What Are Practitioners Asking About Requirements Engineering? An Exploratory Analysis of Social Q&A Sites

Zahra Shakeri Hossein Abad, Alex Shymka, Susant Pant, Ashley Currie, Guenther Ruhe
Department of Computer Science, University of Calgary, Calgary, Alberta, Canada
Email: {zshakeri, alex.shymka, susant.pant, ashley.currie, ruhe}@ucalgary.ca

***Abstract*—Requirements Engineering (RE) and all its underlying activities, such as requirements identification, evolution, validation, communication, and management, are still the key factors in successful product development. Therefore, proper implementation of this process is necessary to obtain a quality product. A better understanding of the most challenging RE-related topics for practitioners will greatly help to identify the areas of RE that may require extra attention by researchers and project managers. However, there has been very little experimental work towards identifying a practitioner's needs on the implementation and understanding of RE activities and tasks. Therefore, in this paper, we use data from popular social Q&A sites (i.e. Stack Overflow, Programmers Stack Exchange, Project Management Stack Exchange, and Quora), and analyze 4,553 questions and answers to examine what requirements engineers' needs are and what they ask about. To this end, we applied Latent Dirichlet Allocation-based (LDA) topic models and statistical analysis to explore the main related topics to RE. Our findings show that software practitioners are asking about requirements communication, evolution, validation, and modelling. Furthermore, we determined common RE challenges and issues, identified the main types of questions practitioners ask (i.e. what, how, why, when, where, and RE domain), and categorized these questions based on various aspects of software products (e.g. functionality, quality, and release planning). Our findings help highlight the challenges facing requirements engineers that require more attention from the software engineering - specifically requirements and product engineering - research communities in the future and establish a novel approach for analyzing the content of social Q&A websites.**

I. Introduction

Requirements Engineering (RE) plays a crucial role in any software development process [1]. RE is a large challenge and there are many activities related to it that need to be addressed throughout every software development life cycle. Requirements need to be gathered, specified, verified and validated, modelled, planned out, implemented and changed as needed. The challenge is designing a system that sufficiently addresses the customer's demands. These requirements can often be interpreted differently than their original intentions for better or for worse and more problems become apparent as development progresses. Proper implementation of this process is necessary to obtain a high-quality product. A better understanding of the most challenging RE-related topics among practitioners will greatly help us to identify activities that impede the RE process from being implemented properly. To summarize, this paper's main purpose is not to propose solutions to RE issues we find, but instead to identify the largest problems in RE as identified by practitioners.

The findings of an exploratory empirical study by Kauppinen [2] show that the RE process is a demanding venture, and human factors plays a key role in the success of implementing this process. Furthermore, they proposed that defining simple and specific RE processes and activities, as well as providing the systematic usage of these activities, makes the RE process more beneficial for software development companies. We, therefore, conducted a large-scale exploratory study of social Q&A communities (i.e. Stack Overflow[1], Programmers Stack Exchange[2], Project Management Stack Exchange[3], and Quora[4]) to observe the key areas of discussion about RE among software practitioners. These sites specifically were picked for the volume of content they could provide due to their general popularity; they were superficially also the most popular Q&A sites regarding Software and Requirements Engineering topics.

We applied Latent Dirichlet Allocation-based (LDA) topic models and statistical analysis to explore the main topics of discussions about RE among practitioners. In this paper, we go over our process for collecting and preparing the data set related to RE, analysing our findings by composing topic models from our collection, and conducting a survey with 44 participants to evaluate the accuracy of these topic models. We then examine these topics for their implied meaning, and present them as findings. Moreover, we conducted a statistical analysis by categorizing a sample of the titles we collected.

Our findings shows that the main discussion topics related to RE among software practitioners are (a) a dissatisfaction with the level of communication and interaction between customers and developers; (b) a need for tracking additions and changes to requirements, and organizing them; and (c) a desire for greater communication between executives, clients and developers in order to avoid a tough deadline or bad management. Furthermore, we found that none of these topics are significantly more important over another; nor are they so distinct from each other as to be mutually exclusive. By looking directly at the text of these posts, we also deduced that

[1]http://stackoverflow.com/
[2]http://programmers.stackexchange.com/
[3]http://pm.stackexchange.com/
[4]https://www.quora.com/

978-1-5090-3694-3/16 $31.00 © 2016 IEEE
DOI 10.1109/REW.2016.46

RE 2016 Workshops, Beijing, China
IWSPM Workshop Paper

(1) developers and stakeholders seldom comprehend the requirements elicitation process in certain software development paradigms, specifically in Agile methodologies; (2) developers seek an easy way to transform their clients' requirements into well-defined system functionalities; and (3) project planning should motivate the team and the customers with progress, not cause them to overwork because they are lagging behind schedule. The two significant contributions of this study are as follows:

- Exploring and identifying the main discussion topics about RE among software development practitioners, which provides insights into the different categories and problem areas of RE.
- Conducting statistical analysis on the data set retrieved from Q&A websites to measure the proportions of the types of questions asked about RE. This helps identify the challenges facing requirements engineers that require more attention from requirements and product engineering research and development communities in the future.

The rest of this paper is structured as follows: In Section II, we present related work and discuss the scope and limitations briefly. Section III describes the details of our study methodology, including Research Questions (RQs), Data Collection, and Data Preparation. The details of our data analysis (e.g. topic modelling and statistical analysis) and the evaluation of the results of our data analysis process are described in Section IV. In Section V, we discuss the results and findings of this study based on our research questions. Threats to the validity of the results of our study are discussed in Section VI. Finally, in Section VII, we conclude our study and provide recommendations for future research.

II. Related Work

This section will discuss general research on what practitioners are discussing on social Q&A sites regarding software engineering and requirements engineering.

Barua et al. [3] use topic modelling on StackOverflow posts to identify the main topics that developers discuss on the site, with results showing that developers are most interested in discussions involving web development, mobile applications, Git, and MySQL. Another study by Rosen and Shihab [4] follows a similar topic modelling technique but focuses on mobile developers. Two of their main focuses are what types of issues mobile developers discuss and what types of questions they ask (i.e. what, how, or why). They found that mobile developers mainly ask questions about "how" something should be done. Similarly, Bajaj et al. also use topic modeling to determine what topics web developers are discussing on Stack Overflow [5], their end goal being to determine common challenges and misconceptions among web developers. On the RE front, Xiao et al.[6] compare three methods of requirements elicitation from software related Q&A sites, their goal being to determine which method is preferable.

Our research is notably different from these works because it directly addresses the *process* of Requirements Engineering rather than a small facet of it (like requirements elicitation), or the developer's perspective of a particular product or niche (like mobile development). We seek a holistic view of RE to better evaluate its strengths and its weakness through the eyes of the social Q&A sites.

III. Study Design

A. Research Questions

We formulate our study in the following two RQs:

- **RQ1:** *What are the main categories of topics of discussions about RE among practitioners?*

This RQ aims at grouping together common issues in RE (according to data from the Q&A sites) and finding any outstanding lessons that can be derived from those groups, to understand which parts of the RE process may be supported by further research.

- **RQ2:** *What types of questions are requirements engineers asking (e.g. What, why, how, who, others)?*

This RQ aims at better discerning the questions asked by practitioners of RE, and collecting them into broader understandings of current issues with RE, as another method of examining the process from the point-of-view of practitioners.

B. Data Collection

We queried the Stack Exchange Data Explorer (SEDE) to collect our data set. The Data Explorer is an open source tool for running arbitrary queries against public data from the Stack Exchange Network. The Stack Exchange Network sites we queried were Stack Overflow, Programmers Exchange and Project Management. We conducted 3 iterations of querying the network to refine our query and acquire our final data set for analysis. To collect data relevant to Requirements Engineering (RE) our logic for the query was to find posts containing the word "requirement" and at least one other word from a set of keywords related to RE. If the query picked up a relevant post it would return the entire discussion by collecting every post in that thread as well.

A post can be either a question or an answer, and certain properties are exclusive to questions, such as Titles and Tags. StackOverflow's database size is far larger than the other two sites; it was too large to impose our query's logic on the body of all posts. In order to prevent the query from timing out, we had to filter StackOverflow by the Title or Tags containing our main criteria "requirement", and then the body of the question containing one of the keywords. This meant we collected threads where only the question was found to be relevant. For the other two sites, it was feasible to look for "requirement" plus a keyword in the body of a post, which let us examine relevant answers as well. The set of keywords for this first iteration is indicated in Table II.

By manually analysing 1,000 posts from iteration #1, we discovered there were a great deal of irrelevant results collected. "Requirement" was a common word used in general discussion; two of the most popular topics not related to RE were about "job requirements" and "hardware requirements", which were returned despite being irrelevant because

TABLE I: Number of posts in our corpus over three iterations

Iteration	#(Stack Overflow)	#(Programmers Exchange)	#(Project Management)	#(Quora)
#1	3,204	12,930	2,731	158
#2	2.278	4,764	1,165	15
#3	1,315	2,462	776	25
Total	**4,553**			

"requirement" had appeared in conjunction with another of our common keywords. For the next iteration we decided the words "develop," "team," and "project" needed to be removed from the query because they had too many common usages. We also chose to exclude threads with the tags "job market", "interview", "hiring", "skills" and "career" to help eliminate the *job requirements* topic, and added words that were used in relevant posts so we would still collect them in the absence of the broad terms we removed. Thus, we added {evaluat, elicit, verif, client, use case, user stor[yi], estimat, specification, prototype, methodology, gathering, expcetation} to our query's set of keywords. The full set is indicated in Table II.

Our second iteration's data volume was scarcer but a lot more relevant (see Table I). We manually analysed 800 random results and found that this time they were generally related to Software Engineering, which - despite being an improvement - was still too broad. Our keyword set contained 20 words, so we decided to cut it down to words more exclusive to RE. Our keyword set for the final iteration is indicated in Table II. "Functional " intentionally has a space after it because we found that functionality was used more in context to coding questions. We chose to avoid "functionality" from being included as an extension of "functional." We also added {python, java, ruby, c[+#]} to our list of excluded tags because they were the most common language tags, and they were seldom used in an RE context. We also took "skills" out of our excluded tags, because it was not very common and could still be used in conjunction with RE posts.

A Note on Quora: The posts on Quora were initially included in our study, but we dropped them after a closer critique of Quora's questions. Data had to be collected manually from Quora's search function, as the site lacks the automatic querying that Stack Exchange's Data Explorer has. However, Quora's problem is its lack of meaningful data in both quality and quantity with regards to RE: closely related Topics like *Requirements Engineering* had only a small number of posts (i.e. 56) and larger ones like *Project Management* had too many to allow for manual collection (i.e. 8.6K). Of the few threads we could gather, only a small number had any text in their posts beyond titles, and fewer had any answers. We decided to iterate our results from Quora down by hand, keeping only relevant and highly-rated posts, but this gave us a relatively tiny number of posts that would not add to our data in any way. Therefore, although Quora is a Q&A site with some discussion of RE issues, we decided its content added too little to the volume of text already obtained from StackExchange to be meaningful.

TABLE II: Search terms used for retrieving data from Q&A websites

Iteration	Keywords
#1	engineer, analysis, stakehold, communicat, project, manag, develop, team, functional, scope, customer
#2	analysis, stakehold, communicat, manag, functional, scope, customer, evaluat, elicit, client, verif, use case, user stor[yi], estimat, specification, prototype, methodology, gather, expectation
#3	verif, validat, elicit, use case, user stor[yi], estimat, specification, prototyp, gather, captur, functional , document

C. Data Preparation

The data returned from Stack Exchange was not immediately ready for analysis. Stack Exchange returned the data from each query in CSV format and with HTML syntax. We had to convert each row into a text file in order to prepare the corpus for our algorithms. This copied the title of each column into each text file and there were a number of other formatting problems left over from the CSV file. We also approached our data cleaning process in an iterative manner. The first time we ran and analysed our topic modelling and word frequencies, we did minimal cleaning. Each step performed in the third and final iteration is described below as well as its evolution throughout our iterations.

Step 1 - *Convert Alphabetical Text to Lowercase:* We performed this step for all three iterations. We removed case sensitivity to ensure we did not analyse the capitalization of a word as a separate case than its lower-case counterpart.

Step 2 - *Remove HTML:* This step needed to be performed before punctuation was removed because angled brackets were indicators of HTML tags. This was noticed after the first iteration. For the second and final iterations, we removed everything between angled brackets. This removed the Tags from question posts as well.

Step 3 - *Manual Transformation:* We replaced words that were synonymous to each other since they would either appear too scattered due to their independence. Some of these words had to be united before the corpus was stemmed. For the second iteration, "end user" was replaced with "client." "customer" and "customers" were also replaced with "client" because they stemmed to "custom" alongside words like "customize," which is unrelated and inflated its count. For the final iteration, "functional " took on a unique stem to represent functional/non-functional requirements (F/NFR): "ffunction". This avoided words like "functioning" or "functionality" inflating the word count for F/NFRs. We have replaced "ffunction" with "Functional" in our results. We also concatenated "high level" to "highlevel" and "low level" to "lowlevel',' and replaced "class diagram" with "model."

Step 4 - *Removing numbers and punctuations:* We performed this step for all three iterations. This joined hyphenated words together, rather than separate them. For example, "non-functional" became "nonfunctional".

Step 5 - *Removing stopwords:* Stopwords are common words that provide no meaning on their own, such as "the". Here we used the default set of stopwords in the **tm_map** package for **R**.
Step 6 - *Strip Whitespace:* We removed excessive whitespace such as newlines, double spaces and tabs.
Step 7 - *Stemming:* We performed this step for all 3 iterations. Stemming is the process of reducing words to their origins by removing suffixes. For example, "functionality", "functioning" and "functional" would all become "function". However, as we mentioned above for this specific case, we wanted to separate "functional" (as in functional requirements) from other forms of 'function", which is why we used our makeshift word"ffunction".
Step 8 - *More Manual Transformation:* In the second iteration, "build" and "creat" were replaced with "implement". "user stori" and "use case" were replaced with "model." For the final iteration, we replaced "nonfunct" with our word "ffunction." This was done after stemming for convenience and before removing our additional words.
Step 9 - *Remove Additional Words:* For the second iteration we deleted an additional 60 words. Most of these provided no meaning in any context and were not included in the basic set. "requir" was also removed, since - at this point - all of the words are relevant to requirements engineering. For the final iteration our additional set contained an additional 113 removed words, most of which were common words that provided no meaning in an RE context.

IV. Data Analysis and Evaluation

A. Data Analysis

1) Topic Modeling: Topic modeling is an unsupervised text analysis techinique that groups a small number of highly correlated words, over a large volume of unlabelled text [7], [8], into *topics*.

During the process, we analysed 4,553 posts (obtained in the previous step) using the Latent Dirichlet Allocation (LDA) algorithm into models of three to seven topics. A topic, as defined in the LDA approach, is a probability distribution over a vocabulary [8], for which the **topicmodels** package in **R** was needed to correctly implement the LDA algorithm. This procedure also required the Gibbs sampling option, as it provides greater accuracy than the variational algorithm [9]. The algorithm we used is detailed in Algorithm 1. In this algorithm, we first defined the requisite variables, assigned a random value to them, and then looped through it a specific number of times. Each loop assigned a word instance of our corpus to a sample topic. We defined the initial value of the variables used in this algorithm as follows:

- $n_{d,k}$ (the number of words assigned to topic k = 6
- k (number of topics) = (3..7), $iteration$ = 200
- $nstart$ (defining different starting points for the sampling purpose) = 5
- As we started the algorithm randomly, it is necessary to discard the initial iterations. To this end, we discarded the first 400 iterations, which is called the *burn-in period*.
- $thin$ = 50. This parameter is used to reduce the correlation between samples (e.g., over 200 iteration, we took every 50^{th} iteration for further analysis)

In this algorithm, z represents the topic assignment for each of the N words w in our data set.

Input: words $w \in$ documents d, $nstart$, $burn-in$, $thin$, $n_{d,k}$
Output: topic assignments z
begin
  randomly initialize z and increment counters
  foreach *iteration* **do**
    foreach *word* w **do**
      foreach *topic* k **do**
        $\theta_{d_w,k}$ = *calculating the document/topic distribution for topic* k, *word* w *in document* d
      end
      $topic \leftarrow$ sample from $multnomial(\theta_{d_w})$
      $z[w] \leftarrow topic$
      update counts according to new assignments
    end
  end
  return z
end

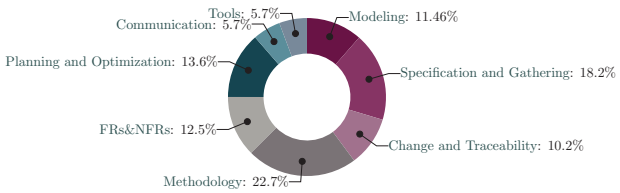
Algorithm 1: LDA Gibbs Algorithm [10] for exploring the most popular RE topics on Q&A sites

2) Statistical Analysis: In this section we examined a sample of 150 titles from the threads we collected, and recorded the context of the question based on the criteria outlined in Table IV. Titles were ignored when they did not provide enough context.

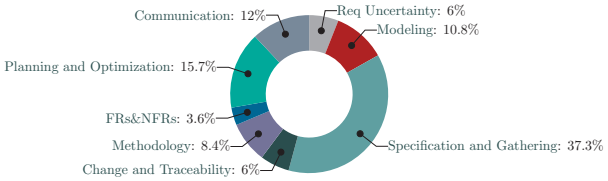
- Question type: The question types were not mutually exclusive. For example a question could be asking How to do something during a specific life-cycle stage (When). 12% of the questions were concerned with Who: who is in charge of an aspect of the product, or how to deal with a stakeholder. 10% were concerned with Why, asking about the purpose of certain practices. 59% were What questions, which asked for more information on one of the topics. 23% of the question had When criteria and were concerned with a certain specific stage of development. Finally 56% of the questions met the How criteria and were asking how to solve a problem.

What type of questions should we be asking? are these people asking the right questions?

Is this what you expect as the most frequent type of question asked and why



(a) Details of the questions of type "What"



(b) Details of the questions of type "How"

Fig. 1: Proportion of RE-related concepts addressed in different question types

TABLE III: Topic Modeling Results (**K** represents the number of topics)

K=3		
Topic#1	**Topic#2**	**Topic#3**
model	implement	test
document	function	time
client	validate	team
design	new	change
specify	method	product
want	type	estimate

(a)

K=4			
Topic#1	**Topic#2**	**Topic#3**	**Topic#4**
test	implement	model	client
make	validate	document	team
problem	new	implement	product
time	type	design	process
want	service	specif	manag
change	method	functional	estimate

(b)

K=5				
Topic#1	**Topic#2**	**Topic#3**	**Topic#4**	**Topic#5**
client	test	implement	make	team
model	document	new	time	product
want	design	valid	know	process
implement	specify	service	change	manage
business	write	database	good	estimate
problem	functional	type	better	agile

(c)

K=6					
Topic#1	**Topic#2**	**Topic#3**	**Topic#4**	**Topic#5**	**Topic#6**
implement	document	client	test	make	team
type	implement	estimate	model	want	product
valid	tool	time	design	good	process
new	inform	task	ffunction	change	manage
service	version	cost	write	know	agile
database	control	possible	specify	problem	business

(d)

K=7						
Topic#1	**Topic#2**	**Topic#3**	**Topic#4**	**Topic#5**	**Topic#6**	**Topic#7**
implement	time	team	model	client	test	design
new	know	product	document	manage	change	ffunction
service	make	estimate	specify	business	write	function
database	good	agile	detail	change	implement	implement
valid	problem	task	implement	want	spec	differ
return	start	sprint	tool	process	bug	support

(e)

- Domain: 67% of the questions belonged to the problem domain. A majority of the titles analysed belonged to this domain and addressed stakeholders, management, and more conceptual concerns like *"How do you manage customers with regards to changing requirements?"* 11% of the questions belonged in the solution domain. Most of questions were about achieving tangible results from requirements specifications, or maintaining development through requirements change such as *"How did you adapt your unit tests to deal with changing requirements?"* 22% of the questions encompassed both domains, and most of these questions were about eliminating uncertainty and translating stakeholder defined requirements into tangible development *"Format for getting clear directions on data parameters from users"*.

- Product: These question addressed criteria related to working software or business prospects. 48% of the titles addressed one of these aspects. The most popular topic among these was functionalities at 46% . Functianlity questions were concerned with translating requirements into working code and coding for the final product, An example would be *"Is writing software in the absence of requirements a skill to possess or a situation I should avoid?"*. 21% of the posts were concerned with release, these posts had to to with meeting and tracking estimations, and obtaining a shippable product. e.g. *"Why is a software development life-cycle so inefficient?"*. 12% of the posts had financial concerns, these were concerned with the development efficiency and the financial success of software. e.g. *"What types of requirements add the most value?"*. 11% of posts were concerned with improving relationships and overall client satisfaction, e.g. *"What specifications in software development are relevant to the clients?"*. Finally 12% were related to quality of the product, these posts were about internal code quality and improving coding practices. e.g. *"Iteratively improve software architecture & quality in an agile process?"*.

How much of RE should be aiming to get the most financial value vs to what point should start prioritizing other areas such as quality, unit test, meeting deadlines, client satifaction etc.

TABLE IV: RQ2- Question Types

-What, Why, and How?
- Requirements {communication, Change, Relationships, Uncertainty, Prioritization, Planning, Modelling, Traceability, Specification, Estimation, Gathering}, FRs/NFRs, and tools and applications for RE.

-When?
- Requirements {Change, Validation, Modelling, Prioritization, Implementation}

-Where?
- Distributed requirements engineering

-Who?
- *Users:* who will actually operate and interact with the system
- *Developers:* who design, build, and maintain the system, such as analyst, designer, programmer, tester, and so on
- *Decision-makers:* who have executive power and control over projects decisions to build the system, such as the managers of the development and users organizations, business, and product managers.
- *Clients:* who will pay for the system

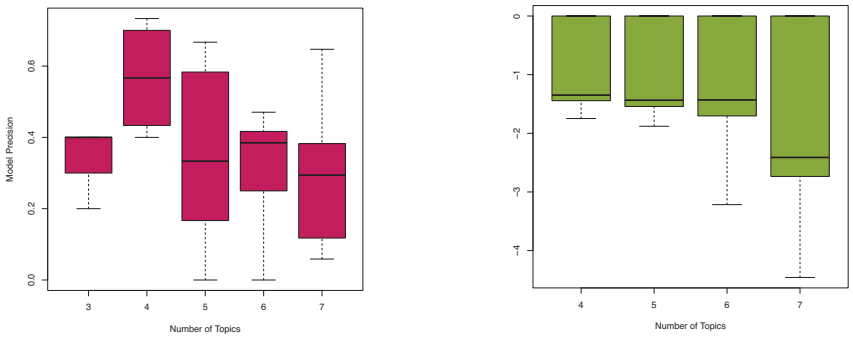
-RE Domain
- *Problem:* requirements that represent the stakeholders' needs and expectations
- *Solution:* requirements that represent the subsequent layers (e.g. system requirements)

-Product
- Functionalities, quality, customer satisfaction, financial, release management

What factor is most prioritized and do u agree with that?

B. Evaluation

The problem with Topic Modelling specifically for RE questions on Q&A sites is the lack of previous data, without which it is difficult to critique our models. So, in an attempt to judge the merit of our explored topic models, we used **word intrusion** and **topic intrusion**, two quantitative evaluation techniques proposed by Chang et al. [11], the process and results of which we will detail for the remainder of this section.



(a) Word Intrusion (b) Topic Intrusion

Fig. 2: Evaluation of the precision and relevance of the topic modeling results

1) Word intrusion: This method measures the quality of the inferred topics by calculating their "cohesiveness". To measure the coherence of the explored topics in our study, we followed all the steps proposed by Chang et al., as follows:

Step 1: As illustrated in Table III (a-e), we first selected the six most probable words from each topic. We then added a seventh, random word - with low probability in the current topic - from a list of words outputted by the LDA algorithm. Those seven words were displayed in a random order to participants, who were asked to select the word that did not fit with others in the topic. As the new word was selected from a list of low-probability words, we also reduced the probability that this intruder semantically belongs to the other words of each topic.

Step 2: Next, we measured the *model precision* for different values of *k* by calculating the degree to which the participants had recognised a unified concept behind the topic. For this purpose we used the following equation proposed in [11]:

$$MP_k^m = \Big(\sum_s \mathbb{1}(i_{k,s} = \omega_k) \Big) / S \tag{1}$$

When MP_k represents the model precision of model m for a specific value of k. Further, ω_k denotes the index of the intruding word for topic k, $i_{k,s}$ represents the intruder selected by s^{th} subject from a set of words listed in topic k, and S denotes the number of subjects.

2) Topic intrusion: This method measures whether the contents of a document align with the topics it has been assigned to, according to human judgement.

Step 1: The three most relevant topics - with the highest probability according the LDA algorithm - were first isolated into a group. Then, an *intruder topic* was added to this group. An excerpt was then selected for each document which the participants read, and from which they had to deduce the intruder topic from the group of four (which was displayed in a random order).

Step 2: To quantify the degree to which our explored topic models agreed with human judgement, we used the *Topic Log Odds (TLO)* [11] parameter, which utilizes the following equation 2 :

$$TLO_d = \Big(\sum_s \log \hat{\theta}_{d,j_{d,*}} - \log \hat{\theta}_{d,j_{d,s}} \Big) / S \tag{2}$$

When θ_d represents the probability that document d belongs to each topic, $j_{d,s} \in \{1...K\}$ represents the intruding topic detected by participant s for document d, and $j_{d,*}$ denotes the *true* intruder.

Each participant conducted a total of 12 or 13 tasks, with 8 or 9 tasks completed for word intrusion and 5 for topic intrusion. For word intrusion, the topics were split up into three groups of 8:

- *Group 1*: 3 topics from K=3, 4 topics from K=4, 1 topic from K=5
- *Group 2*: 4 topics from K=5, 4 topics from K=6
- *Group 3*: 2 topics from K=5, 6 topics from K=7

This left one topic from K=7 unused, which we put in Group 3 for a total of 9 topics. 5 topic intrusion tasks were then added to each Group. 15 participants did the 12 tasks in Group 1; 12 participants did the 12 tasks in Group 2; 17 participants did the 13 tasks in Group 3. Table V lists the statistical details of the participants and their assigned tasks. Figure 2 represents boxplots of the precision for various values of K (i.e. 3,4,5,6, and 7).

K=3 is exempted from the topic intrusion boxplot because the intruder topic was too easily identifiable: 80% of responses recognized the intruder when K=3, whereas in word intrusion, only 33% recognised the intruder. This fault in the topic intrusion process gave a heavy bias towards K=3, so we have decided to exclude it. Nonetheless, the figures indicate that with a larger number of topics, their meaning becomes less coherent to the reader - one reason could be because the topics become too niche to be consistently associated with a large set of documents. With this in mind, we have concluded that having **4 topics** is most appropriate for further analysis, as the intruder is more consistently identified than in other values of K; implying that these 4 topics are the most coherent and applicable topics with regards to our data.

TABLE V: Qualitative and quantitative information about the projects included in our study

Statistical information			
K	**# Participant responses (WI)**	**# Documents**	**# Participant responses (TI)**
3	45	44	44
4	60	44	44
5	63	44	44
6	82	44	44
7	119	44	44
total	369	220	220

TABLE VI: The main activities of RE process

RE Activities [12]–[14]
-Elicitation: Identifying the users and project sponsors needs and expectations, projects stakeholders, sources of requirements, and understanding the application domain
-Modelling: Representing the RE process as well as a whole range of RE artifacts. Enterprise, data, behavioural, and domain modelling are some general categories of RE modelling.
-Communication: Circulating the identified requirements among project's stakeholders.
-Validation and Verification: Evaluating the completeness and correctness of the elicited requirements and artifacts.
-Evolution: Managing and monitoring requirements changes (e.g. adding, deleting, or modifying requirements)

V. Results and Findings

A. RQ1: the main categories of discussion topics about RE

Knowing that our explored topic models when K=4 were the most cohesive, this section presents our interpretations of what exactly those topics mean, providing an RE-centred understanding of these topics with reference to Table VI.

A significant portion of RE-related questions on Q&A sites can be grouped into more specific scenarios in RE Activities and interactions between Stakeholders and Developers. The breadth of the questions can be classified as *"Who has RE-related problems?"* versus *"Which part of the RE phase does this problem most relate to?"*

This section gives the rationale behind our results in Table VII and greater depth on each topic. Our interpretations are then defended using examples from our data set.

> **Finding 1.1**: One significant topic of interest lies in developers desiring to better understand their clients' requirements, as well as increase their client's understanding of the development process, through increased communication.

Topic 1 - Test, Make, Problem, Time, Want, Change: We determined Topic 1 to be related to **requirements validation**, a process that includes checking requirements to see if they satisfy a client's demands and making changes as necessary. However, validation alone does not completely cover the idea behind the topic; instead, it is one way to *avoid* the common RE pitfall of scope creep, which is a much closer approximation of the topic's idea. More specifically, this topic focuses on the the developers' limitations (represented in the topic-words Test, Make, Problem and Time) against the clients' interests (represented by Want and Change). This fault line between the two groups is especially apparent in cases where the client seeks to add or change requirements from their initial proposal, which is discussed to a greater length in Topics 2 and 4. Due to the implied focus on interaction between developers and clients, **requirements communication** is in fact the most important part of this topic. Requirements validation, therefore, is an RE activity that is *facilitated* by requirements communication.

An example of such a concern being discussed is in a post at Programmer's Exchange[5]. The original question focuses on project estimation when the user demands new requirements, which falls under Topic 4, but the response focuses on developing a high level of communication between the client and the development team and thus avoiding a situation where the customer is always insistent on new requirements.

even when user demand new requirements, developers can still focus on the communication to avoid customer being insistent on these new requirements

> **Finding 1.2**: Questions that directly asked for input from the Q&A community were mostly about technical details on implementing the project and checking whether the system works as desired.

Topic 2 - Implement, Validate, New, Type, Service, Method: Topic 2 seems most appropriate when tied with **requirements validation and verification** (V&V). Validate as a keyword clearly vouches for this. Implement, Service and Method are also relevant words, as they hint at the architecture behind the system: whether it is being implemented using the right method, for the right service (i.e. V&V), or if it is being implemented at all. This is in contrast to Topic 1, where requirements validation was only tangentially related to the key focus of that topic.

Many posts in Topic 2 focused on implementing the architecture of the system rather than eliciting or testing them directly. Many questions were about technical details of a requirement that took full advantage of the social aspect of the Q&A sites, asking for direct input or software help on their projects from other members of the community. These points were reflected in a post on[6], the author of which has a set of requirements in mind for his project but is unsure of how to implement them as required. The question focuses not on a discussion of RE, but rather the technicalities and processes behind actually validating the author's method - and a follow-up question asking the community directly what they would do instead.

> **Finding 1.3**: A common desire among developers is to better track, document, organize and present their requirements. This is summed up as requirements evolution, and this topic consists entirely of addressing those issues using requirements modelling.

Topic 3 - Model, Document, Implement, Design, Specification, Functional: This topic goes through keywords from

[5]programmers.stackexchange.com/questions/63428
[6]programmers.com/questions/220199

every aspect of RE, illustrating a large focus on **requirements evolution** over the course of the entire RE phase rather than a smaller part that can stand alone. Documenting and tracking changes in requirements over time are an important part of software development, and this is best facilitated by **requirements modelling** which focuses on presenting and visualizing data in a comprehensible manner to the development team. The words Functional and Specification, for instance, are not intended for clients or end-users, but rather for developers to dichotomize their customer's demands to a level that it is easy to implement. As such, the topic is focused less on the client or the product, and more on effective development practices that developers apply to build their product's requirements.

For example, the author of a post on Project Management[7], sorted as Topic 3, questions and critiques his team's documentation process of putting together "large Microsoft word documents" and asks for feedback from the community on a Wiki instead. At StackOverflow, another document from Topic 3[8] asks for effective tools "to aid in initial project design," defining and tracking requirements over time and collaborating with "the team when putting together the specifications." Other examples asked for resources and methods to document and track non-functional requirements as well.

> **Finding 1.4**: Often, a development team is rushed to release a product for a deadline from a years-old project estimation. Requirements engineers are concerned about how to avoid the crunch through better estimates, better customer relationships, and a healthier development process through communication between clients, developers and executives.

Topic 4 - Client, Team, Product, Process, Manage, Estimate: We found this topic to be most directly related to **requirements communication**, specifically regarding communication between clients, developers, and decision-makers. Managing requirements (from Manage and Process, which reflect concepts from Topic 3) might indicate some form of requirements modelling, but other words show a clear focus on delivering an end product. Consider the words Estimate, which inherently suggests a deadline by which the product should be complete; Product, again implying an end product for delivery; and Client, to whom the product must be delivered. In this context, Process, Manage and Team are keywords related to developing, implementing and building the software's requirements, rather than gathering or changing them, and are thus not strictly related to RE. However, it is a direct consequence of the RE phase, which is why this topic is still important.

A document assigned to Topic 4 that reinforces this interpretation comes from Programmers' Exchange[9] as a response to a poster frustrated with project managers (PMs) who misuse estimates. The post vouches for PMs who are capable of communicating with developers. On another note, a Project

[7]pm.stackexchange.com/questions/9555
[8]stackoverflow.com/questions/4187927
[9]programmers.stackexchange.com/questions/162499

TABLE VII: Lists of Stakeholders and Requirements Activities most involved with each Topic

Topic	Requirements Activities	Involved Stakeholders
#1	Validation; Communication	Developers; Clients
#2	Validation; Verification	Developers
#3	Evolution; Modelling	Developers
#4	Communication	Developers; Clients; Decision-Makers

TABLE VIII: Quantitative data on each topics' popularity against other topics

Topic	Percent of total questions	Average probability from all posts (%)	Average probability when most probable (%)
#1	24.23	25.47	35.65
#2	21.50	23.67	40.22
#3	25.76	25.13	35.55
#4	28.51	25.73	38.02

Management question[10] details a situation where a lack of communication has stalled and "demoralize[d] the team," due to the lack of a Product Owner.

> **Finding 2**: No specific topic significantly outweighed the others in popularity. However, rather than having clean divisions between the 4 topics, their interests often overlapped, reducing the probability that they can be 'solved' in a simple or elegant manner.

This section gives an analysis of Table VIII, which summarizes 3 sets of data for the 4 topics.

TODO checkpoint

'Percent of total questions' shows what portion of the data set was sorted to each topic, which is trivial to analyse: they are all in close proximity to each other, with the greatest difference being of 8.01% - a relatively small amount. 'Average probability from all posts' is the mean probability of each topic over all 4,553 posts, which once again shows that all topics are almost equally popular on Q&A sites.

The last column is the average probability of a topic *n* over all posts where topic *n* has the highest probability. What stands out is how low the numbers are, despite being a mean of all of the highest values for each post: not a single value is over 50%. This means that, while the posts themselves are near-equally distributed across the 4 topics, the distribution of those topics inside those posts is not so black-and-white. No topic is consistently discussed alone; often ideas and keywords from other topics are used frequently - almost equally, in some cases. Therefore, the belief that these 4 topics give a sound and complete overview of the issues in RE must be disabused. While they do reflect ideas in RE that may need to be revised based on further research and evaluation, they are simply a guide to understanding a practitioner's view of those ideas rather than presenting them as a concrete problem to 'fix' or 'solve'.

[10]pm.stackexchange.com/questions/9985

B. RQ2: Question Types

The largest portion of What questions, at 22.7%, is concerned with methodologies. These were two types of questions: One asked how to do one of the other activities with respect to their methodology, and the other asked which methodology is best for their project. Both types show that a large portion of uncertainty comes from attempts to adopt and follow the correct methodology. Both types show that a large portion of uncertainty comes from attempts to adopt and follow a 'correct' methodology. Additionally, 8.42% of the How questions were about methodology. These were questions about how to streamline or properly apply the methodology that has been adopted. Note that Agile was our 25th most frequent word with 1,328 uses and even appeared in our topic modelling for K= 5, 6, and 7. Agile methodologies are popular, leading to greater uncertainty about it. This may stem from 'Agile' being incorrectly adopted by those unfamiliar with its principles.

Specification and Gathering was our most popular How topic at 37.37% and our second most popular What topic at 18.2%. Along with the finding that 46% of posts related with their product were concerned with functionalities, this suggests that requirements gathering needs to translate into well defined system requirements. A perfect example of this would be the question *"What is the ideal mindset for a developer participating in a requirements gathering meeting?"*. This shows that there is a gap between how stakeholders describe a system and what a developer needs to know in order to create the system. It can be interpreted that questions about modelling also relate to this problem of better defining a blueprint of the system. *Implement*, *model* and *document* were our 3rd, 5th and 7th most frequent words respectively.

However, "Client Satisfaction" had low popularity at 11% of product related posts. The contrast between the lack of concern for client satisfaction, with the popularity of Specification and Gathering, Modelling and Functionality concerns suggests that developers are more focused on building the system right, rather than building the right system.

Another considerable portion of our results fall into the Planning and Optimization category. This is highlighted by 'release' being the second most popular product aspect at 21% of product type questions. Only once did 'planning and estimation' have an instance of being ahead of schedule, and it was in regards to an inefficient iterative sprint schedule where the developers were waiting for requirements of the next version to be defined. This highlights the importance of planning the appropriate order of implementation. The goal is to reach tangible milestones to satisfy the stakeholders with the progress of the product, while the work done has also been relevant to the next iteration at hand. The other side of planning is optimizing estimations. Estimate was our 16th most common word with 1811 uses.The most common factors for poor estimations come from pressure and pride. Developers did not want to disappoint their stakeholders by giving lengthy estimations or dispute deadlines. Developers also tend to overestimate their capabilities, despite giving some overhead. It resembles the self-referential adage of Hofstadter's Law: *It always takes longer than you expect, even when you take into account Hofstadter's Law.*

Finding 3.1: Often developers and stakeholders do not thoroughly understand the methodology they have chosen to build their product.
Finding 3.2: Developers are focused on building the system right and want requirements translated into well defined system functionalities.
Finding 3.3: Project planning should emphasize an appropriate sequence of development milestones, and developers should not be discouraged from giving and justifying substantial estimations.

VI. Threats to Validity

The points on why Quora was dropped for analysis and why the Topic Intrusion results for K=3 were not shown are discussed in *Data Collection* and *Evaluation* respectively, so we will discuss threats that came from the rest of our process.

Many of the questions on the Stack Exchange sites are considered "closed" by the communities they are part of. That is to say, they are part of the site's archived data, but may no longer be interacted with by the community. However, there was no way to mitigate the selection of these posts in SEDE, so we decided to include them in our study. This calls into question if our study has been a fair evaluation of social Q&A sites, and whether the discussion archived in a "closed" post can be considered relevant to contemporary RE issues.

Additionally, the authors of posts on StackExchange may or may not be professional practitioners of RE. While the large quantity of anecdotal posts that discuss RE issues suggests that the majority of posts are written by such practitioners, StackExchange is an open community where people of any background may post recommendations. This affects our study because any number of questions may be asked by students and beginners rather than practitioners. Once again, we found no way to mitigate this without some method of manual selection.

This leads to the final point: we have chosen not to go through our corpus manually to best keep the effect of our Topic Modelling process. However, this meant that posts that were only tangentially related to RE were allowed into our study, which may have inflicted a bias on our final results. On a more extreme level, some posts may not have been related to RE at all, but may have entered our corpus from using our keywords during the Data Collection process. In this case, we did find an effective way to reduce the irrelevance of the posts selected through the iterative collecting process, which narrowed the scope of our data significantly, but a more extensive filtering process would be welcome.

VII. Conclusions and Future Work

Requirements Engineering activities are a key part to developing a functional and satisfying product for all stakeholders

involved. However, finding what those stakeholders desire out of the RE process can be long and difficult. Using social Q&A sites, we tried to simplify that process by looking at the most popular RE-related topics of discussion amongst practitioners. We found those topics to focus on (1) a dissatisfaction with the level of communication and interaction between customers and developers; (2) seeking direct help and input with implementing functional requirements for a product; (3) tracking additions and changes to requirements, and organizing them; (4) greater communication between executives, clients and developers in order to avoid a tough deadline or bad management.

Furthermore, (5) none of these 4 sections have any great precedence over another; however, they are not so distinct from each other as to be mutually exclusive. By looking directly at the text of these posts, we also deduced that (6) both developers and stakeholders seldom fully comprehend the methodology they have chosen to develop their product with, leading to less work done overall; (7) developers seek an easy way to transform their clients' requirements into well-defined system functionalities; (8) project planning should motivate the team and the customers with progress, not cause them to overwork.

With regards to future research, we intend to develop a tool that enhances communication between a project's stakeholders, focusing on tracking requirements-changes and trade-offs (i.e. requirements debt) [15] over time through novel forms of requirements modelling. Other research can be focused on manual data collection, allowing for more specific data and perhaps even the inclusion of a variety of social Q&A sites like Quora that we could not include in our study. The opposite would be to develop a tool for the automated analysis of Q&A sites, allowing widespread use of this process even outside of academic use - this is also another tool we would like to develop. Another step could be taken by conducting a parallel study with academic papers rather than Q&A sites, and comparing results from the two, or even social media sites such as LinkedIn or Facebook groups.

ACKNOWLEDGMENT

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada, NSERC Discovery Grant 250343-12. We also thank all participants of our survey for volunteering and sparing their time.

REFERENCES

[1] A. Aurum and C. Wohlin, "A value-based approach in requirements engineering: explaining some of the fundamental concepts," in *Requirements engineering: foundation for software quality*, Springer, 2007, pp. 109–115.

[2] "Implementing requirements engineering processes throughout organizations: success factors and challenges," *Information and Software Technology*, vol. 46, no. 14, pp. 937 –953, 2004.

[3] A. Barua, S. W. Thomas, and A. E. Hassan, "What are developers talking about? an analysis of topics and trends in stack overflow," *Empirical Software Engineering*, vol. 19, no. 3, pp. 619–654, 2014.

[4] C. Rosen and E. Shihab, "What are mobile developers asking about? a large scale study using stack overflow," *Empirical Software Engineering*, vol. 21, no. 3, pp. 1192–1223, 2016.

[5] K. Bajaj, K. Pattabiraman, and A. Mesbah, "Mining questions asked by web developers," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ACM, 2014, pp. 112–121.

[6] M. Xiao, G. Yin, T. Wang, C. Yang, and M. Chen, "Requirement acquisition from social q&a sites," in *Requirements Engineering in the Big Data Era*, Springer, 2015, pp. 64–74.

[7] H. M. Wallach, "Topic Modeling: Beyond Bag-of-words," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, ACM, 2006, pp. 977–984.

[8] D. Blei, L. Carin, and D. Dunson, "Probabilistic Topic Models," *Signal Processing Magazine, IEEE*, vol. 27, no. 6, pp. 55–65, 2010.

[9] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling, "Fast collapsed gibbs sampling for latent dirichlet allocation," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '08, ACM, 2008, pp. 569–577.

[10] W. M. Darling, "A theoretical and practical implementation tutorial on topic modeling and gibbs sampling," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 642–647.

[11] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei, "Reading tea leaves: how humans interpret topic models," in *Advances in neural information processing systems*, 2009, pp. 288–296.

[12] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap," in *Proceedings of the Conference on the Future of Software Engineering*, ser. ICSE '00, ACM, 2000, pp. 35–46.

[13] D. Zowghi and C. Coulin, "Requirements elicitation: a survey of techniques, approaches, and tools," in *Engineering and managing software requirements*, Springer, 2005, pp. 19–46.

[14] Z. Shakeri Hossein Abad, M. Noaeen, and G. Ruhe, "Requirements engineering visualization: a systematic literature review," in *2016 IEEE 24rd International Requirements Engineering Conference (RE)*, 2016.

[15] Z. Shakeri Hossein Abad and G. Ruhe, "Using Real Options to Manage Technical Debt in Requirements Engineering," in *Requirements Engineering Conference (RE), 2015 IEEE 23rd International*, 2015, pp. 230–235.