

CHALMERS



Demand driven documentation in Open Source projects

Can Demand driven documentation work as a substitute for regular documentation in Open source projects?

JESSICA ANDERSSON

SOFIA LARSSON

CHALMERS UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Gothenburg, Sweden 2014

The Authors grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet. The Authors warrants that they are the authors to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Authors shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Authors have signed a copyright agreement with a third party regarding the Work, the Authors warrant hereby that they have obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Demand driven documentation in Open Source projects

Can Demand driven documentation work as a substitute for regular documentation in Open source projects?

JESSICA ANDERSSON

SOFIA LARSSON

©JESSICA ANDERSSON, April 2014.

©SOFIA LARSSON, April 2014.

Examiner: MIROSLAW STARON

Supervisor: MORGAN ERICSSON

Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden April 2014

Abstract

This case study researches how demand driven documentation can be exploited by Open Source projects. The study has been conducted on documentation created on the Q&A site StackOverflow, with in-depth research of the two Open Source projects Apache POI and Firefox for Android. The reader will find that documentation provided by the community is in general of good quality and often provided within an acceptable amount of time. One will as well find how projects of different nature can exploit demand driven documentation in different ways. Finally the authors concluded that demand driven documentation could be suitable as a primary resource for information if it is accompanied by other forms of resources. However, the suitability is closely related to the size and activeness of the project's community.

Acknowledgements

We would like to thank our supervisor Morgan Ericsson for the continuous support and guidance which he has given us during our work with this thesis.

Jessica Andersson & Sofia Larsson, Göteborg 20/3/14

Contents

1	Introduction	2
1.1	Purpose	3
1.2	Objective	3
1.2.1	Quality	3
1.2.2	Time to answer	3
1.2.3	Types of documentation	4
1.3	Scope and limitations	4
2	Background	5
2.1	StackOverflow	6
2.2	Apache POI	6
2.3	Firefox for Android	7
3	Related work	8
4	Method	10
4.1	Qualitative assessments	10
4.1.1	Categorisation selection	10
4.1.2	Suitable types of posts	11
4.2	Quantitative measurements	11
4.2.1	Number of answered questions	12
4.2.2	Average score	12
4.2.3	Accepted answers	12
4.2.4	Average time to answer	12
5	Result	13
5.1	Categories	13
5.1.1	Selection	13
5.1.2	Categories	18
5.2	Types of documentation	19

5.2.1	Mapped answers	19
5.2.2	Average quality of answers in categories	22
5.2.3	Consumers of documentation	22
5.3	Quality	23
5.3.1	Number of answered questions	23
5.3.2	Accepted answers	24
5.3.3	Average score	25
5.4	Time to answer	26
6	Discussion	31
6.1	Categories	31
6.2	Types of documentation	32
6.2.1	Mapping answers to categories	32
6.2.2	Average quality of categories	33
6.2.3	Consumers of documentation	34
6.3	Quality	34
6.4	Time to answer	35
6.4.1	Views	36
6.5	General perceptions	37
7	Conclusion	38
	Bibliography	41
A	Questions	42
A.1	POI	42
A.2	Fennec	44
B	Answers	46
B.1	POI	46
B.2	Fennec	49
C	Question-Answers	50
C.1	POI	50
C.2	Fennec	51
D	Categorisation of answers	53
E	Categorisation of POI related answers	68
F	Categorisation of Fennec related answers	71
G	References	73
H	Abstraction of code examples	75

I	Time to answer	76
I.1	Queries	76
I.1.1	Average time to get a correct answer	76
I.1.2	Maximum time to get a correct answer	77
I.1.3	Minimum time to get a correct answer	77
I.1.4	Standard deviation to get a correct answer	77
I.1.5	Median time to get a correct answer	78
I.1.6	Time to get an accepted answer for a certain question	78
I.1.7	Maximum amount of views of a question with at least one answer	78
I.1.8	Minimum amount of views of a question with at least one answer .	79
I.1.9	Average amount of views on questions with at least one answer . .	79
I.1.10	Standard deviation of views on questions with at least one answer	79
I.1.11	Median amount of views on questions with at least one answer . .	79
J	Quality	80
J.1	Queries	80
J.1.1	Number of questions with at least one answer	80
J.1.2	Number of questions with an accepted answer	80
J.1.3	Average number of answers for questions without accepted answer	80
J.1.4	Average max score of answers for questions without an accepted answer	81
J.1.5	Average score of answers	81

Definitions and Acronyms

	Definition
API	Application programming Interface
FAQ	Frequently Asked Questions
Fennec	Firefox for Android
POI	Apache POI (Poor Obfuscation Implementation)
Q&A	Questions and Answers

1

Introduction

DOCUMENTATION IS AN essential part of the process in software engineering projects. The many different kinds of software documentation artifacts should maintain valuable information for evolving a product during its life time. It should also enable a new user or participator to rapidly get started with the project. Without access to relevant information, evolving and maintaining a system will become a complex and time consuming task [1].

The developers in Open Source projects are typically volunteers and also most likely geographically dispersed. It is therefore most probable that the contributors will never have the opportunity to meet face to face. The developers might even be contributing to the project working from different time zones, making it difficult to have a “live” conversation. To ensure that knowledge will be kept within a project with these limitations should documentation be of great importance. Still, in open source projects is documentation often neglected by its contributors. And even if the motivation behind their choice to volunteer in the project is individual, it is generally to produce code, which satisfies the contributor’s own functional needs, and not to produce documentation [2].

When documentation is created, it should as well be maintained. If it is not properly maintained it will eventually no longer be able to serve its purpose [1]. Knowing what to document and on which level could therefore be considered a critical factor for an Open Source project where resources for documentation are limited.

Since the introduction of the Internet, many new channels have emerged as knowledge sources within the field of software development. In the Open Source society a lot of information is created on demand, developers asking for guidance in how to solve their problems in, for example, project specific email threads and internal as well as external Q&A forums, such as StackOverflow [3].

1.1 Purpose

The authors of this thesis believe that a lot of developers and projects today benefit from the complementary documentation that is being produced by volunteers through answering questions on different forums, such as StackOverflow. This thesis aims to research whether this effort could be exploited by Open Source projects to document less and still satisfy the need of documentation.

1.2 Objective

This thesis is conducted as a case study to provide an exhaustive analysis researching if *Demand driven documentation* can work as a substitute for regular documentation of Open Source projects. This question will be answered through looking into two different types of projects and how they are being documented on a *Demand driven documentation* site. These projects will be analysed based on, what the authors believe to be, the characteristics which differentiates Demand driven documentation from regular documentation — such as accessibility and quality. Since the suitability might vary for documentation with different purposes and forms, the research will also explore whether certain types are better suited to be documented in this form than other.

1.2.1 Quality

Two core quality attributes of documentation is that it should be maintained and up-to-date [1]. But documentation found on forums such as StackOverflow can be written by anyone. Answers may be of low quality and since authors are not obliged to update their posts, they will most likely become out of date. StackOverflow use different systems to guide developers to relevant answers to their questions. Answers can, for example, be accepted by the asker, and thereby assure other users that the provided answer helped the asker to solve their issue. Answers can also be voted on by other users and the answer's score reflects the least amount of users which has found the answer helpful. By using this information, the authors will answer their first sub question: *Is documentation provided by Demand driven documentation, in general, of good quality?*

1.2.2 Time to answer

In some cases the answer, that the user is looking for, has already been provided by another question and answer. But the user is not always as fortunate and must then ask the question him- or herself and hope that someone will provide an answer in a timely manner. Which raises the following sub question: *Does the time to receive an answer affect Demand driven documentation suitability as a knowledge source?*

1.2.3 Types of documentation

All documentation might not be appropriate to be presented in the form of Demand driven documentation while some documentation might be well suited for it. This study aims to develop a guidance of how to prioritise what documentation should be developed by the project, and what, if any, could be left for the community to produce. Therefore, is another important sub question to answer: *What kind of documentation is suitable for Demand driven documentation?*

1.3 Scope and limitations

This case study investigates Demand driven documentation created on the Q&A site StackOverflow. Two Open Source projects has been researched in depth; the framework, Apache POI, and the Android application, Firefox for Android. The collected answers related to the two projects is answers the authors encountered during their work with the two projects.

During the research the authors did not find any entries on StackOverflow related to their work with Firefox for Android. This increased their interest in why these two projects differentiated and therefore were all entries, related to the Firefox for Android application, with the tag “Fennec” (the project’s internal name) investigated. This excludes answers related to the Firefox for Android add-on API. The API has been considered to be it’s own project to make a clear distinction between the two researched projects as one framework (Apache POI) and one product (Firefox for Android).

2

Background

SOFTWARE DOCUMENTATION comes in various shapes and sizes. Among these one can find artifacts such as design specifications, in-line comments, javadoc and documentation provided by the code itself. The different types of projects that fits into the description of software development is great and the ways to run these projects are as many. The process of documenting can vary just as much, but it always has one important goal; To contain the relevant information needed to evolve and use the software.

It is time consuming to keep documentation up to date and in a study, conducted at a big telecommunication company among their software engineers [4], it is concluded that software documentation is not likely to be updated in the same pace as the software is changed. This is regardless of what kind of documentation it is with exception of testing and quality related documentation. The theory is supported by a survey from the same study where, according to 68% (where 44% somewhat agree, 24% strongly agree), the following is valid: “Documentation is always outdated relative to the current state of a software system.” [4, p. 36].

Another great challenge when it comes to producing good documentation for a software project, is being aware of what information is valuable for the user, contributor and project team member. As stated earlier, documentation is a time consuming task. Spending time on producing documentation, which might not benefit the project, is therefore an existing risk. If a project would instead focus on producing documentation when needed, on demand, only necessary time would be spent on documenting, to the greatest benefit for the consumers.

2.1 StackOverflow

StackOverflow is a Q&A-site for questions related to coding and programming [5]. All questions are written and answered by users at the site, but anyone is free to access the information. Each question can have zero to many answers. In order to produce a good user experience, where the users easily access appreciated questions and answers, a voting system exists. Members can choose to up- or down-vote questions and answers, depending on the answer’s accuracy. The highest voted answer will be promoted through displaying it directly beneath the question. The user who has asked the question can as well highlight an answer’s accuracy through marking it as the officially accepted solution.

The site is transparent about what sort of questions they find applicable to the site and has produced the following two short lists defining good and bad questions [5].

The Good

- Specific programming problems.
- Software algorithms.
- Coding techniques.
- Software development tools.

The Bad

- Questions you haven’t tried to find an answer for (show your work!).
- Product or service recommendations or comparisons.
- Requests for lists of things, polls, opinions, discussions, etc.
- Anything not directly related to writing computer programs.

2.2 Apache POI

Apache POI, here on referred to as POI, is an Open source project that aims to “create and maintain Java APIs for manipulating various file formats based upon the Office Open XML standards (OOXML) and Microsoft’s OLE 2 Compound Document format (OLE2)” [6]. The project mainly aims to allow reading and writing MS Excel files in Java, but the framework also supports MS Word and MS PowerPoint. POI released its first version in August 2001 and is maintained solely by volunteers [7].

On POI’s project site there is information about the project itself and its different sub modules. A user can also find the API documentation, some code examples and FAQ. There is also guidance of how to get involved with the project as a contributor. Other channels used within in the project is through their instance of the bug management tool Bugzilla and their email list.

2.3 Firefox for Android

Firefox for Android (internally, and here on, known as Fennec) is Mozilla's web browser for the Android platform. The project was first released in the beginning of 2010 [8] and are continuously releasing new versions. The evolution of the project is driven by the mobile team at Mozilla together with the help of anyone who would like to contribute.

There is extensive documentation on the project's wiki [9] including a product plan for each module, development documentation on how to get started, FAQ and in-depth information about certain modules. Valuable information can also be found in their bug tracking tool, Bugzilla. Information is as well floating through the community's many different, and active, IRC channels and email threads.

3

Related work

IN THIS SECTION the reader will find short summaries of research, within the area of software documentation, which have been considered as relevant when conducting this thesis.

Answers from StackOverflow have previously been analysed by Nasehi et al. in their research of “What makes a good code example?” [10]. They present in their paper characteristics that answers, containing code, with four or higher up-votes have in common. An, for this thesis, especially important discovery they made was that the answer is rarely very general, but instead accommodate the special need of the asker. They have also brought up the question of Demand driven documentation’s role as a knowledge source and conclude that it can not be used as a solitary source.

Treude, Barzilay and Storey have as well researched the value of StackOverflow [11], but with a focus on the questions. One of their research questions attends what sort of questions there exist on the website. To answer this, they have chosen a subset of all questions and then identified categories applicable to these questions. Another research question attends how the categories a question belongs to affects the quality and amount of the answers given. One of the conclusions drawn from the result was a mapping done between the category of a question and the existence of an answer or not. It was found that questions belonging to the categories *review*, *conceptual*, *how-to* and *novice* were more prone to receive answers than others [11].

Another similar study of what the researchers refer to as “Crowd documentation” was conducted by Parnin et al. [12]. Their focus on the subject is regarding the coverage of APIs which can be found on a site like StackOverflow and identifying whom the crowd really is. Based on the findings from three different Java APIs with 80 percentage of coverage, they draw the conclusion that it is not viable to trust the crowd to provide full coverage of an API. However, they did believe that higher coverage would be achieved by promoting answers by giving away bounties on unanswered questions [12].

Other researchers who have contributed to this area is Berglund and Priestley [3]. They have also recognised the value of documentation created by the community and have identified email threads and forums as important resources. In order to successfully create Open source documentation, they suggest requirements for a documentation framework which uses these resources. Their approach is, instead of using them as separated resources, assemble these channels and integrate them into a FAQ. This would technically be done by annotating the questions and answers using meta data from which the posts can be recognised. One especially interesting conclusion they have made is regarding the completeness of the documentation. They concluded that coverage should not be considered the main goal of a documentation project, instead should the demand be of greater significance. As well could important user behaviour be extracted from this method, if no one is seeking the documentation of a functionality — the functionality itself might not be of value at all [3].

4

Method

THE RESEARCH HAS been conducted through qualitative and quantitative measurements with data gathered from StackOverflow.

4.1 Qualitative assessments

In order to answer what type of documentation is suitable for Demand driven documentation the thesis writers has acted as experts, using their personal experience from their work with the Open source projects; POI and Fennec. They have made qualitative judgement of what kind of answers they have encountered during their work and mapped these to matching categories which describes the answer's characteristics.

Since the authors were not able to find any categorisation suitable for mapping the different kinds of answers from Demand driven documentation sites were new categories developed within the thesis.

4.1.1 Categorisation selection

In order to create suitable categories for mapping the answers from a Demand driven documentation website, an existing framework was searched for. None was found that appeared sufficiently suitable for the creation of the categories. So within the work of this thesis, an iterative method to extract these categories was developed and is described further in the rest of this section.

Data selection

The data used to build the categories was randomly selected from the StackOverflow database from an assortment based on the following criteria:

1. The answer must have at least one positive vote.
2. The answer must have been posted within the month of March, 2013.

To ensure, to some extent, that the data used to create these categories were of decent quality, only answers with at least one positive vote were considered. In order to try to find the most diversified subset of answers the thesis conductors decided to collect answers from the most active month of the current year, which in this case was March.

Categorising answers

At first an initial set of categories were defined by reasoning and from inspiration of similar studies based on questions on StackOverflow [11]. These categories were then iteratively reviewed by two isolated reviewers. Each iteration consisted of 20 answers to be categorised. Each answer could belong to more than one category. If the reviewer was missing a category, this was added as a suggestion for the following iteration. The result of each categorisation was compared between the two reviewers after each iteration. Where differences emerged between the categorisations, the answer and the involved categories were reviewed together by the both reviewers. When a certain category had remarkably different result, the definition of the category was revised for possible changes to make its description less ambiguous.

4.1.2 Suitable types of posts

The suitable types of posts for Demand driven documentation was researched through mapping all the answers, which have been manually collected during the research. Each answer was also considered to be written to an audience of *Contributors* and/or *End users* of the projects. For every answer it was decided whether it adds knowledge to a contributor, an end user or both.

4.2 Quantitative measurements

All quantitative measurements have been based on the StackOverflow database. Data related to the researched projects has been withdrawn, as well as data from StackOverflow's total population from the sites online Data Explorer [13]. Since this is a "live" resource, which constantly grows, all data queried from the site has been timestamped and can be found in its corresponding appendix.

4.2.1 Number of answered questions

To explore how reliable StackOverflow could be as a primary resource one must know how probable it is that one will receive an answer to their question. Therefore has the rate of answered questions been compiled.

4.2.2 Average score

Since it is important to not only receive an answer, but to get an answer of good quality was the score of each answer collected.

4.2.3 Accepted answers

Data was collected to see how many of the questions asked get an accepted answer. And since one must not accepted an answer as correct were also questions without an accepted answer analysed. This data was used to determine if non-accepted answer still brought value to the users.

4.2.4 Average time to answer

Data has as well been gathered regarding how much time it takes before a question receive an, to the asker, accurate answer. Not all users must ask the questions themselves, instead they can exploit that someone else might have asked the question and received answers. Therefore was the amount of views per question reviewed as well.

5

Result

THIS SECTION presents the results of the qualitative and quantitative measurements that have been conducted on the researched projects. Data withdrawn from the whole data set of StackOverflow has also been presented, when possible, to increase the general understanding of the site.

5.1 Categories

In the following subsection the reader will find some highlights from the elicitation of categories performed by two reviewers, and how the different categories evolved during the selection. A full presentation of all iterations can be found in Appendix D.

5.1.1 Selection

In the first iteration, the reviewers started with a set of six categories, presented in table 5.1. The set was handpicked by the reviewers and is partly inspired by the categories found in [11] with additionally categories the reviewers found applicable from their experience of using StackOverflow as an information resource.

Category	Description
Code example	The answer contains a code snippet explanation.
Code clarification	Explains how a code snippet works.
How-to guides	The answer sequentially describes how to do something.
Error resolving	The answer points out where the problem arises.
Reference	The answer refers to another source.
Design explanation	The answer describes why and/or how a system/API/language is working in a certain way.

Table 5.1: Snapshot of the categories for iteration 1.

After the first iteration, the reviewers agreed that two categories were missing from the set and added *Design suggestion* and *Not an answer*. The later marks posts that is not regarded as answers since they do not provide, and aim to provide, any help to the asker. This does not include invalid answers. *Not an answer* is therefore regarded as an “exclusive or” category, that is, if an answer is categorised as *Not an answer*, it can not be categorised as any of the other categories as well. The refinements before iteration 2 are displayed in table 5.2. The new categories are highlighted in grey.

Category	Description
Code example	The answer contains a code snippet explanation.
Code clarification	Explains how a code snippet works.
How-to guides	The answer sequentially describes how to do something.
Error resolving	The answer points out where the problem arises.
Reference	The answer refers to another source.
Design explanation	The answer describes why and/or how a system/API/language is working in a certain way.
Design suggestion	Advising on a different design.
Not an answer	Not an answer.

Table 5.2: Snapshot of the categories for iteration 2.

Between the sixth and seventh iteration, the description of *Code example* and *Design suggestion* were refined and clarified. In *Code example* it was acknowledged that the description needed to clarify that fluent text in a paragraph, highlighted as code, was not to be regarded as a code example and that only separated sections with code highlighting

belonged to this category. In *Design suggestion* the reviewers found the necessity to clarify that the design suggestion had to distinguish itself from the solution the asker had tried to achieve. The changes can be seen in table 5.3.

Category	Description
Code example	The answer contains a code snippet, that is not in fluent text.
Code clarification	Explains how a code snippet works.
How-to guides	The answer sequentially describes how to do something.
Error resolving	The answer points out where the problem arises.
Reference	The answer refers to another source.
Design explanation	The answer describes why and/or how a system/API/language is working in a certain way.
Design suggestion	Advising on a design/solution, different from the current (if any).
Not an answer	Not an answer.

Table 5.3: Snapshot of the categories for iteration 7.

After the seventh iteration it was acknowledged that although the new description of *Design suggestion* was written together by both reviewers, they still had different understandings of what sort of answers were to be included in this category. To resolve it, the reviewers extended the description to clarify that; new solutions in both code and fluent text should be regarded as a design suggestion. Further on was the description of *Code example* refined, but the meaning stayed the same since the refinement was only verbal. The categories with descriptions used during iteration 8 can be found in table 5.4.

Category	Description
Code example	The answer contains a code snippet, that is not in fluent text.
Code clarification	Explains how a code snippet in either question or answer works. (Not API clarification)
How-to guides	The answer sequentially describes how to do something.
Error resolving	The answer points out where the problem arises.
Reference	The answer refers to another source.
Design explanation	The answer describes why and/or how a system/API/language is working in a certain way.
Design suggestion	Advising on a new design/solution, different from the current (if any), presented in code or text.
Not an answer	Not an answer.

Table 5.4: Snapshot of the categories for iteration 8.

Although the result between the reviews of *Code clarification* was similar in all iterations, it was decided to rewrite the whole description of the category. The reason was that the reviewers found a code snippet containing in-line comments and it was decided to separate these from the code clarification. The resulting description can be found in table 5.5.

Category	Description
Code example	The answer contains a code snippet, that is not in fluent text.
Code clarification	Clarifies a piece of code presented in either the question or the answer, or both, in fluent text.
How-to guides	The answer sequentially describes how to do something.
Error resolving	The answer points out where the problem arises.
Reference	The answer refers to another source.
Design explanation	The answer describes why and/or how a system/API/language is working in a certain way.
Design suggestion	Advising on a new design/solution, different from the current (if any), presented in code or text.
Not an answer	Not an answer.

Table 5.5: Snapshot of the categories for iteration 9.

Between iteration ten and eleven it was decided to rename *Error resolving* to *Pointing out error*. It was decided since solving an error always fell under *Design suggestion* and the reviewers found a need of classifying those who could explain to the asker where their problem arises even if they did not provide a solution for it. Worth to notice is that the category had covered these cases earlier as well, but now the name of the category corresponded to the coverage better. The changes in *Design explanation* was mostly verbal, making the description match the mindset of the authors.

Category	Description
Code example	The answer contains a code snippet, that is not in fluent text.
Code clarification	Clarifies a piece of code presented in either the question or the answer, or both, in fluent text.
How-to guides	The answer sequentially describes how to do something.
Pointing out error	The answer points out where the problem arises.
Reference	The answer refers to another source.
Design explanation	The answer describes the characteristics of a system/API/language.
Design suggestion	Advising on a new design/solution, different from the current (if any), presented in code or text.
Not an answer	Not an answer.

Table 5.6: Snapshot of the categories for iteration 11.

Before the last iteration it was decided to remove the category *How-to guides* from the set. The decision was based on its few occurrences. The understanding of the authors was that the few occurrences was due to that the description was just a very narrow description of how almost all answers on the website were formed, as a description of how to do something, but presented in a certain order. The category then seemed to be more of a quality attribute than defining a certain type of documentation and would therefore not bring value to the study.

5.1.2 Categories

After thirteen iterations, each consisting of twenty answers, the result of each iteration diverged by less than, or equal to 20 percentages, and was then considered mature. The outcome of the iterations was seven categories; *Code example*, *Code clarification*, *Pointing out error*, *Reference*, *Design explanation*, *Design suggestion* and *Not an answer*. The final set of categories is described in table 5.7.

Category	Description
Code example	The answer contains a code snippet, that is not in fluent text.
Code clarification	Clarifies a piece of code presented in either the question or the answer, or both, in fluent text.
Pointing out error	The answer points out where a problem arises.
Reference	The answer refers to another source.
Design explanation	The answer describes the characteristics of a system/API/language.
Design suggestion	Advising on a new design/solution. Different from the current, if any, presented in code or text.
Not an answer	Not an answer.

Table 5.7: Definitions of categories gathered from elicitation.

5.2 Types of documentation

Beneath the reader will find the result from the categorisation of the researched answers from POI and Fennec presented, highlighting the most occurring and diverse categories. As well as their average quality, which have been judged by the StackOverflow community. Finally the reader will also find the distribution of the consumers of the documentation.

5.2.1 Mapped answers

The answers gathered regarding the POI and Fennec projects were manually mapped to their corresponding categories. The distribution is presented in table 5.8. See the complete categorisation of POI and Fennec related answers in Appendix E and F.

Category	POI (%)	#	Fennec(%)	#	Average(%)
Design suggestion	68.12	47	50.00	13	59.06
Reference	53.62	37	53.85	14	53.73
Design explanation	37.68	26	65.38	17	51.53
Code example	40.58	28	23.08	6	31.83
Pointing out error	11.59	8	30.77	8	21.18
Code clarification	15.94	11	15.38	4	15.66
Not an answer	5.80	4	0	0	2.90

Table 5.8: The amount of answers containing the corresponding category.

Design suggestion

The most repeatedly found characteristic was answers providing a new way to solve the given problem.

Reference

Many answers contains references to sources outside the answer. These are often used as a compliment to the other categories, guiding the reader to where more information can be found. In a few cases (7.47%), a reference is the only information given back to the asker. Such answers have in average a lower score than other answers on the site (0.57 compared to 2.12) and only one have been accepted. Detailed data can be viewed in Appendix B.

The references in the answers related to the two researched projects appears to be of different nature. 64.29% of the references in the analysed Fennec answers refers to information hosted by the Mozilla foundation. The corresponding percentage for POI related sites is 29.73%. The data is presented in Appendix G.

Design explanation

Answers describing the characteristics of the given project was the category which differed the most between the researched projects. POI, the API, had only 37.68% answers with explanatory content. While this was the most occurring category (65.38%) for the application Fennec.

Code example

One of the categories frequently found among the POI answers was code examples. Through these the developers communicate everything from fault descriptions to ideas of new ways to handle a problem. Code examples have, based on the sample data, as

well proven to be a very attractive way of communicating an answer. More than 77 percentages of the accepted answers (see Appendix B) related to POI consisted of at least one code example. In Fennec related answers, conversely, code examples are rather rare.

During further analysis of the answers related to POI it appeared to the reviewers that many of these examples were very specific to the question asked. And the questions might have been asked since detailed examples of solving that particular problem was not existing. This theory was tested through comparing the 28 code examples related to POI with the ones provided in the project's own documentation to determine their level of abstraction. Pursuing this, each code example from the answers was compared to the ones provided on the project site with the following alternatives:

1. There is no code example on the site describing the requested functionality. (No example)
2. The provided code example gives a more general use case, not describing this particular case and/or context, compared to the requested functionality. (Lower abstraction level)
3. The provided code example is the same or similar to the requested functionality. (Equal abstraction level)
4. The provided code example is more narrow than the requested functionality. (Higher abstraction level)

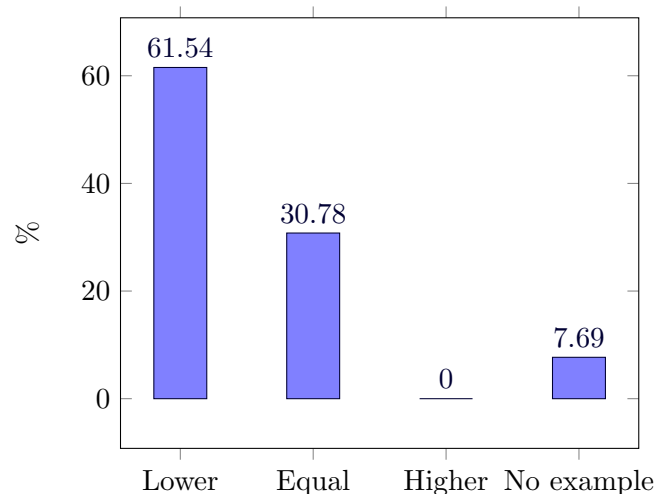


Figure 5.1: The distribution of the 28 code examples' abstraction level compared to the POI's own documentation.

As presented in figure 5.1, most of the code examples were more detailed than the ones provided by the project itself. The majority of the other examples, around 30 percentages, had the same level of abstraction and were at times even an exact replica of the code example given on the project site.

5.2.2 Average quality of answers in categories

The quality of an answer have been measured by its score. In order to measure the average quality of a category, the score for each answer was collected, summarised for each category and then divided by the number of answers containing the category. The result is presented in table 5.9

Category	POI	Fennec	Average
Design explanation	13.42	0.71	7.06
Reference	10.05	0.64	5.35
Code example	4.89	0.67	2.78
Design suggestion	3.87	1.00	2.44
Code clarification	3.91	0.25	2.08
Pointing out error	2.38	0.13	1.25
Not an answer	1.50	0	0.75

Table 5.9: The average number of votes for each category

5.2.3 Consumers of documentation

To get a better understanding of the audience on StackOverflow, each of the answers was considered from the view of different types of users. It was established if the documentation in the answers, were to be documented by the organisation itself, could be considered to be written for an *End user* or a *Contributor*. The characteristics of an End user variates greatly between the two projects. An End user of Fennec was considered to be a user of the application, while End users of POI was considered as developers using the API in their projects. A contributor, on the contrary, would be considered, independent of project, to be someone who is working with the project's source code.

The data in table 5.10 was collected from Appendix B and displays the distribution of the consumers of the answers collected from each project. The reader will find that most of the answers related to POI were written for End users, conversely for Fennec were answers written to Contributors more common. Some answers did not fit into either of the consumer types. These are answers which, even though they could provide great

insights, could not be expected to be documented by the project itself. This could for example be language specific answers or domain knowledge.

Project	End users (%)	#	Contributors (%)	#	Not closely related (%)	#
POI	40.58	28	21.74	15	47.48	30
Fennec	30.78	8	42.31	11	30.78	8

Table 5.10: The distribution of consumers

In some cases both Contributors and End users can make use of the same documentation. In POI the consumers have an overlapping of 5.80%, while in Fennec the rate is of 3.85%.

5.3 Quality

In the following subsection the reader will find the result of the analysis made on the quality of the answers given on StackOverflow both in general and the researched projects. This will be analysed from the rate of answered questions, accepted answers and highest rated non-accepted answers.

5.3.1 Number of answered questions

In a snapshot of the database of all questions at StackOverflow there was a total number of 6,776,503 questions. Out of those, 6,048,918 questions have received at least one answer. The rate of POI and Fennec is presented in table 5.11.

	Answered questions (%)	#	Questions with accepted answers (%)	#
General	89.26	6048918	58.61	3971921
Fennec	73.07	19	29.63	8
POI	91.18	31	52.94	18

Table 5.11: Rate of answered questions and questions with an accepted answer.

5.3.2 Accepted answers

Out of all questions asked at StackOverflow with at least one answer, 3,971,921 questions have been marked to have an accepted answer, i.e. a solution for the asker, resulting in a rate of 58.61% for accepted answers.

The reader can in table 5.11 find that the quality of the POI sample is similar to the whole population. While Fennec related answers has both less answered questions, and as follows, less accepted answers. For both the population and the two sample sets, one will notice that even though many questions have been answered, not as many have an accepted answer. Since it is optional to mark an answer as accepted, data regarding the questions with at least one answer, but with none accepted, have also been gathered and is presented in figure 5.2.

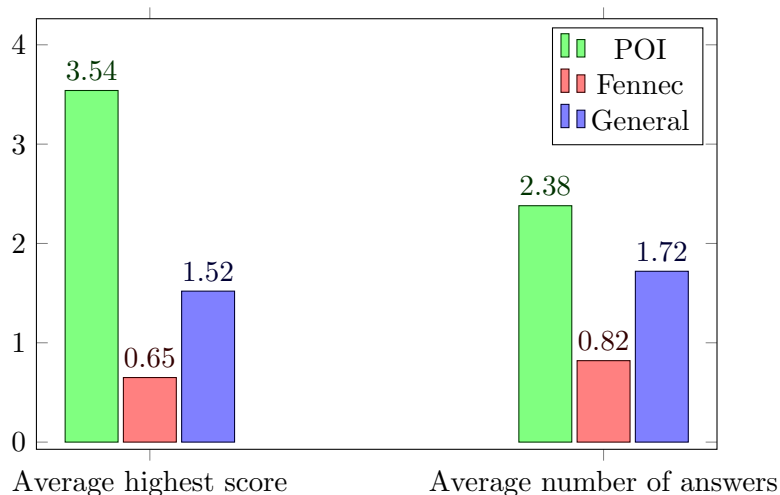


Figure 5.2: Average highest score and number of answers for questions with no accepted answer.

One will in 5.2 find that in average a question, without an accepted answer, has a top rated answer with a score of 1.52. POI related questions has twice as good top answers. Fennec, conversely, has an average score of less than one for the highest rated answers. The relation between the two sample sets and the population for the average number of answers follow the same curve.

As have been presented in table 5.11, the rate of accepted answers in Fennec differentiates from those in POI. In order to explore whether the existence of a certain category has an impact on the acceptance of the answer, the distribution of acceptance over categories was collected and is presented in table 5.12.

Category	POI (%)	#	Fennec (%)	#	Average(%)
Design suggestion	83.33	15	50.0	4	66.67
Reference	50.00	9	62.5	5	56.25
Design explanation	27.78	5	62.5	5	45.14
Code example	77.78	14	0	0	38.90
Pointing out error	16.67	3	25.0	2	20.83
Code clarification	38.90	7	0	0	19.44
Not an answer	5.55	1	0	0	2.78

Table 5.12: The distribution of accepted answers over categories.

5.3.3 Average score

The average score of all answers for both projects differentiates a lot from the average score of the entire site. Fennec answers have in average less than half of the mean score. The average of answers in POI, conversely, is more than three times as great.

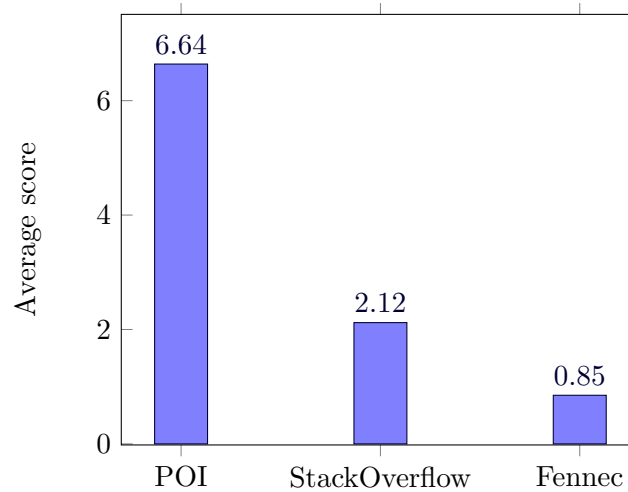


Figure 5.3: The average score on the different data sets

5.4 Time to answer

	Min	Max	Mean	Std dev	Median
General	0.00	2816270.84	10104.14	72280.80	25.22
POI	3.66	1686.67	197.73	447.37	55.40
Fennec	5.30	312109.66	65732.13	118694.51	131.79

Table 5.13: Time, in minutes, to receive an accepted answer.

In table 5.13 one can see data regarding how much time it takes to receive an answer in general and in the corresponding projects. It is a grand variety in how much time it takes to receive an accepted answer for all samples. Therefore has outliers, for POI and Fennec, been identified to give a more realistic view on the average time.

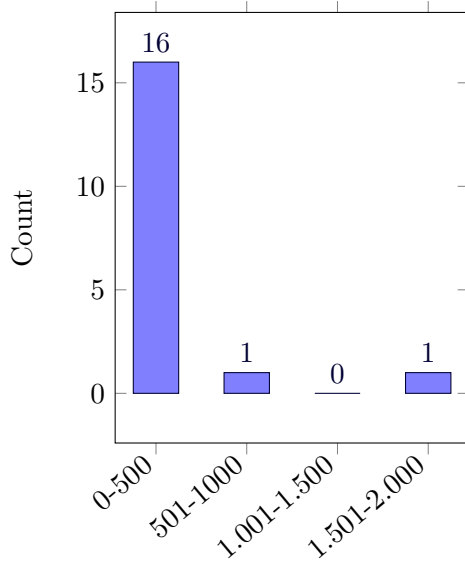


Figure 5.4: Time, to receive an accepted answer for POI related questions, in minutes.

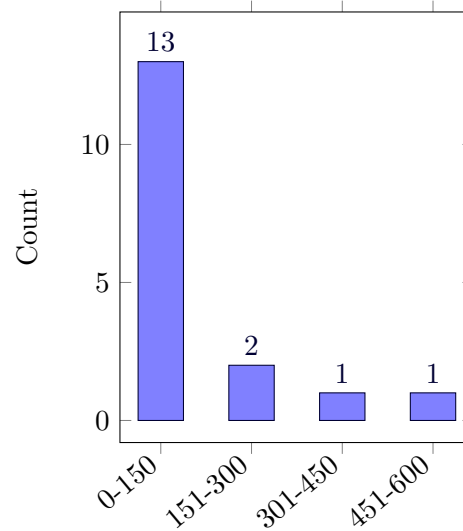


Figure 5.5: Time, to receive an accepted answer for POI related questions after removing outliers, in minutes.

One outlier was detected, with a confidence of 95% using the Dixon Q-test, in the POI related sample set. After removing the outlier one can find the new distribution in figure 5.5 with a mode showing that an accepted answer will most frequently be received within 106.8 minutes. In this case the mode is very close to the mean, 110.1 minutes, and contradict the hypothesis that the mean would be misleading to use. No outliers were detected in the Fennec data set. The mode for Fennec is displayed in figure 5.6, with a peak of 26099.2 minutes, approximately 18 days, which is the time range most

questions are likely to be answered within. In this case could the mean, which is closer to three times as great, be regarded as an unrealistic representation and the mode could be consider to give a more representative picture. The data, after removal of outliers, can be viewed in table 5.14.

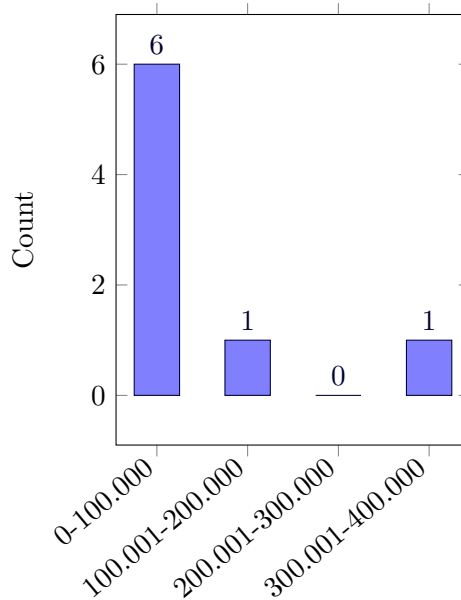


Figure 5.6: Time, to receive an accepted answer for Fenec related questions, in minutes.

	Min	Max	Mean	Std dev	Median	Mode
POI	3.66	562.59	110.14	152.26	41.44	3.66-106.80
Fenec	5.30	312109.66	65732.13	118694.51	131.79	5.30 - 26099.15

Table 5.14: Time, in minutes, to receive an accepted answer after removal of outliers.

Looking at the general time to receive an accepted answer on the site, and its great variation, should the median be considered a more representative number. The median shows that at least half of the questions asked on the site will receive an answer within 25.2 minutes.

To reduce duplicated questions, it is possible to suggest a merge between two similar questions. If a moderator consider the suggested content to be similar enough, the entries will be merged into one, choosing the, according to the moderator, best formed question. It is therefore possible for answers to have an earlier creation date than the asked question. The merged questions can therefore give an improper view of the time

to receive an accurate answer. Unfortunately can these questions not be identified by the public database. To reduce the most extreme outliers have answers created before the asked question, time to answer is negative, been disregarded when collecting data.

Fortunately one must not always be the one asking the question. One could instead take advantage of that somebody else might already been in demand for the same information, and that an answer could be found as soon as the post has been located. Therefore has the number of views per post been researched and are displayed in table 5.15.

	Min	Max	Mean	Std dev	Median
General	2	1712323	1324.12	6179.36	313
POI	169	150478	11101.35	29744.84	1871
Fennec	52	1649	348.68	380.96	238

Table 5.15: Amount of views per question with at least one answer.

Due to the great variance in the samples were the authors curious about the distribution. A more detailed look into the distribution of the views for the two projects can be viewed in figure 5.7 and 5.9.

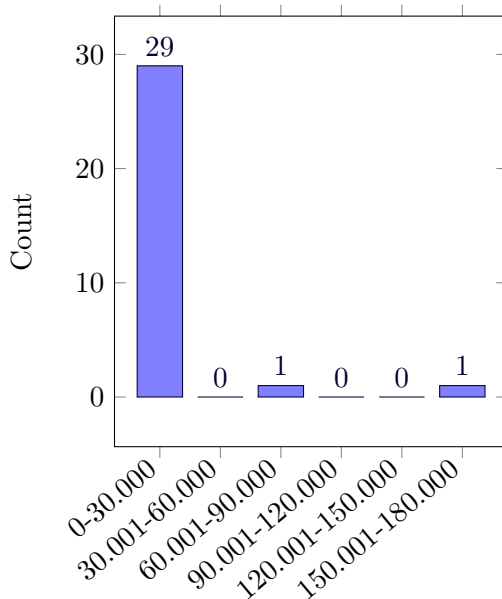


Figure 5.7: The view distribution for POI.

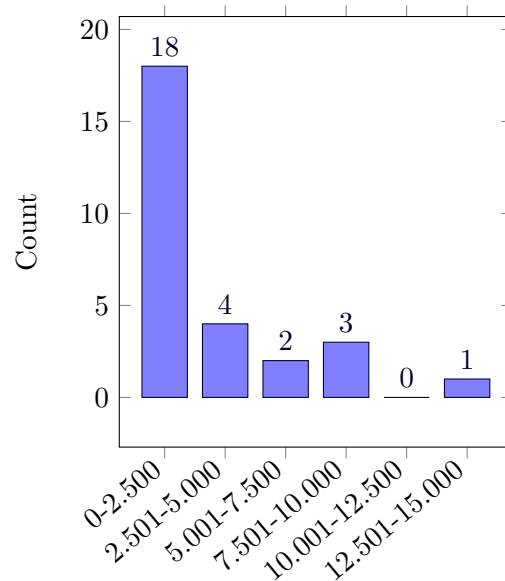


Figure 5.8: The view distribution for POI after removing the outliers.

In figure 5.7 the reader can find that the majority of the POI posts has from zero to 30,000 (or 169-24.455, to be more exact) views, but also two potential outliers. If so, by

removing these, a more realistic representation could be found. In order to confirm this hypothesis an outlier detection test was performed. Due to the size of the sample set, the sample can be considered to have a normal distribution. After performing Grubb's outlier test, it can be found that the three questions with the greatest amount of views can, with a 95% confidence, be considered as outliers and be removed from the sample. The average amount of views of POI without outliers is then 3166.5 and a mode can be found between 169 and 2492. From the perspective of the mode, the mean would be an unrealistic representation of the distribution. The median, conversely, could describe the general amount of views better. Figure 5.8 shows the distribution of the views after removing the outliers and the data regarding the two projects, without outliers, can be found in table 5.16.

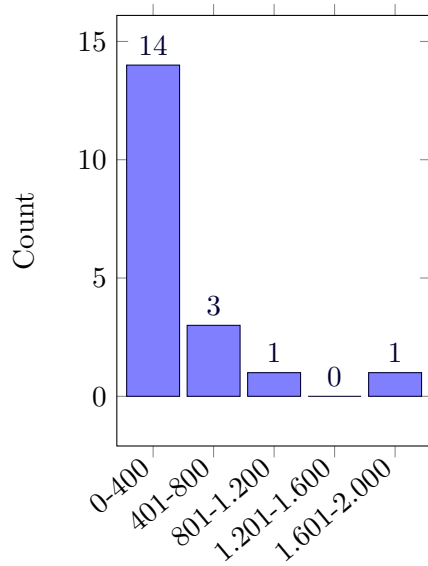


Figure 5.9: The view distribution for Fenec.

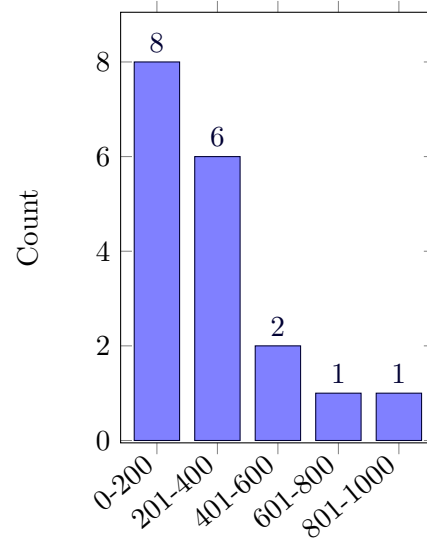


Figure 5.10: The view distribution for Fenec after removing the outliers.

The authors suspected that outliers might be present in the distribution of Fenec related posts, presented in figure 5.9. A Dixon Q-test showed that, with a 95% confidence, one outlier was present. After removing the outlier, a more realistic mean of 276.4 views could be found as well as a mode between 52 and 182. The new distribution of the posts is presented in figure 5.10.

	Min	Max	Mean	Std dev	Median	Mode
POI	169	13494	3166.5	3615.11	1369.5	169-2492
Fennec	52	838	276.44	220.64	226	52-182

Table 5.16: Amount of views per question with at least one answer.

6

Discussion

IN THE FOLLOWING SECTION the reader will find a discussion of the authors interpretation of the result presented in the previous chapter.

6.1 Categories

Since no relevant set of categories was found during the literature study, categories was elicited during thirteen iterations, each consisting of twenty questions. The fact that several of the original categories are still present in a modified form, could have affected the result more than intended. The answers reviewed during each iteration was randomly selected from all answers posted in March 2013. The time span was selected since it was the most active month of the previous year and therefore considered to be the month with most probable diversity. Since the set is from a specific time frame, it could be affected by trends outside of the authors knowledge. The trends are most likely to affect which technology the answers concern and not their presentation, and therefore not affect the final set of categories.

To ensure the result to be as objective as possible, the reviewing was conducted with the two reviewers separated from each other. But due to the fact that the reviewers are of the same academic background and have similar experiences, the result of the elicitation could be biased.

While the thesis conductors were interested in categories defining the characteristics of answers at StackOverflow, another option would have been to use documentation artifacts which frequently are used within software development projects. The content of the answers could then have been mapped to artifacts. But answers are not likely to be presented as a single artifact but as a composition of information, related to many different types of artifacts, required to solve the askers problem. Therefore were characteristics considered to be more valuable for the research.

6.2 Types of documentation

6.2.1 Mapping answers to categories

The, in average, most frequently occurring category in the answers was *Design suggestion*. The answers, marked as a Design suggestion, advice on a new design or solution to help solving the askers problem. Due to the attributes of StackOverflow, aiming for people to ask for help and guidance, the authors did not find it surprising that this characteristic had such a high occurrence. Thus it is to be expected that Design suggestion is the most frequent occurring category in this study.

Overall, a high rate of references can be found in the answers, which could indicate that the projects do not exploit the possibilities of Demand driven documentation. Many of the references as well refer to pages connected to the projects, even though the distribution differs between the projects. More than twice the rate of references, leading back to the project site, exist in the answers connected to Fennec than in the answers connect to POI. This is a remarkable difference and it is believed that it can be directly connected to the information found at the project sites and what kind of project it is; A framework or a product. Mozilla runs quite a covering site harbouring a lot of information about the Fennec project and given that the information sought in answers very seldom is code examples, which is the only non-abstract information, it seems logical that references in answers connected to Fennec leads back to the project site.

Two categories that differentiates greatly are *Design explanation* and *Code example*. Code examples is close to twice as frequent in POI than in Fennec, and for Design suggestion it is the other way around. It is believed, by the authors, that this is not by chance, and that the explanation lies within the difference of the nature of the two projects. Since POI is a framework it is understandable that there exists a need of code examples describing how to use the API, while Fennec is a product and thus has lesser need of code examples. Even though, while having less need of code examples, Fennec is a big and complex product and thus the need to explain how different parts of the product works is to be expected.

Looking at the distribution of Design explanation versus Code example in the projects and putting those in relation to references leading back to the projects, a connection between high rate of Design explanation and references to the project sites can be found.

After performing the comparison between the code examples found at StackOverflow with the ones provided by POI one can agree with Nasehi et al. [10] that these types of answers are in general very specific, facilitating a certain need requested by the asker. Another remarkable rate to acknowledge is the large amount of code examples which were of the same abstraction level, and even at times an exact replica, as the examples provided in the projects' own resources. This could, for example, indicate that the information is hard to find on the projects' provided documentation or that the user favours using StackOverflow as a tool to find answers to their questions.

6.2.2 Average quality of categories

Each category was reviewed from a quality perspective, using the average voting score and accepted answer rate, in order to decide on what type of documentation is suitable as Demand driven documentation. The top categories of the two projects were *Design explanation*, *Reference*, *Code example* and *Design suggestion*.

The *Design explanation* category consists of answers providing information about the given system, API or language. The answers containing this type of characteristic had the highest average score and was the third most occurring attribute in accepted answers. The average score indicates that the information with this characteristic is sought by many and that the information existing at project sites might not impact the demand of further explanations. The accepted answer rate of the two researched projects varies greatly. Design explanation is the most accepted category for Fennec, shared with *Reference*, which indicates that this often is the type of answer the asker is looking for. While for POI related questions, this attribute is only half as common in an accepted answers, which supports the theory of Design explanation being more suitable for a product than an API. This, together with the authors knowledge about the two project's own documentation, could confirm that POI's documentation is not as elaborated as Fennec's, and therefore must information be sought elsewhere. A remarkable detail about Design explanation is that POI's average score is 13.42 and Fennec's is only 0.71. As mentioned earlier, it is believed that the characteristic is an important one for both projects and thus a similar score would be expected. Since the scores differs so greatly it suggests that the Fennec community is smaller and less active at StackOverflow compared to the POI community.

The category with the second highest average score and accepted answer is *Reference*. This might be surprising to the reader since answers which only consists of a reference, as presented earlier, had in average a lower score than other answers. But considering that a reference could be used as a guidance to where a user can find more detailed information about the subject, it could be of great value when presented in combination with other attributes.

Some other interesting categories, from a quality perspective, is *Code example*, *Design suggestion* and *Code clarification*. Their importance vary between the projects; POI answers with code examples or code clarification are, relatively, higher rated than design suggestion and vice verse. A possible explanation could once again be recognised in the nature of the two projects, POI as a framework and Fennec as a product and their StackOverflow users. A code example, or clarification, is of great interest to someone trying to learn a new framework. When using, or extending a product, it is not as valuable since the same code should not be written twice. A design suggestion, conversely, is more likely to be of interest to someone who needs guidance when extending a product. Of course contributors to POI do as well fall under this category, and might therefore be another confirmation to that the majority of POI users of the site is End-user, which will be elaborated further in section 6.2.3.

As the reader has probably noticed, it is in general a great difference between the average score of the two projects, which will be discussed further in the Quality section 6.3.

6.2.3 Consumers of documentation

As can be seen from the result in the POI related answers, there is almost twice as many answers directed towards End users compared to Contributors. While looking at the Fennec project, it is the other way around, but with a difference of 1.5 times more answers directed to Contributors than to End users. It is believed that the difference in distribution comes with the differences in projects, that is, POI is a framework while Fennec is a product where the end user does not produce code with the help of the product. Because of the fact that StackOverflow is dedicated to code related questions, not as many of the Fennec related End user questions belong on a forum such as StackOverflow, which could result in the distribution seen in this study.

When the reviewers decided if an answer was directed towards a specific consumer they had different choices; End user, Contributor, End user and Contributor *or* Not closely related. As can be seen in the result, End users and Contributors in POI can make use of the same documentation to a greater extent than in Fennec. Again, this is believed to come from the differences between the projects. And thus one might argue that, in Fennec, the End users' questions do not belong on StackOverflow, and therefore might documentation regarding the use of the product not be suitable as Demand driven documentation on such a forum. Though another site, without the same restrictions, could be a better choice.

6.3 Quality

Most of the questions asked on StackOverflow have at least one answer. But is it possible to actually rely on that the given answer is correct? An asker can highlight a good answer by marking it as an accepted answer. Though, marking a question as accurate is optional, which might explain why only 59% of the questions asked on the entire site have been marked as accepted. And even fewer in the researched projects.

To thoroughly evaluate whether the information on StackOverflow can be trusted, one should, due to the human factor, also consider that answers which have not been accepted to be of value. If not for the asker him- or herself, but for some other developer in search for help. As it appears, the highest voted answers for each question (without an accepted answer) have in average one and a half up-votes. Which shows that at least one developer have had use of the provided answer. In general, of all written answers, the average number of votes is just above two. Which one could interpret as at least two different developers has received valuable information from the given answer. Excluded then is all registered users which did not vote on the answer, as well as the non-registered users which did not have the option at all. Once again the two projects present very

different result, as showed in both Figure 5.2 and Figure 5.3. POI has an average score high above the general average. While Fennec has not even attained half of it. This could be interpreted as Fennec related answer is of less quality than POI answers. Another, probably more likely, explanation might be that the Fennec community is not as active on StackOverflow as developers working with POI and therefore do Fennec related answers not get as many votes.

Another important angle to explore when looking at the quality of the site is the possibility of someone intentionally creating malicious information. But except from the different quality moderators which exist on StackOverflow one can as well flag a post as malicious [14]. When six flags are given to the post it will be removed from the site, and its creator will pay a penalty with their reputation. This does not eliminate the problem of wrongly provided information, but prevents it to last. How this works on other forums should be explored before using it to create documentation on demand.

Another very important factor when considering StackOverflow as an asset for documentation is if the questions asked actually receives an answer? Fortunately does the average question get closer to two answers. Also here (Figure 5.2) can POI be found to have a lot greater values than Fennec, which could be considered another indication of the activity of the two communities.

Another aspect which has been discussed in some related studies [15] [4] is whether correctness is as important as it historically has been stated. In the result of Forward and Lethbridge's survey they find that software professionals view maintenance as one of the most critical factors in determining the effectiveness of the documentation. In contradiction, the professionals also state that outdated documents still can provide relevant information and be used as a valuable resource. As mentioned in the paper could this conflict indicate a misconception between industry and the academia, where students are taught of the significance of maintenance.

6.4 Time to answer

To be able to use StackOverflow as a primary knowledge resource, one must be certain that the information sought can be accessed. Since the information is created on demand, it might not always exist and the developer will have to ask the question him- or herself. To determine if this is a workable situation for a project and a developer who, most likely, is in need of an answer as soon as possible, one must take a look at the time it normally takes to receive an accurate answer. In this period, one must also include the time it would take to create this documentation internally.

As can be seen in table 5.13, in average a question will be answered, and marked as correct, after seven days, which could be an enormous amount of time for someone who might be prevented to continue his or her work. But with a standard deviation of 50

days one could see that some of these questions take a great amount of time to answer, but most likely not all.

When examining the median, it shows that, at least, half of the questions asked on StackOverflow will receive an answer within 25.2 minutes. This could, in the opinion of the authors, be an acceptable amount of time for creating and receiving the answer a developer must have in order to proceed his or her work. But as it depends on the nature of the project and team must this be evaluated per case.

When reviewing the result of the two researched projects, a great variation between the projects can once again be observed. In average will an accepted answer for a POI related question be received after 110 minutes, while an answer related to Fennec is most likely to be received within 18 days. This gives us, once again, a picture of Fennec as a less active community on StackOverflow. The Fennec data set in this case study is a lot smaller than the POI data set, which could effect the results negatively. But since this includes all possible posts, corroborates this only the view of Fennec as less active community on StackOverflow.

Since the launch of StackOverflow in 2008 [16] has the site grown to 2,819,800 users (2014-02-18). The amount of time for a question to get a correct answer when the site was first launched is very likely to have been much greater. It would therefore be very interesting to look into how this has changed over time. It could likely also show that the average time to receive an acceptable answer is even faster today.

6.4.1 Views

As previously mentioned, a user must not always ask his or her question. Often has someone else already asked the same or a similar question, which answers will help the user in solving their problem. As shown in the result presented in table 5.15, each question has 1324 views in average. Due to the likelihood that outliers might skew this number, could the median, 313, instead be considered as a more reliable presentation. Even so, the amount shows the reader that many others have been interested in this type of information. And hopefully have they found an answer to their question in an even shorter amount of time than the original asker.

When taking a deeper look into POI and Fennec one, once again, notices a big difference between the two projects. POI related questions are most likely to be viewed in between 169-2492 times, while Fennec questions are most likely to be visited 52-182 times. Based on this it can be, once again, concluded that developers looking for information regarding POI is a lot more active on StackOverflow than ones looking for Fennec related answers.

6.5 General perceptions

It has, several times through the result, been suggested that Fennec is likely to be a less active community on StackOverflow than POI. A possible reason for this could be the extensiveness of the project's own documentation which, as previously mentioned, is greater than POI's. If the project is thoroughly documented, and its developers are familiar with its structure, there will be no reason for the developers to seek information elsewhere. Also might there be other channels, dedicated for discussing the code related issues, which are more active and therefore more suitable for a Fennec developer. Another important possibility is that the nature of the forum is not as suitable for a project like Fennec, a product, in comparison with POI, a framework.

7

Conclusion

THE PROJECTS STUDIED had a high amount of answers that provided a different solution for the asker. These types of answers were often accepted as well as they held a high average number of votes. Since the two projects had similar results in this area, being so different in their type of project, a conclusion can be drawn that the providing of a different solution is encouraged independent of project. It is likely to come from the characteristics of StackOverflow as a Q&A-site.

During the study, different needs have been identified for different kinds of projects. For one, a great need for explaining the characteristics of the system was identified for product projects. While on the other hand, for framework projects, it was found a greater interest in concrete examples of how to use the framework. Thus a need for different kinds of documentation depending on the type of project can be concluded, based on the results from the study. But in order to draw more general conclusion, a variety of projects needs to be studied.

When looking at the consumers of the documentation in this study, it can be seen that the type of consumers might impact on how well the project can exploit demand driven documentation on StackOverflow. The conclusion is that if the documentation sought is strongly related to code, it would be well suited to use StackOverflow. Otherwise, if the documentation has little or nothing to do with code, a similar site, without the restriction to code related questions, should be explored.

The general quality of answers on StackOverflow is by the authors considered good. Questions has in general one alternative answer, and even though one must not approve an answer, by either accepting or voting, questions have in average two up votes and the majority have been accepted. The variation in quality between the two projects indicate that there is a strong relation between each of the following statistics; number of answers, accepted answers and score, with the activeness of its corresponding community. But to conclude this, similar studies must be performed on other projects.

Does the time to receive an answer affect Demand driven documentation's suitability as a knowledge source? To answer this for a certain project the authors suggests one should investigate the time it takes to first identify the potential need of the documentation, write it down and then compare it to the time it takes to receive an accepted answer. Notable is that there is also a possibility that the pre-created information might be of no use, and the time spent on writing the documentation was wasted. Answers have in general two up-votes and over a thousand views, from which it can be assumed that more people than the asker found the information helpful.

In this study it has clearly been shown that the activeness of the community strongly affects how successful this form of documentation is for a certain project. Before exploiting Demand driven documentation as a primary source one must therefore look at the size of the community and potential activeness on a site like StackOverflow. In future research it would be interesting to look into how big and active a community must be to successfully document upon demand. API's has also been considered to be projects more suitable for this kind of documentation, in comparison with product projects. Due to the similarities between an API and a programming language one could assume this would hold true for programming languages as well. This should in the future be confirmed by conducting similar studies on other projects. It has as well been concluded that not all types of documentation is as suitable for Deman driven documentation as others, and therefore the authors agree with Nasehi et al. [10] that Demand driven documentation should not be used as a solitary source.

Bibliography

- [1] N. Chapin, Trends in preserving and enhancing the value of software, in: Software Maintenance, 2000. Proceedings. International Conference, San Jose, CA, 2000, pp. 6–8.
- [2] C. C. Gacek, B. Arief, The many meanings of open source, Software, IEEE 21 (1) (2004) 34–40.
- [3] E. Berglund, M. Priestley, Open-source documentation: in search of user-driven, just-in-time writing, in: Proceeding SIGDOC '01 Proceedings of the 19th annual international conference on Computer documentation, New York, NY, 2001, pp. 132–151.
- [4] T. C. Lethbridge, J. Singer, A. Forward, How software engineers use documentation: The state of the practice, Software, IEEE 20 (6) (2003) 35–39.
- [5] S. E. inc, Welcome to stackoverflow, accessed on 2 December 2013 (December 2013). URL <http://stackoverflow.com/about>
- [6] T. A. S. Foundation, Apache poi - the java api for microsoft documents, accessed on 5 December 2013 (December 2012). URL <http://poi.apache.org/>
- [7] T. A. S. Foundation, Apache poi - who we are, accessed on 4 February 2014. URL <http://poi.apache.org/who.html>
- [8] M. Foundation, Firefox for nokia n900 release notes, accessed on 4 February 2014 (January 2010). URL <http://www.mozilla.org/en-US/mobile/1.0/releasenotes/>
- [9] Mobile/fennec, accessed on 24 February 2014. URL <https://wiki.mozilla.org/Mobile/Fennec>
- [10] S. Nasehi, J. Sillito, F. Maurer, C. Burns, What makes a good code example?: A study of programming q&a in stackoverflow, in: Software Maintenance (ICSM), 2012 28th IEEE International Conference on, Calgary, 2012, pp. 25–34.

- [11] C. Treude, O. Barzilay, M. Storey, How do programmers ask and answer questions on the web? (nier track), in: Software Engineering (ICSE), 2011 33rd International Conference, Honolulu, HI, 2011, pp. 804 – 807.
- [12] C. Parnin, C. Treude, L. Grammel, M.-A. Storey, Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow, Georgia Institute of Technology, Tech. Rep.
- [13] Stackoverflow database.
URL <http://data.stackexchange.com/stackoverflow/queries>
- [14] S. E. inc, Flag posts, accessed on 21 March 2014 (March 2014).
URL <http://stackoverflow.com/help/privileges/flag-posts>
- [15] A. Forward J, T. Lethbridge C, Master's thesis.
- [16] S. Perez, With debut of web apps q&a site, stack exchange perfects automated site launch process, accessed on 5 February 2014.
URL http://readwrite.com/2010/07/08/with_debut_of_web_apps_qa_site_stack_exchange_perfects_automated_site_launch_process#awesm=~ov10pghTuoXs2V

A

Questions

In the following two tables is data presented regarding the questions related to the two projects. The data presented is:

- Views, the number of times the question has been viewed (including its answers).
- Accepted answer, 1 if the question has an accepted answer — otherwise 0.
- Time to receive an accepted answer, Time, in days, from a question has been posted until an accepted answer has been provided.
- Max score for questions without any accepted answer.

A.1 POI

Question ID	Views (#)	Accepted answer (true/false)	Time to receive an accepted answer (days)	Max score for questions without any accepted answer
	2014-02-17	2013-11-21	2014-02-03	2014-02-04
70947	150478	1	0.003143094135802	-
326941	7271	0	-	18
845492	2492	1	0.234034182098765	-
1010673	5835	0	-	2

1958289	435	0	-	1
3148535	13494	1	0.02878074845679	-
3819421	1290	0	-	1
4212861	80547	1	0.004699151234568	-
4655565	4325	1	0.015101774691358	-
4935228	443	1	0.07414274691358	-
5578535	24455	0	-	7
5692860	265	0	-	1
5842254	3406	0	-	5
5937373	3516	1	0.013618094135802	-
6530191	9849	1	0.00619425154321	-
8063336	1899	1	0.060339544753086	-
9372630	9831	0	-	0
9641844	526	0	-	1
9717009	1449	1	0.048170563271605	-
9925108	8987	1	0.152489274691358	-
10120241	725	1	0.062201350308642	-
11231469	-	0	-	0
11345146	4916	0	-	8
11423861	1066	1	0.390688850308642	-
11952312	1036	1	0.002542322530864	-
12681246	1871	0	-	1
13226694	1040	0	-	1
13507676	-	0	-	-
15748753	831	0	-	-1
17658254	169	1	0.003604205246914	-
17662645	388	1	1.17129780092593	-
18122257	621	1	0.174102469135802	-
18515011	-	0	-	-
18696628	686	1	0.026433564814815	-

A.2 Fennec

Question ID	Views (#) 2014-02-18	Accepted answer (true/false) 2014-02-04	Time to receive an accepted answer (days) 2014-02-03	Max score for questions without any accepted answer 2014-02-04
5270984	309	0	-	1
5696164 (closed)	-	0	-	-
5779001	238	1	0.090463387345679	-
6238057	838	0	-	0
7199548	142	1	0.005627854938272	-
7373273	-	0	-	1
7734007	1649	0	-	1
9262532	184	1	0.031905825617284	
9566128	133	0	-	1
10367959	294	0	-	1
11024394	52	0	-	1
11042155	493	0		4
11064680	570	1	130.087029243827	-
11280398	-	0	-	No existing answer
11313709	-	0	-	0
11517695	606	1	0.092574035493827	-
12091562	68	0	-	0
12108164	319	0	-	1
12109405	286	0	-	0
16321438	-	0	-	0
16350353	-	0	-	0
16624255	104	1	216.742817708333	-
16751153	214	1	18.1244099537037	-
16818717	69	1	0.003682175925926	-
20854778	-	0	-	0

20885461	57	0	-	1
21266972	-	0	-	0

B

Answers

In the following two tables is data presented regarding the answers related to the two projects. The data presented is:

- Score, the score of the answer.
- Accepted answer, 1 if the answer is an accepted answer — otherwise 0.
- End user, 1 if the answer is written to an end user — otherwise 0.
- Consumer, 1 if the answer is written to a consumer — otherwise 0.

B.1 POI

Answer ID	Score	Accepted answer (true/false)	End user (true/false)	Consumer (true/false)
	2013-11-21	2013-11-21	2014-01-30	2014-01-30
70976	36	1	0	0
71002	1	0	0	0
71084	8	0	0	0
326944	9	0	0	0
326947	1	0	0	0
326952	8	0	0	1
326955	6	0	0	0

326961	1	0	0	1
326965	6	0	0	0
326971	18	0	0	0
359227	4	0	0	0
502146	0	0	0	0
846065	2	1	1	0
1867839	0	0	1	0
1886848	2	0	0	1
2101832	1	0	0	0
3148572	1	0	0	0
3148849	11	1	1	0
3175573	1	0	0	1
4212908	262	1	0	1
4655716	5	1	0	1
4936369	2	1	0	0
4936527	4	0	0	0
5251032	2	0	0	0
5578611	4	0	1	0
5578641	1	0	1	0
5578666	7	0	1	0
5578864	1	0	1	0
5692880	1	0	0	0
5937590	17	1	1	0
6530243	5	1	1	0
6530305	5	0	1	0
7679109	1	0	1	0
7871056	3	0	0	1
8064397	5	1	1	0
8170894	5	0	0	1
8207156	1	0	1	0
8432023	2	0	0	0
9650459	1	0	0	1

9718034	1	1	1	0
9929030	0	1	0	0
10121623	1	1	1	0
10310946	0	0	1	0
10314524	0	0	1	1
10434668	-1	0	0	0
11215461	1	0	1	0
11345859	8	0	1	1
11429842	1	1	0	0
11952362	3	1	1	0
11952418	1	0	1	0
12120823	1	0	1	0
12478754	-2	0	0	0
12592078	3	0	0	0
12681399	2	0	0	1
13226765	1	0	0	0
13494051	3	0	1	0
13931226	0	0	0	0
14401990	0	0	1	1
17658362	1	1	1	0
17668330	1	0	0	0
17671141	0	0	0	0
17688922	0	1	0	0
17766264	0	0	0	0
18127312	2	1	1	0
18526671	0	0	0	0
18697284	1	1	1	0
18992166	-1	0	0	0
19537135	1	0	1	1
19794112	0	0	0	1

B.2 Fennec

Answer ID	Score 2014-02-04	Accepted answer (true/false) 2014-02-04	End user (true/false) 2014-01-30	Consumer (true/false) 2014-01-30
5271228	1	0	0	1
5780267	0	1	0	1
6733810	0	0	0	1
7199585	2	1	0	1
7735679	1	0	1	0
9263300	1	1	1	0
9566505	1	0	0	1
10369613	1	0	0	0
11028006	0	0	1	0
11042197	1	0	0	0
11128943	4	0	0	0
11128985	1	0	1	0
11519696	2	1	1	0
11545767	0	0	1	0
12091679	0	0	0	1
12144919	1	0	0	1
12296078	0	0	0	1
12987122	0	0	0	0
13021822	0	0	0	0
13054437	0	1	0	0
14575065	0	0	0	1
16637255	0	0	0	0
16818834	1	1	1	0
17072877	3	1	0	0
20716136	1	1	0	1
20895820	1	0	1	1

C

Question-Answers

The following two tables present each reviewed question and its corresponding answer(s).

C.1 POI

Question Id	Answer Id	Question Id	Answer Id
70947	70976	5692860	5692880
	71084	5842254	8170894
	359227		13931226
	12592078		18526671
	8432023	5937373	5937590
	71002	6530191	6530243
	10434668		6530305
	12478754	8063336	8064397
326941	326971		8207156
	326944		11215461
	326952		12120823
	326955	9372630	10310946
	326965		10314524
	326947		14401990
	326961	9641844	9650459

	502146		9717009	9718034
845492	846065		9925108	9929030
1010673	1886848		10120241	10121623
	19537135		11231469	
	1867839		11345146	11345859
	19794112		11423861	11429842
1958289	2101832		11952312	11952362
	3175573			13494051
3148535	3148849			11952418
	3148572		12681246	12681399
3819421	7679109		13226694	13226765
4212861	4212908		13507676	
	18992166		15748753	17766264
4655565	4655716		17658254	17658362
	7871056			17671141
	5251032		17662645	17668330
4935228	4936527			17688922
	4936369		18122257	18127312
5578535	5578666		18515011	
	5578611		18696628	18697284
	5578641			
	5578864			

C.2 Fennec

Question Id	Answer Id	Question Id	Answer Id
5270984	5271228	11280398	
5696164		11313709	
5779001	5780267	11517695	11519696
6238057	6733810		11545767
7199548	7199585	12091562	12091679

7373273		12108164	12144919
7734007	7735679		14575065
9262532	9263300	12109405	12296078
9566128	9566505	16321438	
10367959	10369613	16350353	
11024394	11028006	16624255	16637255
	11128985		20716136
11042155	11042197	16751153	17072877
	11128943	16818717	16818834
11064680	12987122	20854778	
	13021822	20885461	20895820
	13054437	21266972	

D

Categorisation of answers

The table D.1 displays the rate of disagreement between the reviewers for each category and iteration. The disagreement consisted of one thinking an answer belonged to a category while the other thought not. The disagreements were solved by both sitting down and going through each question they had split opinions about and selecting one opinion that they agreed upon after discussion. In table D.1 have the cells with results considered bad, been marked with red, and the iterations where the a category was not used the cell is marked with grey.

Following the reader will find the result of each iteration.

Iteration #	Code example	Code clarification	How-to guides	Error resolving/- Pointing out error	Reference	Design explanation	Design suggestion	Not an answer
13	0	0.15	-	0.1	0	0.1	0.1	0
12	0	0.2	0.1	0.2	0	0.2	0.05	0.05
11	0	0.1	0.05	0	0	0.05	0.1	0
10	0	0.1	0	0.15	0	0.05	0.35	0
9	0	0.25	0	0.1	0.05	0.3	0.3	0
8	0	0.1	0.1	0.2	0	0.1	0.25	0
7	0	0.15	0.1	0.1	0.2	0.2	0.1	0.05
6	0	0.15	0.1	0.2	0	0.1	0.25	0
5	0.1	0.2	0.05	0.1	0.05	0.15	0.2	0
4	0	0.3	0.15	0.2	0.1	0.25	0.25	0
3	0.05	0.1	0.2	0	0	0.15	0.25	0
2	0	0.25	0.2	0.1	0.1	0.2	0.25	0
1	0.05	0.15	0.25	0.05	0.2	0.2	-	-

Table D.1: Rate of disagreement

Answer ID	Code ex-ample	Code clarifica-tion	How-to guides	Error re-solving	Reference	Design explana-tion
15432787	0	0	0	1	0	1
15356723	0	0	0	0	0	1
15423170	0	0	0	0	1	0
15423219	1	0	1	0	1	0
15503517	1	0	0	0	0	0
15147620	1	0	1	0	0	0
15688183	1	0	0	0	0	0
15362026	1	0	0	0	0	0
15568368	0	1	0	1	0	0
15397134	1	0	0	1	0	0
15443634	1	1	1	0	1	0
15474450	1	1	0	0	0	0
15358696	0	0	0	0	0	0
15590847	1	0	0	0	1	1
15507504	0	0	0	0	1	0
15602770	0	0	0	1	0	1
15734266	1	1	0	0	1	0
15663419	1	0	0	0	0	0
15639863	0	0	0	0	1	0
15194865	0	0	0	0	1	0

Table D.2: Iteration 1

Answer ID	Code example	Code clarification	How-to guides	Error re-solving	Reference	Design explanation	Design suggestion	Not an answer
15232784	0	0	0	0	1	0	0	0
15336502	0	0	0	1	1	1	0	0
15576058	1	0	0	0	1	0	1	0
15557552	1	1	0	0	0	0	0	0
15144633	0	0	1	0	0	0	0	0
15579966	1	0	0	0	1	0	0	0
15310424	0	1	0	1	0	0	0	0
15560961	0	0	0	0	1	0	1	0
15163346	0	0	0	1	0	1	0	0
15249568	1	1	0	0	1	0	1	0
15707343	0	0	1	1	0	0	0	0
15694188	1	0	0	0	0	0	0	0
15564549	1	1	0	1	0	0	1	0
15186070	1	0	0	1	0	0	0	0
15616914	0	0	0	0	0	1	1	0
15283077	1	1	1	0	1	1	1	0
15147015	1	0	0	1	1	1	0	0
15313060	1	0	0	1	1	0	1	0
15684098	1	0	0	1	1	0	0	0
15195121	0	0	0	1	1	0	1	0

Table D.3: Iteration 2

Answer ID	Code ex-ample	Code clarifica-tion	How-to guides	Error re-solving	Reference	Design explana-tion	Design sugges-tion	Not an answer
15358188	1	0	0	0	1	0	0	0
15528018	1	1	0	1	0	0	0	0
15216103	1	0	0	0	0	0	0	0
15220071	0	0	0	0	1	1	1	0
15292143	0	0	0	0	1	0	1	0
15575214	1	1	0	1	0	0	0	0
15667367	1	0	0	0	0	1	0	0
15455547	1	1	0	1	0	1	0	0
15176540	0	0	0	0	1	1	0	0
15590643	1	0	0	0	1	0	0	0
15696615	0	0	0	1	0	0	1	0
15209289	1	0	1	0	0	0	0	0
15409367	1	1	0	0	1	0	0	0
15182257	1	1	0	1	0	1	0	0
15558176	0	0	0	0	1	1	1	0
15437605	1	0	0	0	0	0	0	0
15142252	1	1	1	0	1	0	0	0
15289889	1	0	1	0	0	1	1	0
15509181	0	0	0	0	1	1	1	0
15247021	0	0	0	1	0	1	0	0

Table D.4: Iteration 3

Answer ID	Code ex-ample	Code clarifica-tion	How-to guides	Error re-solving	Reference	Design explana-tion	Design sugges-tion	Not an answer
15487377	1	0	0	1	0	1	0	0
15255274	0	0	0	1	0	0	0	0
15543535	1	1	0	0	0	1	1	0
15588396	1	1	0	1	1	0	0	0
15579624	0	0	0	0	1	1	1	0
15133928	1	1	1	0	0	1	0	0
15418734	1	0	0	0	0	0	0	0
15294128	1	1	0	0	0	1	1	0
15656077	1	0	0	1	0	0	1	0
15450557	1	1	1	0	0	1	0	0
15557515	0	0	0	0	1	0	1	0
15515606	0	0	0	1	1	0	0	0
15177782	0	0	0	0	0	1	1	0
15563602	1	0	0	0	1	0	0	0
15391766	1	0	1	1	1	1	0	0
15256211	0	0	0	1	1	1	1	0
15590685	1	0	0	0	0	0	1	0
15487018	0	0	0	0	1	1	1	0
15173173	1	0	0	0	0	0	1	0
15262370	1	1	1	1	0	0	0	0

Table D.5: Iteration 4

Answer ID	Code ex-ample	Code clarifica-tion	How-to guides	Error re-solving	Reference	Design explana-tion	Design sugges-tion	Not an answer
15496088	1	0	0	0	1	0	1	0
15298132	0	0	0	0	1	1	1	0
15669010	1	1	0	0	1	1	1	0
15370932	0	0	1	0	0	0	1	0
15612955	0	0	0	1	0	1	0	0
15528658	0	0	0	0	1	0	1	0
15147006	1	0	0	0	0	0	1	0
15726122	1	1	0	1	0	0	1	0
15420953	0	1	0	1	0	0	0	0
15180041	1	1	1	1	0	1	1	0
15204791	1	0	0	1	0	1	1	0
15407726	0	0	1	0	1	1	1	0
15190225	1	0	0	0	0	1	1	0
15272745	0	0	0	1	1	1	0	0
15132141	1	0	0	0	1	0	1	0
15559617	1	0	0	0	0	0	1	0
15370570	1	0	0	0	0	0	0	0
15575269	1	0	0	0	1	0	1	0
15218505	0	1	0	1	0	1	0	0
15678788	1	1	0	1	0	0	1	0

Table D.6: Iteration 5

Answer ID	Code ex-ample	Code clarifica-tion	How-to guides	Error re-solving	Reference	Design explana-tion	Design sugges-tion	Not an answer
15363737	1	0	0	0	0	0	0	0
15624029	0	0	0	1	1	1	0	0
15504794	0	0	0	1	0	0	1	0
15585963	1	1	1	1	0	0	0	0
15466481	1	0	0	1	0	0	1	0
15499852	1	1	0	1	0	0	0	0
15301571	1	0	0	0	1	1	1	0
15519737	0	0	0	0	0	1	1	0
15387531	0	0	0	1	0	1	0	0
15297395	0	0	0	1	0	1	1	0
15544584	0	0	0	0	0	1	0	0
15383704	1	0	0	1	0	0	0	0
15270067	0	0	0	0	1	0	1	0
15453047	1	1	0	1	0	0	1	0
15233189	0	0	1	0	0	0	1	0
15647226	0	0	0	0	1	1	1	0
15271211	0	0	0	0	1	1	1	0
15697309	1	0	0	0	0	0	0	0
15588585	1	1	0	0	0	1	1	0
15195997	1	0	0	1	0	0	1	0

Table D.7: Iteration 6

Answer ID	Code ex-ample	Code clarifica-tion	How-to guides	Error re-solving	Reference	Design explana-tion	Design sugges-tion	Not an answer
15232375								1
15279919	1	0	1	1	1	1	1	0
15127724	1	0	0	0	1	0	1	0
15190745	1	0	1	1	1	0	1	0
15596291	1	0	0	1	1	1	1	0
15400574	1	1	0	0	1	1	0	0
15638188	1	1	1	0	0	0	1	0
15667445	1	1	0	1	0	1	1	0
15409966	1	0	0	0	0	1	1	0
15307368	1	0	0	1	0	0	1	0
15568949	0	0	0	1	1	1	0	0
15425888	1	1	0	0	0	1	1	0
15713338	0	0	0	1	0	1	1	0
15450385	1	0	0	0	0	1	1	0
15258835	0	0	0	0	0	1	0	0
15588771	1	1	1	1	0	1	0	0
15584775	1	0	0	0	1	0	1	0
15464975	1	0	1	1	1	0	1	0
15338351	1	0	0	0	1	1	0	0
15542737	1	0	0	0	0	1	1	0

Table D.8: Iteration 7

Answer ID	Code example	Code clarification	How-to guides	Error re-solving	Reference	Design explanation	Design suggestion	Not an answer
15524813	1	1	0	0	0	0	0	0
15496895	0	0	0	0	0	0	1	0
15557253	1	0	0	0	1	0	1	0
15630391	1	1	1	1	0	1	1	0
15527048	1	1	0	0	1	1	1	0
15652728	1	0	0	0	0	0	1	0
15202510	0	0	0	0	1	1	1	0
15590478	0	0	0	1	1	1	1	0
15530340	0	0	1	0	0	1	0	0
15722653	1	1	0	1	1	0	1	0
15377322	1	1	1	0	0	0	0	0
15597293	1	0	0	0	0	0	1	0
15405828	1	0	0	0	0	0	1	0
15468671	1	0	0	1	0	1	1	0
15422239	1	1	0	0	1	1	1	0
15175758	1	1	0	0	0	1	0	0
15330283	1	0	1	1	0	1	1	0
15377077	1	0	0	0	1	1	1	0
15292650	1	0	0	1	1	0	0	0
15642898	1	0	0	0	0	0	0	0

Table D.9: Iteration 8

Answer ID	Code example	Code clarification	How-to guides	Error re-solving	Reference	Design explanation	Design suggestion	Not an answer
15689494	0	0	0	0	1	0	1	0
15194234	1	1	0	0	0	1	1	0
15369184	0	0	1	0	0	0	1	0
15185002	1	0	0	0	0	0	1	0
15667018	1	1	1	0	0	0	1	0
15630039	1	1	1	1	1	0	1	0
15402069	1	0	0	0	0	1	1	0
15257620	1	1	1	0	0	1	1	0
15140634	0	0	1	1	1	0	1	0
15526634	1	0	0	0	0	1	1	0
15203986	1	1	0	1	0	0	0	0
15239905	1	1	0	0	1	1	1	0
15147702	1	0	0	0	0	0	1	0
15316012	1	1	1	0	0	0	0	0
15407328	0	0	0	0	1	1	0	0
15522846	1	1	1	0	0	1	1	0
15162687	1	1	1	0	1	0	1	0
15638980	0	0	0	1	0	1	0	0
15319007	0	0	0	0	1	1	0	0
15389644	0	1	0	0	1	1	1	0

Table D.10: Iteration 9

Answer ID	Code example	Code clarification	How-to guides	Error re-solving	Reference	Design explanation	Design suggestion	Not an answer
15699323	1	0	0	1	0	0	0	0
15364061	0	0	0	1	0	1	1	0
15537280	0	0	0	1	1	0	1	0
15459884	1	1	0	1	0	0	0	0
15222802	0	0	0	0	0	1	1	0
15341480	1	1	1	0	1	1	1	0
15289787	1	0	0	0	0	0	1	0
15375394	0	0	0	1	1	0	0	0
15391635	1	0	0	0	0	0	0	0
15200147	1	0	0	0	0	0	1	0
15733841	1	0	0	1	1	1	1	0
15161388	1	1	0	0	1	0	1	0
15566652	0	0	0	1	0	1	1	0
15489235	1	1	0	1	0	1	0	0
15149052	1	1	0	1	0	0	0	0
15718604	0	0	0	0	1	1	1	0
15551583	0	0	0	0	0	1	0	0
15416403	1	1	0	0	0	1	1	0
15169117	1	1	0	0	0	1	1	0
15517070	1	0	0	1	0	0	0	0

Table D.11: Iteration 10

Answer ID	Code example	Code clarification	How-to guides	Pointing out error	Reference	Design explanation	Design suggestion	Not an answer
15553924	0	0	0	0	1	1	1	0
15294655	0	0	0	0	0	1	1	0
15658201	1	0	0	0	1	1	1	0
15264556	0	0	0	0	0	1	0	0
15699252	1	0	0	1	0	0	0	0
15503775	1	1	0	1	0	0	0	0
15495326	0	0	0	0	0	1	0	0
15400955	1	0	0	0	0	0	0	0
15339385	1	1	0	1	1	0	0	0
15342193	1	1	0	0	0	0	1	0
15325966	1	0	1	0	1	1	1	0
15567791	1	0	0	1	0	0	1	0
15557727	1	1	0	0	1	1	1	0
15431424	1	1	0	0	0	1	1	0
15465454	0	0	0	1	1	1	0	0
15230652	0	0	0	0	1	1	0	0
15132837	0	0	0	1	0	0	0	0
15488610	0	0	1	0	1	0	1	0
15501480	0	0	0	0	0	1	1	0
15704628	0	0	0	0	0	1	1	0

Table D.12: Iteration 11

Answer ID	Code example	Code clarification	How-to guides	Pointing out error	Reference	Design explanation	Design suggestion	Not an answer
15401231	0	0	0	1	0	0	0	0
15594262	1	1	0	0	0	0	1	0
15136407	1	0	0	0	1	1	1	0
15726088	0	1	0	0	1	1	1	0
15641905	1	1	1	1	1	1	1	0
15681419	0	0	0	0	0	0	1	0
15285664	1	0	1	0	1	1	1	0
15607117	1	1	0	0	0	1	0	0
15140273	0	1	0	0	0	0	0	0
15704416	0	0	0	0	0	1	0	0
15161092	1	1	0	1	0	1	1	0
15516155								1
15219092	1	0	1	0	0	1	1	0
15510196	0	0	0	0	0	1	0	0
15536349	1	1	0	0	0	0	1	0
15248313	0	0	0	0	1	0	0	0
15488312	1	1	0	0	1	1	1	0
15361914	1	1	0	1	0	1	1	0
15280966	0	0	0	1	0	1	1	0
15621397	1	1	0	0	1	0	1	0

Table D.13: Iteration 12

Answer ID	Code example	Code clarification	Pointing out error	Reference	Design explanation	Design suggestion	Not an answer
15473390	0	0	1	1	1	1	0
15143770	1	1	0	1	0	1	0
15455648	1	1	0	0	0	1	0
15366568	1	1	1	0	1	0	0
15299566	0	0	0	0	1	1	0
15457129	0	0	0	1	1	1	0
15613514	1	0	0	0	0	1	0
15533259	0	0	0	1	1	0	0
15532076	1	1	1	0	0	0	0
15642358	0	1	1	0	0	0	0
15127374	0	1	1	0	1	1	0
15614284	0	0	0	0	0	1	0
15260427	0	0	0	0	1	1	0
15700496	1	1	0	0	0	1	0
15238972	1	1	1	0	0	1	0
15471799	1	1	0	1	0	1	0
15716546	0	0	0	0	1	1	0
15407572	1	0	0	0	1	1	0
15544755	0	0	0	1	1	1	0
15721688	1	0	1	1	1	1	0

Table D.14: Iteration 13

E

Categorisation of POI related answers

Answer ID	Code example	Code clarification	Error resolving	Reference	Design explanation	Design suggestion	Not an answer
70976	1	0	0	0	0	1	0
71084	1	1	0	0	0	1	0
359227	0	0	0	0	0	1	0
12592078	1	0	0	0	1	1	0
8432023	0	0	0	0	1	0	0
71002	1	0	0	0	1	1	0
10434668	0	0	0	0	0	1	0
12478754	0	0	0	0	0	1	0
326971	0	0	0	1	1	1	0
326944	0	0	0	0	0	1	0
326952	1	0	0	1	1	1	0
326955	0	0	0	1	1	1	0
326965	0	0	0	0	1	1	0
326947	0	0	0	1	0	0	0
326961	0	0	0	1	1	0	0
502146	0	0	0	0	0	0	1

APPENDIX E. CATEGORISATION OF POI RELATED ANSWERS

846065	1	0	0	0	0	1	0
1886848	0	0	0	1	1	0	0
19537135	0	0	0	1	1	0	0
1867839	0	0	0	1	0	1	0
19794112	0	0	0	1	1	0	0
2101832	0	0	0	1	0	1	0
3175573	0	0	0	1	1	1	0
3148849	1	1	0	1	0	1	0
3148572	0	0	0	1	0	1	0
7679109	0	0	0	1	1	1	0
4212908	0	0	0	1	1	0	0
18992166	0	0	0	1	0	0	0
4655716	1	0	0	0	1	0	0
7871056	0	0	0	0	0	0	1
5251032	1	0	0	0	1	0	0
4936527	0	0	0	1	1	0	0
4936369	0	0	0	0	0	0	1
5578666	1	1	0	1	1	1	0
5578611	1	0	0	0	0	1	0
5578641	0	0	0	1	0	0	0
5578864	0	0	0	1	0	0	0
5692880	0	0	0	0	0	0	1
8170894	0	0	1	0	1	0	0
13931226	1	0	0	0	0	1	0
18526671	0	0	0	0	0	1	0
5937590	1	0	0	1	0	1	0
6530243	1	0	0	0	0	1	0
6530305	1	0	0	1	0	1	0
8064397	0	0	0	1	0	1	0
8207156	0	0	0	1	0	1	0
11215461	1	0	0	1	0	1	0
12120823	0	0	0	1	1	0	0
10310946	0	0	0	0	0	1	0

APPENDIX E. CATEGORISATION OF POI RELATED ANSWERS

10314524	0	0	0	1	1	1	0
14401990	0	0	0	0	0	1	0
9650459	0	0	0	1	0	0	0
9718034	1	1	0	0	0	1	0
9929030	1	0	0	0	1	1	0
10121623	1	1	0	1	1	1	0
11345859	1	1	1	1	1	1	0
11429842	1	1	1	1	0	1	0
11952362	1	1	0	0	0	1	0
13494051	1	0	0	0	0	1	0
11952418	0	0	0	1	0	1	0
12681399	1	1	1	0	1	0	0
13226765	0	0	0	1	0	1	0
17766264	0	0	0	1	0	1	0
17658362	1	0	1	0	0	1	0
17671141	0	0	1	0	0	0	0
17668330	1	0	1	0	0	1	0
17688922	1	1	0	1	0	1	0
18127312	0	0	0	1	1	1	0
18697284	1	1	1	1	0	1	0

F

Categorisation of Fennec related answers

Answer ID	Code example	Code clarification	Error resolving	Reference	Design explanation	Design suggestion	Not an answer
5271228	1	1	0	1	1	1	0
5780267	0	0	1	1	1	0	0
6733810	0	0	1	1	1	1	0
7199585	0	0	0	0	0	1	0
7735679	0	0	0	1	0	0	0
9263300	0	0	1	0	0	0	0
9566505	0	0	0	1	1	0	0
10369613	0	0	0	1	1	0	0
11028006	0	0	0	1	1	1	0
11128985	0	0	0	1	1	1	0
11042197	1	0	0	0	0	1	0
11128943	1	0	1	0	0	0	0
12987122	1	1	1	0	1	1	0
13021822	1	1	0	0	1	1	0
13054437	0	0	0	1	0	0	0
11519696	0	0	0	1	1	1	0

APPENDIX F. CATEGORISATION OF FENNEC RELATED ANSWERS

11545767	1	1	0	1	1	0	0
12091679	0	0	1	0	1	0	0
12144919	0	0	1	0	0	0	0
14575065	0	0	0	0	0	1	0
12296078	0	0	0	0	1	1	0
16637255	0	0	1	1	0	0	0
20716136	0	0	0	1	1	0	0
17072877	0	0	0	1	1	1	0
16818834	0	0	0	0	1	1	0
20895820	0	0	0	0	1	0	0

G

References

The following table shows which answers that contains at least one reference to resources maintained by the community such as; homepage, wiki, issue tracker, API reference.

Id	Maintained by the community	Id	Maintained by the community
326971	0	5578864	1
326952	0	5937590	1
326955	0	6530305	0
326947	0	8064397	0
326961	0	8207156	0
1886848	0	11215461	0
19537135	0	12120823	0
1867839	0	10314524	1
19794112	0	9650459	0
2101832	0	10121623	1
3175573	0	11345859	1
3148849	1	11429842	1
3148572	0	11952418	1
7679109	0	13226765	0
4212908	0	17766264	0
18992166	0	17688922	0

APPENDIX G. REFERENCES

4936527	0		18127312	1
5578666	1		18697284	1
5578641	0			

H

Abstraction of code examples

The following table presents the result of the comparison of abstraction level made between the POI related answers containing code examples and the code examples provided on their own website.

Accepted answer with code example	No example on the website	Lower abstraction level	Same abstraction level	Higher abstraction level
846065	0	1	0	0
18697284	1	1	0	0
10121623	1	0	0	0
3148849	0	0	1	0
11952362	0	1	0	0
13494051	0	1	0	0
5578611	0	0	1	0
5578666	0	0	1	0
11345859	0	1	0	0
5937590	0	1	0	0
9718034	0	0	1	0
6530243	0	1	0	0
6530305	0	1	0	0
Total	1	8	4	0
%	7.692	61.538	30.769	0

I

Time to answer

I.1 Queries

I.1.1 Average time to get a correct answer

The query gives the average amount of days it takes to get a an answer which also has been accepted by the asker. Negative time can occur due to questions which has been merged with previous answers. These questions are not possible to segregate and therefore has all answers created before questions been ignored. Data was collected 2014-02-18.

```
WITH diffs AS
  (SELECT CAST(a.creationdate - q.creationdate AS FLOAT) AS d
   FROM Posts q
   JOIN Posts a ON a.id = q.acceptedanswerid)

SELECT AVG(d) FROM diffs
WHERE d > 0;
```

I.1.2 Maximum time to get a correct answer

The query gives the maximum amount of days it has taken to get a an answer which also has been accepted by the asker. Data was collected 2014-02-18.

```
WITH diffs AS
  (SELECT CAST(a.creationdate - q.creationdate AS FLOAT) AS d
   FROM Posts q
   JOIN Posts a ON a.id = q.acceptedanswerid)

SELECT MAX(d) FROM diffs;
```

I.1.3 Minimum time to get a correct answer

The query gives the minimum amount of days it has taken to get a an answer which also has been accepted by the asker. Negative time can occur due to questions which has been merged with previous answers. These questions are not possible to segregate and therefore has all answers created before questions been ignored. Data was collected 2014-02-18.

```
WITH diffs AS
  (SELECT CAST(a.creationdate - q.creationdate AS FLOAT) AS d
   FROM Posts q
   JOIN Posts a ON a.id = q.acceptedanswerid)

SELECT MIN(d) FROM diffs
WHERE d > 0;
```

I.1.4 Standard deviation to get a correct answer

The query gives the standard deviation of how many days it takes to get a an answer which also has been accepted by the asker. Data was collected 2014-02-18.

```
WITH diffs AS
  (SELECT CAST(a.creationdate - q.creationdate AS FLOAT) AS d
   FROM Posts q
   JOIN Posts a ON a.id = q.acceptedanswerid)

SELECT STDEV(d) FROM diffs
WHERE d > 0;
```

I.1.5 Median time to get a correct answer

The following query provides the median time it takes from that a question has been posted until an accepted answer has been provided. Data was collected 2014-02-18.

```

WITH diffs AS
  (SELECT CAST(a.creationdate - q.creationdate AS FLOAT) AS d
   FROM Posts q
   JOIN Posts a ON a.id = q.acceptedanswerid)

SELECT(
  (SELECT MAX(d) FROM
   (SELECT TOP 50 PERCENT d
    FROM diffs ORDER BY d) AS BottomHalf)
+
  (SELECT MIN(d) FROM
   (SELECT TOP 50 PERCENT d
    FROM diffs ORDER BY d DESC) AS TopHalf
  WHERE d > 0)
) / 2 AS Median;

```

I.1.6 Time to get an accepted answer for a certain question

The following query provides the time it took from that a certain question was posted until an accepted answer was been provided. Data was collected 2014-02-18.

```

SELECT CAST(a.CreationDate - q.CreationDate AS FLOAT) AS d
FROM Posts q
JOIN Posts a ON a.id = q.acceptedanswerid
WHERE q.id = [questionId];

```

I.1.7 Maximum amount of views of a question with at least one answer

This query provides the maximum amount of views a question has gotten, which has at least on answer. Data was collected 2014-02-18.

```

SELECT MAX(ViewCount)
FROM Posts
WHERE AnswerCount > 0;

```

I.1.8 Minimum amount of views of a question with at least one answer

This query provides the minimum amount of views a question has gotten, which has at least on answer. Data was collected 2014-02-18.

```
SELECT MIN(ViewCount)
FROM Posts
WHERE AnswerCount > 0;
```

I.1.9 Average amount of views on questions with at least one answer

This query provides the average amount of views on all questions which has at least on answer. Data was collected 2014-02-18.

```
SELECT AVG(CAST(ViewCount AS FLOAT))
FROM Posts
WHERE AnswerCount > 0;
```

I.1.10 Standard deviation of views on questions with at least one answer

This query provides the standard deviation of views on all questions which has at least on answer. Data was collected 2014-02-18.

```
SELECT STDEV(CAST(ViewCount AS FLOAT))
FROM Posts
WHERE AnswerCount > 0;
```

I.1.11 Median amount of views on questions with at least one answer

This query provides the median amount of views on all questions which has at least on answer. Data was collected 2014-02-18.

```
WITH views AS
(SELECT ViewCount AS v FROM Posts WHERE AnswerCount > 0)

SELECT(
  (SELECT MAX(v) FROM
    (SELECT TOP 50 PERCENT v FROM views ORDER BY v) AS BottomHalf)
  +
  (SELECT MIN(v) FROM
    (SELECT TOP 50 PERCENT v FROM views ORDER BY v DESC) AS TopHalf)
) / 2 AS Median;
```

J

Quality

J.1 Queries

J.1.1 Number of questions with at least one answer

The following query gives the amount of questions on StackOverflow which has at least one answer. Data was collected 2014-03-04.

```
SELECT COUNT(*) FROM Posts  
WHERE AnswerCount >= 1;
```

J.1.2 Number of questions with an accepted answer

The following query gives the amount of questions which has been provided by an answer which the asker has marked as accepted. Data was collected 2014-03-04.

```
SELECT COUNT(*) FROM Posts  
WHERE AcceptedAnswerId IS NOT NULL;
```

J.1.3 Average number of answers for questions without accepted answer

The following query provides the average amount of answers a question has, which has not yet been provided with an accepted answer. Data was collected 2014-02-04.

```
SELECT AVG(Cast(AnswerCount as Float))  
FROM Posts WHERE PostTypeId = 1 AND AcceptedAnswerId IS NULL;
```


J.1.4 Average max score of answers for questions without an accepted answer

The following query provides the average score of the highest voted answer a question has, which has not yet been provided with an accepted answer. Data was collected 2014-02-04.

```
SELECT AVG(Cast(MaxScore as FLOAT))  
FROM  
(SELECT MAX(a.Score) as MaxScore  
  FROM Posts a  
  JOIN Posts q ON q.Id = a.ParentID  
  WHERE q.PostTypeId = 1 AND q.AcceptedAnswerId IS NULL  
  GROUP BY q.Id) AS sub;
```

J.1.5 Average score of answers

The following query returns the average score of all answers. Data was collected 2014-02-04.

```
SELECT AVG(Cast(Score as Float))  
FROM Posts  
WHERE PostTypeId = 2;
```