EXPERIENCES IN INTEGRATING SOCIAL LISTENING WITH THE QAW

Heeran Youn, Seungho Nam, Eugene Kim, Jiyeon Lee, Heekyeong Bae, Sunmyung Lee, Dongjin Lee
Samsung Electronics
Suwon, Kyunggi-do, Korea (ROK)
{heeran.youn,seungho.nam,ivey.kim, jyeon0.lee, special7, smyung, dongjlee}@samsung.com

ABSTRACT
This paper proposes a novel method to graft “social listening” technique to Quality Attribute Workshop (QAW). QAW is a team facilitation meeting among stakeholders, which was introduced by Software Engineering Institute (SEI). It helps them communicate each other, thereby exposing interactions of quality attributes, and finding new quality attributes which development team may not have surfaced or considered before. Therefore, having all stakeholders in the meeting is very critical in order to cover their quality attribute concerns. In practices, however, inviting real end users to QAW is almost impossible during development phase, even though they are the most important stakeholders whom development them should consider. So we suggest “social listening”, which elicits end users’ voices or thoughts about a product from SNS, such as Twitter, Facebook, and blogs to compensate end user’s absence in QAW.

KEY WORDS
Quality attribute, quality attribute workshop, QAW Social Listening, Social Network Service, SNS, twitter

1. Introduction

As the size and the complexity of software in embedded systems grow exponentially [11], designing the right software architecture becomes one of key success factors in Consumer Electronics (CE) industry [8], [9]. To design a very solid architecture for our product, we reviewed many techniques around architecture design [1-9], [17] and have selectively chosen QAW [10] and ATAM [11] (Architecture Tradeoffs Analysis Method).

For the last 10 years, we have conducted QAW and ATAM in order to identify a system’s architecture and critical quality attributes and to evaluate the architecture. QAW is a very systematic way to translate business contexts into quality attributes, as well as to ensure that all quality attributes are included in the final design. QAW is an interactive workshop with main stakeholders of a project; therefore, important stakeholders’ absence could lead to omission of important quality attributes. Since our team’s products are consumer electronics such as mobile phone, digital TV, and refrigerator, end users are the most important stakeholders whom development teams should care for the most. In practices, however, having them in QAW is almost impossible. Usually marketing representatives or product planning representatives try to play their roles by guessing end users’ expectations but it often turns out that we misunderstood users’ voices after the product’s release.

We started finding solutions to mitigate this weakness and turned our attention to a recent boom of SNS such as Twitter, and Facebook which allow people, most of end users to write their opinions or thoughts about products, companies, or brands instantly. These services also provide huge amount of data that is gathered from their users available to us by supporting developer APIs. To strengthen our QAW, we have developed in-house “social analysis system” named Find Insights by Data (FIND) to compensate absence of real end users in our QAW through peeping social network data. We designed our system to collect Twitter mentions, user reviews or comments about our and competitors’ products via twitter streaming API [20] and cralwer4j, by analyzing morpheme, related words, and their counts, and visualizing these into graphics. This paper only presents Twitter analysis part of FIND system.

This paper is comprised of six sections including this introductory section. Section 2 introduces SEI’s QAW and what we found while applying it to our organization briefly. In Section 3, we will describe our social analysis system, FIND. How we applied our new approach is written in Section 4. Section 5 has a discussion on our experiment’s result and pros and cons of our approach. We conclude this paper in section 6 with some future plans.

2. Introducing QAW

SEI defines QAW as follows:

*The QAW is a facilitated method that engages system stakeholders early in the system development life cycle to discover the driving quality attributes of a software-intensive system.* [10]
Unlike traditional system development ways, QAW was introduced to prevent quality attributes from being missing, vaguely understood, or weakly articulated. Figure 1 describes QAW’s steps.

QAW itself is a very helpful method to find out the critical quality attributes and communicate the quality attributes very clearly with efficiency. Detailed steps and the process of QAW are written in SEI’s Quality Attribute Workshop, Third Edition. [10]

While we were trying to adopt this QAW in our organization, we found several shortages of it and made some improvements on following points: 1) Making a well-gelled team, 2) Making QAW fun – Introducing gamification, 3) Visualizing Outputs, 4) Prioritizing the scenarios, 5) Providing tools for making QAW easy and fun, and 6) Developing quality attribute evaluation criteria. On top of these improvements, we developed our “social analysis system” FIND to mitigate the risk of not hearing from the end users.

3. Introducing Social Listening to QAW

For last many years, we strived to improve our QAW to design better architecture by making a continual progress in points mentioned in the previous section. However, while the product is in development, we found that it is extremely difficult to have the end users in QAW and accommodate the real end users’ requirements. To overcome this weakness and enhance our QAW, we decided to analyze the social network data. Our improved version of QAW method is comprised of following phases and steps below.

In this section we intended to provide general idea of our process not detailed process and outputs; therefore, we do not provide detailed description of them. Figure 2 describes the overall process of our tailored QAW with social listening technique.
of QA’s on the board. After the participants make an agreement, QAW facilitator marks the target quality level in utility tree.

**Step 14. Business Goal Translation Table Composition**

To give a brief and clear traceability from business goal to QA’s and to architectural elements later, we adopted a table displaying this traceability. This table not only serves as a communication tool for the workshop audience but also gives a very clear understanding of how business/mission strategies are implemented as QA’s and architectural elements.

### 4. Social Analysis System

To strengthen our QAW with more accurate feedback from real end users, we turned our attention to social network and customer voices on the web to build a social analysis system, FIND. As a team of four, we designed our system using java language, MySQL database, and PHP. To gather that data, we wrote crawler applications to grab texts on the web and tweets. As a nightly batch, the analyzer application parses the text, finds out the keywords we want and analyzes the text to store the analysis result in the database. To promote the accessibility to the analysis result, we built an intranet website to allow users to run some queries and to see the result visually. We built our system to analyze dozens of news websites, Twitter, our organization’s idea pool, and academic papers. This paper is focused on describing only Twitter analysis part which is used for our experimental purpose.

Figure 3 represents the static view of Twitter analysis part’s architecture of our social analysis, FIND. Twitter Analysis System subsystem represents Twitter analysis part of FIND. TweetViewer layer is a UI layer that allows users to view the analysis result on a web browser. It contains PHP pages and javascript files. TweetAnalyzer layer is a layer that analyzes the tweets to figure out how many keywords showed up, how many related words showed up, if the sentences with keywords are positive or negative, and so on. TweetCollector is a layer which contains classes for collecting tweets through twitter4j and Twitter API’s.
Figure 4 describes the dynamic view of Twitter analysis part’s architecture. TwitterCollector component is an executable process, which uses DataAccessor component to access database. It collects tweets via Twitter API’s and stores tweets as files. TweetAnalyzer component is an executable process which reads files that TwitterCollector stores, and analyzes the files to write result to database using DataAccessor. TweetViewer component is a PHP process which queries database via DataAccessor to display analysis result on a web browser. DataAccessor is a component which allows other component to access database. Data access components for Java and PHP are represented as DataAccessor for brevity.

Step 3. Keyword Selection
As a team, we decided to choose product names around smart watch as our keywords. Since this product is in its very early stage, it does not have a general noun for it unlike smartphone or it does not have any specific technology in fashion such as LTE for smartphone or curved display for TV; therefore, we chose “galaxy gear”, “sony smartwatch”, and “iwatch.”

Step 4. Social Data Analysis
For about ten days, we collected tweets on three smart watch products and elicited QA’s. We analyzed the tweets in the following way: 1) divide the tweets into sentences, 2) look for the keywords we set in the sentence, 3) increase the keyword count, if the keyword is found, 4) look for the sentimental words from the sentence; we have a pre-determined dictionary of sentimental words with their points, 5) increase positive point or negative point, if sentimental words are found, 6) break down the sentence with keywords into words and keep counts of them. After this analysis, we generated a table for each keyword with related words – words that showed up in a sentence with the keyword. It also provides the total count number, the count number in positive sentence, and the count number in negative sentence.

With this table, we looked through the most frequent words with the keyword. We believed this would be efficient to figure out what people are interested most during the period of time. However, we were not sure if tweets are favorable to the product or unfavorable.

To increase readability, we sorted the list by count number in positive sentence and looked through the top words. Also, we reviewed the most frequent words with keyword in negative sentence. This gave us more insights on the tweets and confidence in the analysis result because some word have high numbers in total count and positive count but a very low number in negative count. We filtered out meaningless words, not relevant words, and words that give no hints on QA. We also decide to elicit at least one QA’s from each product’s tweets since they are all in their early stages and accommodating competitors’ strengths is important in this stage.

After reviewing these tables, we found that tweets on Galaxy Gear has lots of words about interactions with other devices such as “note 3”, “phone”, “s4”. Tweets on iWatch showed that many people tweeted with “iphone” and “bluetooth”. Also, there were many tweets on “men”, “women”, and “sizes”. On the other hand, we could not find many meaningful words from tweets on Sony Smartwatch but just found that Sony’s Smartwatch is good at “basics”.

With this preliminary outcome, we developed some hypothetical QA. As for Galaxy Gear, we assumed that people are much interested in interactions or compatibility
with smartphones, and other devices. In case of iWatch, our assumption was that people talked much about interaction with iPhone and different sizes for different genders. Lastly, our assumption for Sony Smartwatch was that many tweets are about basic functions’ reliability.

To make it sure that these assumptions are correct, we delved into the archived files by searching the tweets with our keywords and words we found. By reading the whole mentions with the words, we came to a conclusion on what the words actually meant and we could identify words for prospective QA’s: 1) interaction with other devices, 2) way of communication with other devices, 3) different sizes’ of display, and 4) excellence at basic functionality.

**Step 5. Candidate QA Elicitation**

After discussion on the QA’s, we consolidated two QA’s ‘interaction with other devices’ and ‘way of communication with other devices’ into one, since they can imply a similar requirement. These are the most mentioned QA’s from the tweets on three products. We concluded our analysis with three following QA’s:

1) Compatibility with other devices
2) Supportability for different display
3) Reliability with basic functions

**Phase 1:**

We invited employees from our organization and did a role playing for this workshop. Since our team was exposed to the result data, we wanted to have a QAW with people not in our team and asked them to be in a mock-up QAW. In this paper, full steps’ outputs are not described for conciseness, since this paper is rather focused on the discussion of QA’s from social analysis system and people’s workshop.

**Step 1: QAW Presentation and Introduction**

We introduced QAW and its process to make it sure that everyone in the room understood them.

**Step 2: Business/Mission Presentation**

Our team’s lead presented the business/mission and the project’s mission. She announced that we gathered to achieve our mission to promote Galaxy Gear product, increasing its market share by 50%. Another goal was to implement a killer functionality to become a market leader.

**Step 3: Architectural Plan Presentation**

Since this project is a kind of mock-up workshop and we do not have access to the real product’s architecture, we assumed that everyone has conceptual architecture in his/her mind. This would work because people in the room have seen the product and what functionalities it has as they all work for a consumer electronics company.

**Step 4: Business/Mission Elicitation Game**

In this step, people in the room started writing business/mission in their languages on post-it notes. With the notes on the wall, we started to categorize them into groups.

**Step 5: Quality Attribute Elicitation**

Based on the categorized missions, we started eliciting QA’s. We identified ten QA’s as follows:

1) Energy efficiency / longer battery life
2) Compatibility with voice / motion control system
3) Compatibility with other smartphones and home appliances
4) Extensibility for sensors and healthcare devices
5) Extensibility for new features – adopting features of smartphone
6) Compatibility with context aware system
7) Performance of display
8) Supportability for different hardware specifications – for different editions of product
9) Supportability for open source platform
10) Security – securing personal information

**Step 6. Social QA Presentation**

At this point, we presented the three QA’s that we obtained from phase 0 and had a discussion with the team.

In this paper we do not provide further steps’ output for briefness, which are not relevant to our focus of integrating social listening technique. However, we provide the priorities of the QA’s from the workshop since it allows us to easily compare the priorities of QA’s from FIND and the workshop. QA’s in Step 5 are written in priority ranking from high to low.

6. Discussion

Before we started this application experiment, we made some assumptions. Firstly, the more tweets people write about a topic, the more people are interested in it. Secondly, twitter is the best option for this purpose, since we can hear from its hundreds of millions of active users worldwide. Thirdly, our system’s sentiment analysis is somewhat reliable even though it’s not perfect. Fourthly, other languages than English do not affect the result significantly, since we process only English language.

With these assumptions, we have seen big improvements on our QA elicitation techniques as follows:

1) Determine what QA’s are ones that customers are really interested.

   In people’s workshop, we had lower priorities of QA “Supportability of different display”, and “Interaction with other devices”. However, in FIND’s report, they were the most frequently mentioned QA’s.

2) Obtain QA’s that workshop participants could not think of.

   Workshop participants did not mention “reliability” of the product. However, we could
see that it’s one of the most important QA’s that people like to see through analyzing Twitter’s data.

Our approach was very time-efficient in finding the most important QA’s from Twitter’s data. With the sorted word lists by frequency, positive count, and negative count, we could easily identify important words by filtering out meaningless words, and looking for the tweets with the words. In a couple of scores of minutes, we could figure out what the words represent and what QA’s people talked about.

However, we also identified some weaknesses of our approach as follows:

1) Tweets depend mostly on time.
   People tend to re-tweet the tweets they see today; therefore, depending on when you collect data, the analysis result might vary.

2) We may need a systematic approach to elicit QA’s from FIND.
   The report from FIND system provides just general words that people mentioned. While reading the whole mentions with the words gives the context, elicited QA’s can vary depending on who the elicitor is. We want to avoid depending on individual’s excellence; hence, our solution was to making an agreement through a team discussion.

3) Finding Innovative QA’s would take more time.
   Our approach did not care whom the tweet is from, how useful the tweet itself is, and how good opinion or idea it has. If innovative ideas only come from a limited number of users, those mentions may not be highlighted in sorted lists. Usually, this resulted in ranking well-known words from re-tweeted news article at the top. While we were looking through all the list of the words, however, we could find some very innovative ideas on the product and it will take you more time to delve into the huge data. For example, we could find that some people want the compatibility with other extreme sport devices such as GoPro camera; therefore, the whole list of the words can be a good source for a product manager who wants to develop an innovative product.

We also identified some major weaknesses and improvement points of our system as follows:

1) Automate the process of QA list generation
   Currently, our system does not provide any insights or a feasible list of QA’s. We manually filtered out irrelevant words and looked for the tweets with the word. We aim to automate this process gradually to generate the list of QA’s ready to use.

2) Support multiple languages
   We found that many tweets are written in all different kinds of language not only English or Korean. This made us have lots of word items having the same meaning in different languages. Later, we plan to use some translator to sum all languages’ tweets with the same meaning.

3) Adopt a follower-weighted approach
   Tweets from influential users would be more important than a virtual user with no followers. It would be more reliable to give weights on users having more followers or on influential users.

Twitter was an excellent source of data in that it provides very responsive API’s with huge data from hundreds of millions of users worldwide. We also considered Twitter as an unbiased source unlike news websites or blogs. However, we have some concerns about it as follows:

1) Are twitter users representatives of our end users?
   We are unsure of if they are really interested in our product or giving reliable feedback. We also found that many of tweets are just re-tweets. People may re-tweet it since they are interested in the tweet; however, this may affect our analysis negatively by increasing the unwanted number of words and not giving meaningful user’s feedback.

2) Are tweets reliable when the number of tweets on a keyword is very small?
   We experienced this problem with “google watch” keyword to see most of tweets are not related with the product, which may be related with the Twitter’s API.

7. Conclusion

With this experiment, we gained many insights on our approach’s application as we described in Section 6. This approach was very helpful in that it was time-efficient using large user-based system, Twitter. We could hear from millions of people with minimal efforts without paying lots of money.

In next versions of our FIND system, we plan to automate the manual process gradually to generate the list of QA’s once we just input keywords. Since we see many different words that have the same idea or intention, introducing machine learning to our system would be interesting to generate the result based on the meaning not the words. Also, we plan to make our system more perfect by merging Twitter’s data with data from other sources such as the product’s user community or blogs.

Since this method is in its very early stage, we will continually modify and make some improvement in process and analysis technique with applying this method to real projects. We also need to develop more systematic approaches on deciding how to prioritize between QA’s from the workshop and from the system analysis result. FIND’s report or twitter’s data is not a silver bullet.
References