CROWDSOURCING FOR REQUIREMENTS GATHERING: AN OVERVIEW

MUHAMMAD FAHMI ISMAIL
SABRINA TIUN
NOOR HASRINA BAKAR

Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia

ABSTRACT

Requirements gathering play a vital role in software development, for software to be usable and likeable it must cater the best to communicate the needs, ideas and wishes from millions of stakeholders, hence always rely heavily on user's perspectives. Thus, crowd involvement is becoming significant in requirements gathering phase of software development be it for software evolution or even for development of new software products. In addition, with nowadays technology the requirement from user can be access throughout review of the software in many reliable sources which part of crowdsourcing. Thus, this paper will present a review on crowdsourcing engineering regards to the previous and recent practices; as well as discussion on potential research area in this field.

INTRODUCTION

Requirements Elicitation or requirements gathering is defined as the progression of obtaining a complete understanding of stakeholder's requirements. It is the initial and main process of requirements engineering phase where "what is to be done" is elicited and modeled (Leite 87). Elicitation process usually involves interaction with stakeholders to obtain their actual needs: what and how they imagine the software to be developed will solve their current problems. The requirements elicitation activities help stakeholders to express their needs and expectations from the new system (Masooma, Asger 2015). One of the traditional methods of requirement elicitation is interview. For many years, team projects interview users and stakeholders for developing a new system or improving existing systems. It is true that interviews can be very successful in many occasions and it does help developers to understand the users need. However, there are also some drawbacks of interviews such as interviewee did not reveal their actual needs due to incomplete understanding of their needs. Additionally, sometimes requirements captures are ambiguous and its scope are ill-defined. There are also cases of miscellaneous miscommunication issues during RE process. Most of the time, important stakeholders who are from managerial team usually are busy people. Their time for RE interviews can be very limited. This may results in some important information might not be delivered during the RE process. Due to fundamental problems during traditional RE activities, developers option for an easier RE activity, that is via online collaboration including mining user reviews from the crowd (crowdsourcing). This is believed can provide benefits for software evolution and new releases.

PROJECT PROBLEM STATEMENT AND OBJECTIVE

Reuse of requirement is very tedious and error prone if done manually. Thus, an automated approach is needed to expedite the reuse process. Requirement documents from legacy systems are usually left in the archive, went wasted. Although improvements to the retired system are usually done from the weaknesses of previous system, very little work proposed on modernizing the new system through reuse of legacy documentations with crowd (potential users) involvement, i.e through automated text extractions. The requirement engineering phase of traditional software development usually did not consider crowd involvement in gathering requirements. As a result, the developed software products sometimes did not meet crowd expectations. It is essential to encourage crowd involvement at the early stage of software development to ensure the software produced can meet potential users expectations. Using unstructured data (from related works) to represent requirements may cause lower precision values. Consequently, the existing approach failed to automatically cluster similar features that were extracted from unstructured textual data. Therefore, the purpose of study is aimed to achieve the following objectives: (1) To identify what are the current approaches for requirements gathering with crowd involvements (2) To collectively summarize the quality of the approaches in the selected studies, and (3) To identify research implications and highlight potential research areas for crowdsource requirements engineering.

LITERATURE REVIEW

Requirement Elicitation

Requirements elicitation process is definitely an important part of system development. Elicitation process requires understanding the user needs of the system before embarking on the system development or system maintenance project. It is the initial and main process of requirements engineering phase. Elicitation process usually involves interaction with stakeholders to obtain their real desires. RE consists the following steps: planning, analyzing, determining, integrating and monitoring the method that need to be executed. For many years, team projects interview users and stakeholders for developing a new system or improve it. Undeniable interview has work in so many occasions and it has helped developers in understanding the need of users in one system, but for a long time of use of this technique, there is occurs weakness in implemented it. The cons of the interview as mention such as in formal approach, interviewee may uncomfortable not revealing much information that needed thus, no new concept can be explored. Then, since questioned are not predefined, so interviewer may lose focus to other question that might be important. After that, it is difficult to repeat in case data reliability is checked (Masooma, Asger 2015). Therefore, after the weakness of traditional technique has been found, the new method has been created, where the crowdsource come in. Because in the IT world, there is always a new ways in gathering requirement thus make it efficient in acquire the information to improve the existing or legacy systems. As crowdsource being part of requirement gathering as well, and because of the limitation for require information from the traditional method such as, interview, and system documentation that being done in document analysis. Thus, crowdsource method using mass user via internet known as crowd involving in giving idea, suggestion, and feedback for the systems after going through beta test. Furthermore, this kind of interaction helps for project team in updating their system to next version or publish new program. There is a benefit of using the crowdsource in

requirement elicitation process, one of it is, and many type of user is participated in providing the data that can be retrieved from many platforms such as internet. For example, the stakeholders are involved in various phases such as crowd involvement, focus groups and the development sprint. Each of those phases contains requirements from a certain level of detail and development that have being done, or in progress. By separating the phases, stakeholders experience the results of their input and are encouraged to bring requirements to the next level with different method in gathering it (Snijders et.al, 2015). Therefore, the organization just needs to use their sources to extract the data from many platforms especially the use of internet for collecting the requirement to modernize the software that they had. There is an emerging organize concept of crowdsourcing where new ideas submission from outside the organization boundaries are obtained, selected, evaluated, coded and integrated into the organizations (Irene and Chris, 2018). This process emerging can build the crowdsourcing ecosystems which make it stable and not in a mess way with the information overload that occurs nowadays. Crowdsourcing built with ecosystems is needed to be include within the Software development itself. For example, the instruments that consists of requirement tools, design tools, coding tools, compilers, debuggers, IDE, performance analysis tools, testing tools, and maintenance tools. All this aspect can be tackling by dividing into smaller task and delivered to the crowds. Second is Project management with elemental value that can be include with ranking, reputation and award systems for products and participants who perform and finish the task given first. This kind of motivation ensures the increase of productivity and good quality of solutions. After that is Social network tools that let member to correspond and support each other. For example, it is provide with ease of use for communication among others, many documentation that can be shared throughout platform and place for sharing info and spread knowledge like blog, twitters, comment or reviews in site space, and indexing for reference. Via today ease of access for technology, crowds can ask question and help others in providing accurate information. Lastly is collaborating tools for example, a blackboard platform where user can see a common area and propose plans to develop the results presented in the common area. Such as provided platform where programmer task are posted and crowds can adjusted, propose and find solutions together.

Crowdsourcing Concept and Its Ecosystem

Crowdsource is an act to acquire information or input from user reviews which derives from mass community in the internet either paid or unpaid to retrieve it. Crowdsourcing is an emerging online distributed problem solving and production model where a problem is solved through the involvement of a large number of people (mhosseini, et al 2014). Organization uses crowdsourcing to build system or upgrade their version of software based on user suggestions and feedbacks. Because gathering information from users via internet is much easier nowadays. As an example, organization can retrieve many user reviews that being published by analyzing user comments in social media, video online such on YouTube or consumer experience throughout the beta test of the product as internet is now the biggest channel of networking for any type of audiences for information gathering. System developers and software engineers are able to encounter a wider scope of user from general public, or the crowd. The term crowdsource was first coined in June 2006 on an article in Wired magazine. The term crowdsourcing represents the action of organization making decision to switch the task that was perform by employees to undefined large network of people in the form of outsourcing or open call (Howe, 2006). As the rise of mobile

technologies, e-commerce and social media in the last decade, crowdsourcing play a vital role in developing every industry to make the most of it. For example, the most referred platform for software development such as stackoverflow, TopCoder and GitHub have increasingly used crowdsourcing to harness the idea from users. Collaboration among developers to help each other in providing solution through the above mentioned platforms grows tremendously.

Gathering Requirements from Crowd

There are examples from using the reviews from the user as a source of reuse requirements. Such as considered user reviews that lead or inspired towards crowd involvements (Bakar et. al, 2016). There are studies that addressed the use of function such as comments in the software or mobile app for one purpose, which is to evaluate the user requests, suggestion or idea to adapt and enhance the features based on user preferences in shaping the features of the systems. Thus, user review can be as prime example of crowd participation and as one of the crowdsourcing technique. For example, many approaches were used in this particular subject area. The studies from the earliest 2000s claim to extract frequent product features from online customer reviews, categorize opinion word such as positive or negative reviews and provide a summary for this collected review. The objective was to supply a review of common product features that can be helpful for future customers and manufacturers. In this approach, simple nouns and verb groups were classify through syntactic chunking. Additionally, fuzzy comparable was used to deal with word option and misspelling (Hu and Liu, 2004). Some studies explain the important of availability for analysing the user reviews for third-party software or mobile applications as a way to harness requirements for future releases of a software. In this research, the used of topic modelling are proposed to extract the most important matters from the client feedback and appraise them on different publicly available data sets (Carreno and Windbladh, 2013). Another study shows the development of Mobile App Review Analyser (MARA), that is able to extract and used review from online mobile apps. MARA was intended to assess all the reviews availability, mining the content and identifying sentences expressing such as requests, contents, and present the summarization in a user- friendly approach (Iacob and Harrison, 2013). There are research which used collocation discovery to take out fine-grained aspects, make use of reaction analysis to extract estimation associated to the attribute, and applied subject modeling to many of user features (Guzman and Maalej, 2014). This research claim to extracted reviews data from both iOS and Android and compare up to the results with manually peer analysed reviews. The outcomes indicate that their proposed approach is effective in extracting the most regularly mentioned features. This type of crowds features are rationally and applicable to app requirements, and sentiment analysis results positively associate to the manually assigned scores. Some research recommends the use of mix reliance models to mine product quality from free text. This proposed approach exploited lexical relations and view perspective to classify features, and the result indicated a reliable improvement in the average precision and recall (Khan et al, 2014). In the end, all this research trend regarding user reviews, indicate the activeness of the participations from the user regarding sharing ideas, give opinions, help with information, all lead to the foundation of active crowd. Thus, with an active user, the use of crowdsourcing for reuse and gathering requirement will be successful. For example, uses user reviews to trace the changes made on mobile apps evolutions (Palomba, 2018). In their work, their findings indicate that on average developers implement 49% of informative user reviews while working on the new apps release.

These changes are important because it is usually followed by an increase in ratings. Secondly, the authors surveyed the developers to confirm that changes made to apps of new versions are made based on crowd feedbacks.

Software Development with Crowd Involvement

From website definition itself, Stackoverflow is a question and answer site for proficient and devotee programmers. It is operate and develop by users as part of the stack exchange network of Q&A sites (Stackoverflow, 2018). Nowadays, the trend of people post question regarding programming in this type of sites is increasing. For example, by asking question for solution, in a matter of time the answer will be given by other user either the answer it suit or not with the one who ask for it. The increasing of trend regarding this sites because its provide a platform for user to ask any question and can give answer about wide variety of programming topics. With this growing of question and answer process, the ideas sharing and information is expanded, because this type of medium accumulate a mass volume of knowledge and include many of lines of source code that can be refer to all users. Other factors of increasing trend is many users can benefit from the source that being shared that simply attached to the question and answer space given and user can just copying and learn from it. In addition, many user can utilize the coding to the max because the answer often contain source code snippets from the provided Q&A platforms (Yuhao et al, 2018). In the nut shell, the trend people post question about programming in this type of platform and websites, lead to the increases of users that lead to the crowdsourcing technique can be harness via sharing ideas and knowledge. Thus, with the mass of information, all the task and work can be complete within the short period of time. Github is a web based version control and collaboration platform for software developers. It is delivered through software as a service (Saas) business model and facilitates social coding by providing a web interface to the code repository and management tools for collaboration with other users thus can be though as a social networking for many software developers (Margaret, 2016). Github also being acknowledge as largest social repository which active github user ask fewer questions and providing more answers than any others platform (Bogdan et al, 2013). The user share and post codes in Github through the version control systems, which is a system that allows developers to document, share and merge changes of software. For example, when developers create a system, they can make changes to the code and release the new versions of the system up to and after the first official (non-beta) release. For a team to develop better software, they need people collaboration in the software engineering. As we know, doing collaboration might be a challenging process especially across different time zones and without face-to-face meetings. Therefore, developers use collaboration tools like GitHub to allow the process of collaborating among the developers happened easily (Kleine et al, 2012). It hosts the source code projects in a variety of different programming languages and keeps track of the various changes made. For example, there will be folder that can be expanded and tag with one with the old version or newest. The use of GitHub as repository in crowdsourcing is seems as a good pace as support collaboration between developers and allows users to access and sharing their project with the teammate or the public. If a developer wants to contribute on a project, they just fork the project, make the changes and then send them a pull request to the original member of the repository using GitHub web interface. The benefits that gain from these joint technologies ensure the potential of increasing the effectiveness of operations in any services

and enhance the efforts to crowdsourcing ideas and knowledge thus improve or tackle any issues faced in a task of project.

Natural Language Text Processing

Some of the technique that consider as NLTP are feedback annotation that called by CRAFT, which is harnessing the power and wisdom of the crowd. The validation for this approach acknowledges challenges that existing in text mining applications to dealing with the peculiarities of natural language processing. This validation check the plugin to approve the effectiveness of real world feedback forums to obtain crowd feedback annotations to show using text mining techniques is possible. This will allow validation procedure a thorough prove CRAFT abilities and limitations. The validation purpose to enriching the feedback categories by assembling user generated categories and shaping them into well defined categories of feedback. This will provide requirements engineer a powerful taxonomy of user-perceived feedback categories. In addition, the identification of a language to be used in CRAFT also being validated to give insight for the crowd members so that there is no difficulty in understanding each item listed in the feedback categories. The validation results in case study suggest that CRAFT has the potential to be an effective means for feedback annotation in place or in addition to text mining applications to reduce monetary and time-related costs of user feedback analysis that lead to creating a plugin or an application for the CRAFT technique following the gamification design principles (Mahmood et al., 2017). Second is the use of GitHub apply the Natural Language Processing (NLP) in the project call GH4RE which is Repository Recommendation on GitHub for Requirement Elicitaion Reuse The filter activity mention in this study being use to perform a filtering process that make use of POS-tagging techniques to change unstructured data into structured data by distilling significant assets, such as verbs, nouns, and proper nouns. Therefore, the activities recognize the usual preprocessing tasks in texts such as removing numbers, whitespaces, and non-alphanumeric terms. The readme text in the project was exported as comma-separated values (CSV) file and presented for data exploration. The activity uses the NLP processing approach to reveal frequent words that might attract user attention due to the lack of user preferences for readme texts. The frequency of words was computed and a Wordcloud visualization technique was applied to display the data. (Roxana et al., 2017)

Machine Learning/Algorithm

Clustering Algorithm or known as cluster analysis is an unsupervised machine learning to select natural grouping and patterns unlabeled data. There are much type of clustering methods such as KMeans and hierarchical clustering. Therefore, having practices in industries suggest new theoretical model such as algorithm and tools that needed to facilitate any software crowdsourcing from industrial practices and research efforts regarding theoretical models and algorithms that have being apply (Emese, Matt and Wei, 2016). Some algorithm handle large of data such as fuzzy collaborative filtering that apply in web based requirement elicitation tools. This collaborative approach identifies and prioritizes requirements of different stakeholders and as recommender engine for requirements. In addition, the use algorithm for requirements prioritization such as Binary Search Tree (BST) in simple, unambiguous manner which will help to make the decision

regarding implementation of a requirement after collecting it from the stakeholders. For example, some expert uses information theoretic approach to protect confidential information such as the algorithm ontologies. This is important studied approach the implementation of crowdsourcing using fuzzy and machine learning (Alpana, Kumar and Gurdeep, 2017). In addition, uses of item-to-item collaborative filtering algorithm to foresee a stakeholders preference for an unrated requirement carry the requirements prioritization by suggesting requirements that are identified by collecting preference information from many users. For example, calculations are produced based on similarities between items such as same requirement that are likely request. Whereby the algorithm matches stakeholders requirements, then combines any similar requirements into a list of recommendations for the stakeholder. To decide the related match for a given requirement, the algorithm discovers requirements that stakeholders be likely need. Thus, the collaborative filtering algorithm incorporates new ratings of requirements dynamically and helps in design the decisions The use crowdsourcing via social network analysis and collect requirements and their ratings and run the collaborative filtering algorithm after that recommends other requirements and calculates the priority of each requirement. Some of the web based requirement elicitation tools infuse with fuzzy collaborative filtering that will prioritize requirement from the stakeholders. This help stakeholder in rate requirement and rank the requirement based on binary search tree approach. The claim being support provides with layout of proposed work such as develop the tools, generated requirement by using singular value decomposition algorithm and prioritize it by using BST -This development is important to shows on work regarding requirement engineering area by improving requirement elicitation process and infuse with fuzzy, thus make it clear on which requirement is most needed. Therefore can be reference how the method using fuzzy logic can be applied in crowdsourcing to improve requirement engineering and importantly upkeep software modernization. Significant findings from this prove on implementation of fuzzy and algorithm help to achieve the overall quality throughout the use of the developed tools save time, and cost in requirement elicitation process thus abolish bias that occurs by using novel method such as fuzzy collaborative filtering makes it more accurate in gathering the requirement from the stakeholders (Shinde, Kaushik, Shashank, 2015). Some research on web based tools that initiate with collaborative filtering from crowdsourcing approach to manifest the prioritize of stakeholders and their requirement for collect requirement and rating from each stakeholders via collaborative filtering algorithm as a result, it will prioritize each requirement before starting the software project. Therefore, the process of collaborative filtering algorithm takes places, just to incorporate and sort rating of requirement dynamically (Soo, Daniele, Anthony 2011). In the nut shell, applying machine learning towards project really give ease of use for modernization of systems.

CONCLUSION

The purpose of this project is to understanding the automated approach which selecting requirement from user reviews that can be more systematic and easier. With this understanding, application being develop for use the proposed extraction algorithm that can be demonstrate as a useful process model for analyze requirements that encouraging crowd involvement that hoping to help in software development and modernization.

ACKNOWLEDGEMENT

This project is financially supported by UKM under research code: GGPM-2017-025.

REFERENCE

Leite, and Julio Cesar S P, 1987. A Survey on Requirements Analysis. Advanced Software Engineering Project Technical Report RTP-071, University of California at Irvine, Department of Information and Computer Science, June 1987.

Masooma and Asger, 2015. Comparison of Various Requirements Elicitation Techniques. International Journal of Computer Applications 116 (4): 8 - 15 April 2015.

E. Bari, M. Johnston, W. Wu, and W. T. Tsai, "Software crowdsourcing practices and research directions," Proc. - 2016 IEEE Symp. Serv. Syst. Eng. SOSE 2016, pp. 372–379, 2016.

D. Mu et al., "Gathering Requirements for Software Configuration from the Crowd," 2017.

M. Hosseini, E. C. Groen, A. Shahri, and R. Ali, "CRAFT : A Crowd-Annotated Feedback Technique," 2017.

F. Palomba et al., "Crowdsourcing user reviews to support the evolution of mobile apps," J. Syst. Softw., vol. 137, no. March 2018, pp. 143–162, 2018.

M. Stade et al., "Providing a User Forum is not enough : First Experiences of a Software Company with CrowdRE," 2017.

E. Groen, J. ; Doerr, and A. Sebastian, "Towards Crowd-Based Requirements Engineering A Research Preview Eduard," in REFSQ 2015, LNCS 9013, 2013, vol. 7830, pp. 247–253.

M. Hosseini, A. Shahri, K. Phalp, J. Taylor, and R. Ali, "Crowdsourcing : A taxonomy and systematic mapping study," Comput. Sci. Rev., pp. 1–27, 2015.

M. Hossain and I. Kauranen, "Crowdsourcing : A systematic literature review."

N. Hasteer, N. Nazir, A. Bansal, and B. K. . Murthy, "Crowdsourcing Software Development: Many Benefits Many Concerns," in International Conference on Information Security & Privacy (ICISP2015), 2016, vol. 78, pp. 48–54.

F. Fotrousi, N. Seyff, and J. Borstler, "Ethical Considerations in Research on User Feedback," in 2017 IEEE 25th International Requirements Engineering Conference Workshops Ethical, 2017, pp. 194–198.

F. M. Kifetew, D. Muñante, E. Letier, and B. Kessler, "Discovering Requirements through Goal-Driven Process Mining," in 2017 IEEE 25th International Requirements Engineering Conference Workshops, 2017, pp. 199–203.

K. Schneider, O. Karras, A. Finger, and B. Zibell, "Reframing Societal Discourse as Requirements Negotiation : Vision Statement," in 2017 IEEE 25th International Requirements Engineering Conference Workshops Reframing, 2017, pp. 188–193.

E. Bakiu and E. Guzman, "Which Feature is Unusable ? Detecting Usability and User Experience Issues from User Reviews," in 2017 IEEE 25th International Requirements Engineering Conference Workshops Which, 2017, pp. 182–187.

H. Tajedin and D. Nevo, "Determinants of Success in Crowdsourcing Software Development," in SIGMIS-CPR'13, 2013, pp. 173–178.

A. Dubey, K. Abhinav, and G. Virdi, “A framework to preserve confidentiality in crowdsourced software development,” Proc. - 2017 IEEE/ACM 39th Int. Conf. Softw. Eng. Companion, ICSE-C 2017, pp. 115–117, 2017.

S. L. Lim, D. Quercia, and A. Finkelstein, “StakeNet: using social networks to analyse the stakeholders of large-scale software projects,” 2010 ACM/IEEE 32nd Int. Conf. Softw. Eng., vol. 1, pp. 295–304, 2010.

H. Wang, Y. Wang, and J. Wang, “A participant recruitment framework for crowdsourcing based software requirement acquisition,” Proc. - 2014 IEEE 9th Int. Conf. Glob. Softw. Eng. ICGSE 2014, no. January, pp. 65–73, 2014.

S. L. Lim, D. Quercia, and A. Finkelstein, “StakeSource: harnessing the power of crowdsourcing and social networks in stakeholder analysis,” 2010 ACM/IEEE 32nd Int. Conf. Softw. Eng., vol. 2, no. 2, pp. 239–242, 2010.

N. Hasteer, N. Nazir, A. Bansal, and B. K. Murthy, “Crowdsourcing Software Development: Many Benefits Many Concerns,” Phys. Procedia, vol. 78, pp. 48–54, 2016.

T. T. D. Latoza and A. van der Hoek, “Crowdsourcing in Software Engineering : Models , Opportunities , and Challenges,” IEEE Softw., pp. 1–13, 2016.

S. L. Lim and C. Ncube, “Social networks and crowdsourcing for stakeholder analysis in system of systems projects,” Proc. 2013 8th Int. Conf. Syst. Syst. Eng. SoSE Cloud Comput. Emerg. Inf. Technol. Appl. SoSE 2013, pp. 13–18, 2013.

A. Adepetu, K. Ahmed, and Y. Al Abd, “CrowdREquire: A Requirements Engineering Crowdsourcing Platform,” 2012.

S. L. Lim and A. Finkelstein, “StakeRare: Using social networks and collaborative filtering for large-scale requirements elicitation,” IEEE Trans. Softw. Eng., vol. 38, no. 3, pp. 707–735, 2012.

D. Damian and A. Finkelstein, “StakeSource2. 0 : Using Social Networks of Stakeholders to Identify and Prioritise Requirements,” in Internation Conference of Software Engineering, ICSE 2011, 2011, pp. 1022–1024.

S. Lim, “Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation,” Ph.D. Thesis. Sch. Comput. Sci. Eng. Univ. New South Wales, Sydney, Austrailia., pp. 1–32, 2010.

J. Howe, “The rise of Crowdsourcing,” Wired, 2006.

K.-J. Stol and B. Fitzgerald, “Two’s company, three’s a crowd: a case study of crowdsourcing software development,” in Proceedings of the 36th International Conference on Software Engineering - ICSE 2014, 2014, pp. 187–198.

K. Mao, L. Capra, M. Harman, and Y. Jia, “A survey of the use of crowdsourcing in software engineering,” J. Syst. Softw., vol. 126, pp. 57–84, 2016.