Armstrong”) and ”laravel” keyword produces two results.

For studying the two frameworks, we extracted contents from all the related questions. We used them to train our model in MALLET topic modeling tool (MALLET, 2017). We extracted the appropriate, relevant topics of each question by Latent Dirichlet Allocation(LDA). Once we were able to extract the individual topic of each question, we started to answer our research questions.

RQ1 What are the issues users of these two framework face?

RQ2 What are the most popular issues asked by developers of these two frameworks?

RQ3 Which issues of these two frameworks are the most difficult?

RQ4 Are the issues of these two frameworks different?

From our experiment, we found 18 issues for Laravel and 20 issues of Django; faced by the developers. We sorted these issues by different metrics and ranked them based on their popularity and difficulty. We observed the contrasting nature of difficulty and popularity. We also noted an exception that installation is a popular issue over both the frameworks and simultaneously, it is also difficult to resolve. Also, we found that about 50% issues are common over both the frameworks. Our findings can help the developers of the frameworks to better understand the need of the users. They can categorize the needs of users to focus on the most difficult and the most popular issues. They can also find similar issues in multiple frameworks those require major improvement.

The rest of this paper is organized as follows. The next section reviews the related works. The methods for data collection, operationalization and analysis are presented next. This is followed by our findings and related discussions. The paper concludes by highlighting possible future extensions of our work.

2 BACKGROUND AND RELATED WORKS

Laravel is an MVC framework for PHP web language, released in 2011. Olanrewaju et al. (Olanrewaju et al., 2015) discussed the most famous MVC based PHP frameworks, evaluated their performance and found that Laravel outperforms other MVC frameworks, dubbed Laravel as the most suitable PHP framework for future web technology. **Django** is an open source web framework built on Python programming language. It was released in 2005 and quickly gained popularity due to re-usability of components; rapid development leads to the development of complex web applications in shorter time.

2.1 Related Work

We looked into works which discuss the issues that the developers face while using web frameworks. We also went through several types of research for better understanding the issues and trends among developers from the empirical analysis of Stack Overflow data.

The empirical studies using Stack Overflow data is now one of the growing trends of empirical software engineering. Several papers using Stack Overflow data has been presented. Some of them use the data to present behavioral research question about the Stack Overflow users. Others use topic modeling to categorize the discussions in the posts.

Allamanis et al. (Allamanis and Sutton, 2013) applied topic modeling on Stack Overflow questions. After associating them with programming concepts, they found that certain types of questions are associated with specific programming concepts. Treude et al. (Treude et al., 2011) analyzed data from Stack Overflow to categorize the kinds of questions that are asked and found that Q&A websites are particularly effective in code reviews and conceptual questions. Mamykina et al. (Mamykina et al., 2011) took the design lessons from Stack Overflow and found that the success of Stack Overflow is because of the quickness of a question getting answered. They also found that

the high visibility and daily involvement of the design team of Stack Overflow help increase Stack Overflow's popularity. Barua et al. (Barua et al., 2014) aimed to analyze the actual textual content of Stack Overflow to help the software engineering community to better understand the thoughts and needs of developers. They found that the topics of interest to developers range widely from jobs to version control systems to C# syntax. They dictated that the topics gaining the most popularity over time are web development, mobile applications, Git and MySQL. Li et al. (Li et al., 2013) performed an empirical study with 24 developers to understand the needs and challenges developers face during the development phase. Beyer et al. (Beyer and Pinzger, 2014) manually analyzed 450 Android related posts to determine common problems developers face. Rosen et al. (Rosen and Shihab, 2016) analyzed Stack Overflow data to determine what mobile developers ask about by using LDA-based topic models. They also determined what popular mobile-related issues are the most difficult, explore platform-specific issues, and investigated the types of questions mobile developers ask. Bajaj et al. (Bajaj et al., 2014) presented a study of common challenges and misconceptions among web developers by mining-related questions about client-side code, written in JavaScript, HTML, and CSS asked on Stack Overflow. They used unsupervised learning to categorize the mined questions and defined a ranking algorithm to rank all the Stack Overflow questions based on their importance.

The related works for this topic revolve around the empirical use of Stack Overflow data to determine the issues developers face, trends of asked topics and behavior by the users. These tasks require textual content analyzing which is done mostly with topic modeling studies. Though several empirical studies have been done with the Stack Overflow data, none have tried to determine the specific issues faced by web framework users. We aim to fill that void by looking into the framework related posts and bringing out the issues developers face.

3 METHODOLOGY

3.1 Data Extraction

The data dump of Stack Overflow contains all the data of the website in XML form. We obtained the data dump of August 2016. Among the XML files residing in the dump, we used only the files relevant with our purpose, namely *posts.xml*, *tags.xml* and *users.xml*. We ran a script to extract these XML files to corresponding MySQL database tables. The database schema of stack exchange files is available in this link (Meta, 2017).

3.2 Data Processing

Having built our database, we focused on retrieving the Laravel and Django related questions from the Post table. The Post table contained a whopping 32,209,817 entries. Among these entries, 12,350,818 are questions. From these questions, we used a tag-based search to extract 59,360 Laravel related questions and 130,588 Django related questions.

After that, we targeted on extracting the body of each question and preparing them for textual analysis using topic modeling. The bodies can contain HTML tags and several unwanted strings. We trimmed out the HTML tags and unwanted strings from the bodies. Then, we created separate files for each body and kept them in a directory. These files will represent a corpus of documents to be used in topic modeling.

3.3 Topic Modeling using LDA

Inference of topics from a corpus of documents is performed by Latent Dirichlet Allocation (LDA). According to the LDA model, each document contains a mixture of topics. Topics are also allowed to exist across several documents. So it is easy to discover themes and ideas that represents all the documents as a whole (Rosen and Shihab, 2016).

To perform topic modeling, we used MALLET version 2.0. It is a Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text (MALLET, 2017). The MALLET topic modeling tool uses LDA.

Running MALLET on a corpus requires two hyperparameters; namely: topic numbers and words per topic. After assembling our data, we ran the MALLET tool with different combinations of these hyperparameters. We performed the experiment with 20 topics and 10 words per topic, 20 topic and 20 words per topic, 50 topics and 20 words per topic. After each run, We manually checked the words in each topic and through consensus, we concluded that the result found with 50 topics and 20 words per topic captures the theme of data better than the others. So we retained 50 topics and 20 words per topic as our final hyperparameters. The output is distributed among two files. One, namely "key.txt", contains word sets of each topic along with their Dirichlet parameters. The second file, namely "composition.txt", contains

Table 1: Set of words, their merging and given name of Laravel topics.

Merged given name of set of words	Set of words
authentication	email send password mail address message emails user laravel sending reset code messages users function account hash link username sends
	user users login admin laravel authentication logged auth password check access username create profile role permissions system account log roles
date_time	date event time events format day dates month carbon days year current database datetime timestamp set code timezone start fire
database	table user users model tables pivot relationship models user_id company relation column eloquent list data relationships roles columns group role
	table key migration error database foreign column migrations tables create primary sql created add run constraint laravel seed migrate sqlstate
	model models eloquent relationship relationships method relation related object eager relations table collection property loading set tables attribute load attributes
	table database update data record insert row delete model field records column code rows save laravel deleted method fields values
	query eloquent laravel table sql result column results queries select builder mysql join count order rows raw tables return database
	database data laravel connection save mysql create store stored set mongodb document created application databases default saved project settings contacts

the probability of each topic for each document of the corpus.

Having analyzed the words for each topic, we noticed that some of the word sets are similar in meaning. So we merged word sets conveying similar meaning and named the topics manually. Through this merging process, we obtained 18 topics for Laravel and 20 topics for Django. Thus we obtain the answer to our first research question, the issues users of these two frameworks face.

The naming of the topics was performed through consensus among the authors and the peers. A few samples of the naming and merging process have been shown in 1 and 2. The full list could not be tabulated here due to space constraints. The complete list can be viewed in (Authors, 2018b) and (Authors, 2018a). In both of the frameworks, we named a topic as "General Issue", which contains words which could not be classified to any specific topic. However, the human consensus is error prone. The naming and merging process of the topics may vary depending on the perspectives of developers. We are not claiming it to be anything concrete.

3.4 Finding Popular and Difficult Issues

With our topics identified by human understandable names, we focused on answering our research questions. To answer our first research questions, we calculated the most dominant topics of each post with the help of the composition file generated by MALLET. Next, We calculated total questions and total views for each topics using these posts where that particular topic is dominant. Then we sorted the topics by view per question ratio. Naturally, the topic that has the most view per question is more popular.

To answer our third research questions, we determined the difficulty level of each topic by associating with it the mean and median duration in minutes of getting an accepted answer after a related question is posted. We ranked the topics by median time as according to Rosen et al. (Rosen and Shihab, 2016) the mean is likely to be skewed by long latency responses.

3.5 Finding Common Issues

For answering our final research questions, we manually checked the two sets of topics: The first set being the topics generated from Laravel (L) related posts and the second set being the topics generated from Django related posts (D). Using these sets, we obtained three other sets as follows:

- $L - D$ (Issues only faced by users using Laravel Framework)
- $L \cap D$ (Issues those are common for both users of Laravel and Django)
- $D - L$ (Issues only faced by users using Django Framework)

Table 2: Set of words, their merging and given name of Django topics.

Merged given name of set of words	Set of words
file upload	file image upload images files uploaded django save download path video user photo uploading picture code media thumbnail imagefield uploads
form	form forms data fields field view user validation model forms.py submit save views.py formset post template input modelform django create
	page button ajax html view click code jquery javascript function form user django template submit views.py link load display call
	form field select fields model widget list choices selected user values choice option input dropdown add set forms django box
	date time event day events model datetime month format dates year field days current calendar hours django set number start
migration	database django migration migrations run table south tables error created model migrate data app create models syncdb command manage.py running
authentication	file line return usr/local/lib/python c:python usr/lib/python lib/python error traceback exception init pid call kwargs import recent request response python tid
	email user password login facebook authentication username django users send account app token registration auth emails address oauth social google
	user users model profile create permissions group add created django permission custom access userprofile username logged groups view user's app
	user login page session logged view django redirect site users log cookie set authentication middleware url redirected sessions cookies access

4 FINDINGS

4.1 Evolution of Framework Related Posts

We looked into the trend of Laravel and Django related question in Stack Overflow quantitatively. First, we measured the percentage of Laravel related post to total post in each year. We measured the same for Django related post. The public data dump we used for topic modeling had was updated until August 2016. But for the trend of the Laravel and Django related post, we used the Stack Exchange Data Explorer (Exchange, 2017b) which enabled us to run queries online with the latest Stack Overflow data. The trend is presented in Figure 2.

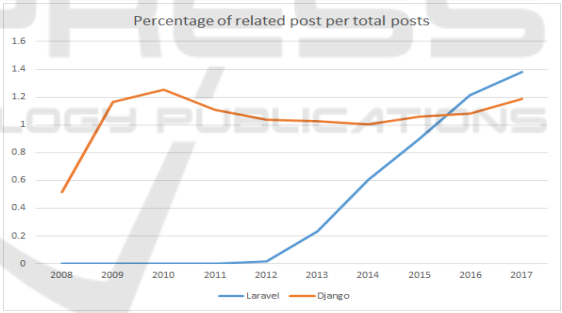

Figure 2: Percentage of related post per total posts per year.

4.2 RQ1

To answer our first research questions, we analyzed the word sets generated by mallets and manually provided understandable human names for the topics. The naming was done with consensus among peers proficient in these two frameworks. The topics for Laravel framework were named as *Authentication, API, Date, Database, Documentation, Form, Image, Installation, General Issue, Language, MVC, Application, Session, Syntax, Transaction, Unit Testing, Elixir, and View*. Similarly, the topics for Django framework were identified by the following names: *Admin, API, App, Authentication, Cross-Site Request Forgery, Documentation, Encoding, File Upload, Form, General Issue, Installation, Localization, Migration, Model, Pep, Query, Server, Template, Unit Testing and View*. Collectively, these are the issues that developers of these web frameworks face.

4.3 RQ2

Having named the topics, we attempted to rank them by popularity. For that purpose, we calculated the total number of questions and total views for each topic.

Table 3: Popularity rank of Laravel and Django Issues.

Rank	Laravel				Django			
	Topic	Question	View	V:Q	Topic	Question	View	V:Q
1	Installation	5459	5814400	1065.1	Encoding	1962	3356472	1710.74
2	Documentation	1329	1302098	979.758	Installation	6610	9888933	1496.06
3	General-Issue	10337	9619519	930.591	Server	3633	4856344	1336.73
4	MVC_basic	10766	9371661	870.487	Admin	5693	7372228	1294.96
5	Database	9638	8236913	854.629	*CSRF**	3117	3830184	1228.8
6	Session	2207	1837996	832.803	Migration	3062	3729092	1217.86
7	Syntax	3842	3195133	831.633	General-Issue	10207	12225460	1197.75
8	date_time	680	531020	780.912	App	12187	14481413	1188.27
9	Form	3189	2451902	768.862	View	6922	8009187	1157.06
10	Image	652	482527	740.072	file-upload	3119	3566093	1143.34
11	Authentication	2838	2009346	708.015	pep	3343	3799548	1136.57
12	API	1118	772301	690.788	Template	11401	12633372	1108.09
13	View	4305	2817349	654.436	Documentation	1791	1971313	1100.68
14	Application	1140	743720	652.386	Query	6488	6638810	1023.24
15	language-translate	181	112114	619.414	Form	15615	15826055	1013.52
16	Elixir	430	240794	559.986	Model	20068	19028206	948.186
17	unit_testing	448	242270	540.781	Authentication	9284	8680880	935.037
18	Transaction	801	342016	426.986	Localization	775	690861	891.434
19					unit_testing	1235	1042258	843.934
20					API	4076	3003249	736.813
*CSRF** = Cross Site Request Forgery								

Table 4: Difficulty rank of Laravel and Django Issues on basis of time to get accepted answer.

Rank	Laravel			Django		
	Topic	Mean	Median	Topic	Mean	Median
1	unit_testing	16034.7	373	API	18763.1	133
2	Elixir	7812.91	171	Installation	18599.5	114
3	Installation	13118.8	105	unit_testing	53229.6	114
4	API	15978.9	94	Localization	13841.3	110
5	Session	11696.4	61	Server	18771.6	101
6	Transaction	4502.43	55	Migration	17148.1	61
7	Application	10116.3	51	*CSRF**	13895.6	58
8	Authentication	8466.89	50	file-upload	11193.4	57
9	View	6251.58	45	Admin	17932.7	50
10	language-translate	5954.43	44	Authentication	12119.6	49
11	Database	5504.18	40	App	15416.5	47
12	date_time	3156.09	39	Documentation	14462.9	46
13	Image	4565.55	39	General-Issue	14890	42
14	General-Issue	7311.93	38	Model	13844.1	41
15	Form	3314.9	37	Form	8715.9	39
16	MVC_basic	6001.73	35	pep	10295.4	34
17	Documentation	4898.51	32	Template	13135.9	32
18	Syntax	6876.79	31	Query	11216.7	31
19				Encoding	7906.35	26
20				View	9458.15	20
*CSRF** = Cross Site Request Forgery						

The ratio of view count by the number of question for each topic has been taken as the ranking metric for popularity. Table 3 presents the popular topics for each framework, sorted from most popular to least popular. From this table, it is evident that installation is a major issue over both the frameworks. Installation is the first step of adopting a framework. Laravel has several installation alternatives which may get confusing for beginners. Django users also face trouble with installation. So it is no wonder that it is one of

Table 5: Common Issues between two frameworks.

Laravel Only	Common Issues	Django Only
Elixir	Api	Localization
Session	Unit_testing	Server
Transaction	View	CSRF
Application	Installation	File Upload
Language	Authentication	Admin
Date_time	Documentaion	App
Image	DB/Model/Query/Migration	Pep
MVC_basic	Form	Template
Syntax	General-Issues	Encoding

the most popular issues for both the frameworks.

Among other issues of Laravel, we can infer from the table that the framework is lacking in quality documentation or tutorial for new users. Moreover, the changes between the version which are continuously rolling for the betterment of the framework also create confusion among the users. The beginners often find Laravel framework as their first MVC pattern framework and need to understand its underlying meaning and implementation. So it is one of the most popular topics with the highest number of question.

Similarly, we can infer that topics related to JSON encoding, deployment and proper installment of a server along with its possible errors and bugs are the most popular topics related to Django framework.

4.4 RQ3

To answer our next research question, we attempted to determine the difficulty level of the topics by associating with it the mean and median duration in minutes of getting an accepted answer and ranking the topics by median time. The topics for each of the framework are presented in Table 4, sorted by their difficulty measurements.

An interesting observation from the difficulty ranking of the frameworks is the contrasting nature of difficulty and popularity. More formally, *Issues with less popularity are difficult,* which is somewhat intuitive. For example, the top two difficult issues of Laravel are Unit Testing and Elixir which are also among the least three popular issues of Laravel. The similar phenomenon can also be observed in case of Django.

Another important finding by observing the two tables is that issues related to installation are popular as well as difficult to answer. This somewhat indicates that, regardless of any framework, users face problems while installation and the problems are also difficult to resolve. This denotes lack of well-documented installation guideline or confusing installation process for both the frameworks.

4.5 RQ4

Having observed all the issues for both of the frameworks, we separated the issues in 3 columns as shown in Table 5. We find that about 50% of the issues are common in both frameworks and the other 50% spans over the unique issues for individual frameworks.

5 CONCLUSION AND FUTURE WORK

Stack Overflow is one of the largest question-answer sites for the programmers and developers. We endeavored to bring insight into the web framework related question in Stack Overflow. We extracted bodies from questions of Laravel and Django and used topic modeling to find the topics. We ranked the topic according to popularity and difficulty and determined the common topics between these two. We found that the most popular topics in Laravel are installation, documentation, general issues, MVC basic, database, session, syntax, date-time, form. The popular topics in Django are encoding, installation, server, admin, Cross-Site-Request-Forgery, migration, general issues, app, and view. We found that the topics in two frameworks are about 50% similar and both frameworks have 50% of the topics as their unique topics. In this way, we made a comparison between these two frameworks that would help the developers to further better their product and the developer community will also be helped through establishing a statistical way to find what web developers are talking about.

In future, we want to conduct an empirical study involving developers from industry as a complementary of this study. Both Laravel and Django are an open-source framework, and their source code is available on Github for feedback, bug report, etc. We want to analyze with Github statistic of the frameworks' repositories to find more about what the community is talking and contributing to these frame-

works. We want to extend our research to other frameworks. Determining these frameworks' issues will enable us to conduct a comparative analysis, and we would be able to establish a list of common issues comprising almost all of the web frameworks. We would also like to conduct a study which will predict the activity level of the frameworks in future, e.g., how many users will keep using a framework or switch to others. The impact of the version change on the users can also be a significant study in this area.

REFERENCES

Allamanis, M. and Sutton, C. A. (2013). Why, when, and what: analyzing stack overflow questions by topic, type, and code. In *Proceedings of the 10th Working Conference on Mining Software Repositories, MSR '13, San Francisco, CA, USA, May 18-19, 2013*, pages 53–56.

Authors (2018a). Django Set of Words. [Online; accessed May 2018].

Authors (2018b). Laravel Set of Words. [Online; accessed May 2018].

Bajaj, K., Pattabiraman, K., and Mesbah, A. (2014). Mining questions asked by web developers. In *11th Working Conference on Mining Software Repositories, MSR 2014, Proceedings, May 31 - June 1, 2014, Hyderabad, India*, pages 112–121.

Barua, A., Thomas, S. W., and Hassan, A. E. (2014). What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Software Engineering*, 19(3):619–654.

Beyer, S. and Pinzger, M. (2014). A manual categorization of android app development issues on stack overflow. In *30th IEEE International Conference on Software Maintenance and Evolution, Victoria, BC, Canada, September 29 - October 3, 2014*, pages 531–535.

DBLP (1993). DBLP Computer Science Bibliography. [Online; accessed May 2018].

Exchange, S. (2017a). Stack Exchange Data Dump. [Online; accessed August 2017].

Exchange, S. (2017b). Stack Exchange Data Explorer. [Online; accessed August 2017].

Fernández-Conde, C. and González-Calero, P. A. (2002). Domain analysis of object-oriented frameworks in framedoc. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering, SEKE 2002, Ischia, Italy, July 15-19, 2002*, pages 27–33.

Ginige, A. (1998). Web engineering: Methodologies for developing large and maintainable web based information systems. In *In: Proceedings of the IEEE International Conference on Networking the India and the World CNIW'98, Ahmeddabad*.

Github (2018a). Github Repository of Django. [Online; accessed May 2018].

Github (2018b). Github Repository of Laravel. [Online; accessed May 2018].

Github (2018c). Github Repository of Vue JS. [Online; accessed May 2018].

Github (2018d). List of web frameworks. [Online; accessed May 2018].

Hevner, A. R., Linger, R. C., and Walton, G. H. (2007). Next-generation software engineering introduction to minitrack. In *40th Hawaii International International Conference on Systems Science (HICSS-40 2007), CD-ROM / Abstracts Proceedings, 3-6 January 2007, Waikoloa, Big Island, HI, USA*, page 276.

Li, H., Xing, Z., Peng, X., and Zhao, W. (2013). What help do developers seek, when and how? In *20th Working Conference on Reverse Engineering, WCRE 2013, Koblenz, Germany, October 14-17, 2013*, pages 142–151.

MALLET (2017). MALLET. [Online; accessed August 2017].

Mamykina, L., Manoim, B., Mittal, M., Hripcsak, G., and Hartmann, B. (2011). Design lessons from the fastest q&a site in the west. In *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*, pages 2857–2866.

Meta, S. E. (2017). Database Schema of Stack Overflow Data. [Online; accessed August 2017].

Olanrewaju, R. F., Islam, T., and Ali, N. (2015). *An Empirical Study of the Evolution of PHP MVC Framework*, pages 399–410. Springer International Publishing, Cham.

Rosen, C. and Shihab, E. (2016). What are mobile developers asking about? A large scale study using stack overflow. *Empirical Software Engineering*, 21(3):1192–1223.

Treude, C., Barzilay, O., and Storey, M. D. (2011). How do programmers ask and answer questions on the web? In *Proceedings of the 33rd International Conference on Software Engineering, ICSE 2011, Waikiki, Honolulu , HI, USA, May 21-28, 2011*, pages 804–807.