4/2/2019

Crowd-based Requirements Engineering

Progress Report

Altug, Yusuf Cagri

Contents

Introduction and Motivation

Although different project management methodologies exist on software development engineering, there is always such a phase that you have to gather requirements. Every process involves this stage. It shows that Requirements Engineering has a crucial role on software development process. The main purpose of the software development process is to fulfill the requirements without missing any item of requirements. Most of the time a software can be considered as poor by looking whether it completes its requirements or not. However, what happens when requirements are poor? In such a case, even if your software fulfills the all requirements, it can fail or cannot stand a long time.

Obviously quality of a software is proportional to quality of the requirements. On the other hand quantity does not mean that software is successful. Quantity of the requirements represent the size of the software. In traditional requirements engineering, requirements are elicited with stakeholders who are specialized persons on their domains by using techniques like interviews, workshops, ethnography. Every item is validated by these stakeholders. This approach makes it possible to find out detailed requirements which is called as *depth* in some literatures. However, it is hard to determine requirements of every aspect of software which is called *breadth*. In requirements engineering, the purpose is to find out optimum between these two. To find out the optimum requires a huge involvement of the stakeholders. This causes loss of time and money.

Requirements are elicited according to needs of the users. These needs are determined by the stakeholders in the gathering requirements stage. However, nobody except users knows what they exactly need in the first place. There are lots of feedbacks and related information about these at runtime. In order to analyze and react to these information, lately a new approach called as *Data-driven RE* , *Feedback-based RE* , *Crowd-centric RE* , *Crowd-based RE* in the requirement engineering literature. This article aims what is this new approach, what are the pros and cons and what are the theoretical approaches and the practices on the market.

Crowd Based Requirements Engineering Approach

A *Crowd* is a group of people who are interested in a particular product. This makes these people stakeholders of the product. They interact with each other through the product or other communication tools. This causes that they effect each other's ideas and decisions about the product. They continuously generate data mostly in text format like reviews, reports, documents and so on. Also there are auto generated log data like duration on which page, user operations and mouse clicks. All these data exists on somewhere like databases or cloud and ready to mine. Mining these and extracting something meaningful can help to determine the requirements of the product. The main idea of the crowed based approach is to use crowed as requirements sources by the data they produce while using the product at runtime.

This approach has some important steps to complete. These are gathered under the titles respectively.

Motivation of the Crowd

First step of the approach is to gather a motivated crowd. Users continuously use the product, but willingness of giving feedback about the product is not always there. Without the feedback from the users, it is hard to determine requirements just by looking usage logs. That is a significant challenge for the approach. For the challenge above there are some solution techniques like *gamification*.

Even if you can find the crowd, there will be always some user types like loyal ones, the ones who are tolerant to privacy or not and so on. While loyal user can give feedback anytime and in any circumstances, the ones who are not tolerant to share personal data may not be willing to give feedback. This is also a challenge that should be solved.

The more detailed explanations about the challenges and related solutions will be given on the paper itself.

Gathering and Analyzing User Feedback

There are lots ways to gathering feedback. For example social media like Twitter, Facebook and Instagram can be used to get feedback about the product. Also, making survey is another way. These kind of channels of gathering feedback can create huge amount of data. Bigger the size of data, harder to examine it. The main challenge hare is to create meaningful information. Considering the type of the data which is most likely natural language in text, linguistic analysis techniques can be very useful. Also fast growth of machine learning and artificial intelligence technologies give promises about this.

Also these data can be classified into categories like bug report, feature requests. This categorization can help to specify more detailed requirements. The more detailed explanations about the challenges and related solutions will be given on the paper itself.

Implementation of Requirements and Monitoring

After creation of the requirements, implementing those is the duty of the development team. However, after every iteration on the software development process you should observe the results. That is the same for our approach. Actually the procedure is the same as the stage *Gathering and Analyzing User Feedback*. First the data should be gathered, second the data should be turned into a meaningful statement. After that user satisfaction can be understood. Monitoring of the data is a continuous process. There are lots of examples on the market like product reviews and related fix versions in *Apple Store.* The way how to monitor user satisfaction will be examined on the paper. There is a continuous challenge on every step of this approach.

Conclusion

Crowd Based Requirements Engineering seems promising. This approach can reduce cost of eliciting requirements. It can rises quality of the requirements with a proper quantity. Also with this approach live products can easily adapt themselves to changing needs of users. They can survive in this challenging market. However, the challenges are not the ones easy to solve. Especially, understanding the input of the user is very important to gather proper requirement from it.

In this progress report I got the basic information about Crowd Based Requirements Engineering. In the following article I will examine more article. I will give example about how to use solutions for related challenges. I will give more details about the stages and state of art.

References

Eduard C. Groen, M. K. (n.d.). *How Requirements Engineering can benefit from crowds*. Retrieved from re-magazine: https://re-magazine.ireb.org/print/how-requirements-engineering-can-benefit-from-crowds

Ghanyani, U.-S., Murad, M., & Mahmood, W. (2018). Crowd-based Requirement Engineering. *I.J. Education and Management Engineering*, 43-53.

Groen, E. C., Seyff, N., Ali, R., Dalpiaz, F., Guzman, E., Hosseini, M., . . . Stade, M. (2017). The Crowd in Requirements Engineering The Landscape and Challanges. *IEEE SOFTWARE*, 45-52.

Snijders Remco, D. F. (2014). Crowd-Centric Requirements Engineering, Hosseini Mahmood ,Shahri Alimohammad ,Ali Raian. *ACM 7th International Conference on Utility and Cloud Computing* (pp. 614-615). IEEE.