International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075, Volume-8 Issue-8S, June 2019

Prioritizing based on Crowd Preferences to Requirements Elicited from Crowd's Sentiments

Nurul Afiqah Abdul Aziz, Sa'adah Hassan, Novia Admodisastro

***Abstract:** The current trend in software engineering is using crowd as source of ideas for developing software product.* **Crowd** *is described as a large and heterogeneous group of stakeholders.* **Crowd** *is also used as a source of requirements, for example, the requirements can be elicited from their reviews, reports, or feedback. However, the challenge is to prioritize the requirements knowing that the data is also large and heterogeneous. Requirements prioritization is an approach to identify requirements that need to be selected and implemented first in software development. This paper presents an approach to prioritize requirements elicited from crowd and the prioritization is based on the crowd preferences in order to ensure the customer satisfaction. A prototype tool is developed as a proof-of-concept and evaluation purposes. It is expected that the proposed approach able to helps developer to prioritized requirements based on customers' preferences as it would consider their satisfaction of the software product.*

Index Terms: crowd, elicitation, requirements prioritization

I. INTRODUCTION

The idea of using crowd to get more ideas is getting noticeable nowadays. Where, crowdsourcing is the practice of obtaining product ideas by soliciting contributions from the wider public rather than just from employees or suppliers [1]. Information, such as ideas for improvements, user sentiments derived from reviews and comments from crowd are useful for application developers as potential software requirements [2]. Considering some of the feedbacks also may include suggestions, feature requests, and other messages on improving the application [3]. In general, it represents the 'voice of users' for development and improvement of future product releases. For example, work by Guzman and Maalej [2] that took a large number of reviews to get the right sentiments by using Natural Language Processing (NLP) and data mining technique. By extracting these feedbacks, we would be able to find software requirements that would generate quality scores in developing the product. Previous work [4] also mentioned about an approach to elicit creative and innovative requirements from crowd's feedback and reviews.

Revised Manuscript Received on May 22, 2019.

Nurul Afiqah Abdul Aziz, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia
Sa'adah Hassan, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia
Novia Admodisastro, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia

However, it is almost impossible for software developer to implement all the elicited software requirements. Thus, the elicited requirements need to be prioritized. Requirements prioritization is an essential process in software development to determine which requirements can be implemented under certain constraints, such as time, cost, quality and resources that would ultimately results in customers' satisfaction [5]. The outcome from the requirements prioritization process is a set of prioritised requirements that are significant for the software product. The information about the priorities not only allow restriction to the least important requirements, but also to help project managers to resolve conflicts, make plans for the deliverables and necessary trade-off [6]. Requirements prioritization has been a concern for software product development and thus, there are many research and tools creation to resolve the prioritization issues. Previously, according to Lehtola, *et al.*, [6], a lot of company goes by tacit knowledge from experiences or feelings to determine the truly important requirements. Nowadays, there are various prioritization approaches can be adopted by developers, such as pair wise comparisons, and negotiation approaches [7]. However, it is also mentioned that stakeholders are required to have an understanding on how development works and relies on their learning experiences for the process of prioritizing the requirements [8].

The aim of this work is to propose an approach that focuses on crowd's preferences for prioritizing requirements elicited from crowd. In which, prioritizing requirements elicited from wider scope of end users or prospect customers would gain high level value in the market segment as it would consider their satisfaction towards the product. Accumulating this analysis allows the best basis to the product or release.

The rest of this paper is organized as follows. Section 2 discusses in brief the background and related work. In section 3, we describe the proposed approach. The finding and discussion are then presented in the following section prior concluding this paper.

II. RELATED WORK

Firesmith [9] mentioned requirements prioritization as an essential process in software engineering as it gives perfect implementation order of the requirements for planning software versions and supplying desirable functionality as early as possible.

*Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication*

However, the term 'importance' in defining requirements prioritization need to be specified between the stakeholders in each case. For example, it could refer to the urgency of implementation, preference of customers, importance of requirements for the product or even strategic importance for the company [6]. The challenge is acquiring the right requirements for a software product, in which implementing the right requirements excretes satisfaction of the customer. There are several techniques that have been adopted into selection of right requirements based on criteria such as time, cost, or importance. It depends on the experts, communication with the stakeholders, and other factors in regards to the requirements itself.

Table. 1 Comparisons between the Prioritization Techniques

Technique	Scalability	Stakeholder Participation	Speed (Scale 1 – 8)	Customer Satisfactory	Result Scale
Hierarchy CV	Low	Low	3	Low	Ratio
AHP	Low	Low	8	Low	Ratio
Hierarchy AHP	Low	Low	4	Low	Ratio
100 Dollar Method	Low	High	3	High	Ratio
Priority Group	Low	High	4	Medium	Ordinal
Bubble Sort	Low	Medium	6	Low	Ordinal
Binary Search Tree	High	Medium	5	Low	Ordinal
MoSCoW	Medium	High	2	High	Nominal
Numeral Assignment	Low	High	1	High	Nominal
Kano's Model	High	High	3	High	Nominal

Table 1 tabulated a summary of the techniques based on empirical studies done by other researchers, where requirements ordering that can be represented in three general scales which are, ratio scale, ordinal scale, and nominal scale [10][11][12][13][14]. The ratio scale allows us to quantify how much important of one requirement than another, where the scale often ranges from 0 - 100 percent. While, ordinal scale prioritization techniques produce ranked lists of requirements but can only tell that one requirement is more important than another without mention how much [10][13]. As for nominal scale method, requirements are assigned to priority groups based on the priority with regards to their importance [15]. In addition, Table 2 is a summary of the existing tools that support the prioritization techniques.

Table. 2 Supporting Tools for Prioritization Techniques

Technique	Supporting Tools
Hierarchy CV	-
AHP	easyAHP
Hierarchy AHP	PriEsT
100 Dollar Method	-
Priority Group	-
Bubble Sort	VisualAlgo SORTING
Binary Search Tree	Binary Search Tree Visual Algo
MoSCoW	Scrum Desk
Numeral Assignment	-
Kano's Model	Kano's Survey Scrum Desk

It can be concluded that there are certain prioritization techniques appear to be able to serve a quality-based requirements prioritization, including 100 dollar method, MoSCoW, Numeral Assignment, and Kano's model. Yet, technique that focusing on agile methodology, scalability could only been found in MoSCoW and Kano's model. Since the size of the requirement sets varies according to each project the term scalability is used here to describe the ability of each technique to scale up with increasing number of requirements. However, Kano's model seems the most promising method, in which it aims to provide satisfaction to the customers. Kano's model [16] was introduced to satisfy user needs by listing three different attributes of requirements. It is considered as a good tool to address the effort of implementation of a feature based on the customers preferences, thus pinpoint the quality attributes needed to prioritize requirements [17][18]. Kano classifies features into four categories which are:

Performance: is a linear or one-dimensional attribute in the Kano literature. In this case, it refers to the increase in functionality leads to increased satisfaction of customers.

Attractiveness: it refers to unexpected features where if presented, cause a positive reaction, therefore increasing the satisfaction level.

Must-be: it means that the product feature are expected by the customers and could cause fluctuation that leads to downright satisfaction if it is not presented.

Indifference: this feature fall along the horizontal line where features proposed does not make any difference to the customer whether or not they are present.

In Kano's model, users are asked about the product features to uncover their perception through Kano's questionnaire. It consists of a pair of question that is evaluated through functional and dysfunctional forms. In each form, users are required to answer based on these fixed possible choices; I like it, I expect it, I am neutral, I tolerate it, and I dislike it. There are exceptions throughout the answers provided by customer whereby they could not fully understood the question or the proposed features which are not exactly what they are looking for.

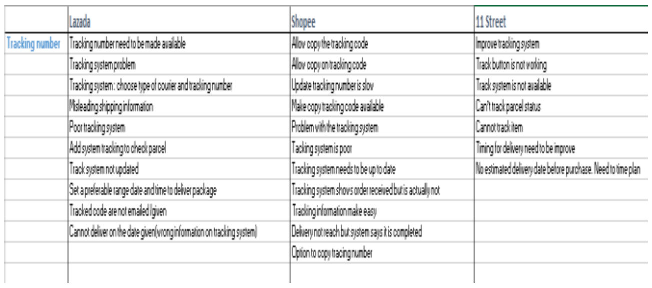
So, they would provide answers like “dislike” for functional question and “likes” for dysfunctional question. In both of these cases, the category would fall into Reverse. For conflicting response such as “like” for both question form, the category would fall under Questionable. Therefore, Walden [19] provides an alternative evaluation that also include questions that asks customer about the feature’s importance by providing a ranking method from 1 to 9, going from “Not at all important” to “extremely important”. This will offer information on the relevancy of the feature to the customer. While, Berger *et al*., [20] proposed a quantitative analysis of customer satisfaction into Kano's model by calculating customer coefficient through two values, SI (Satisfaction Index) and DI (Dissatisfaction Index). However, the simplest way to work with Kano result is through Discrete Analysis. Then, William DuMouchel [21] proposed continuous analysis methodology that serves the same purpose as SI and DI method, in which it calculate the average Functional, Dysfunctional and Importance values over all answers and the standard deviation for the scores.

III. THE PROPOSED APPROACH

This section demonstrates the proposed approach by using a case study and prototype tool. Prioritization is done based on the crowd’s preferences to the likability of the requirements through functional, dysfunctional and self-state importance. Then, the results are tabulated and analyzed.

A. Requirements Elicitation and Categorization from Crowd’s Sentiments

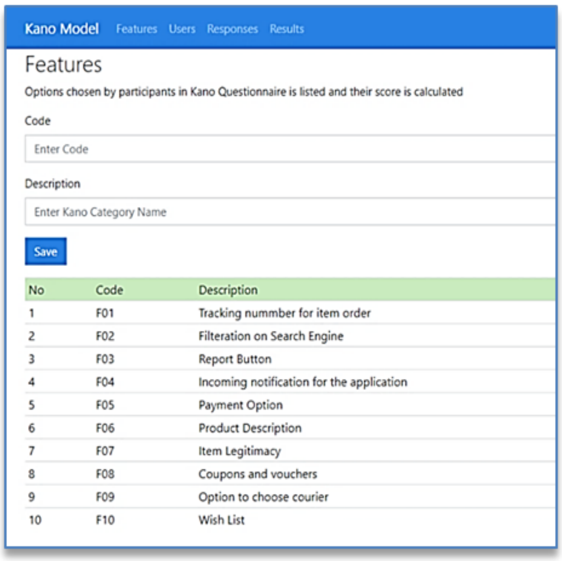
At this stage, it involves the process of extracting data (the crowd’s sentiments) and translating them into a list of requirements to be prioritized. For this study, data were collected from four (4) e-commerce applications that can be found from Google Play [25]. The applications are Lazada, Shopee, Carousell, and 11Street. Each review that talk about the features for the applications were read through and all comments in a positive and negative feedback were recorded. Based on the collected data, each of the feedbacks was compared and sorts them into categories that would fit a requirement. Ambiguous feedbacks are discarded in order to avoid confusion. Figure 1 shows snippet of the sentiments collected from different e-commerce applications. Through the reviews of the users, the sentiments and opinions associated to the features that users most likely concern about are collected and categorized respectively.



	Lazada	Shopee	11 Street
Tracking number	Tracking number need to be made available	Allow copy the tracking code	Improve tracking system
	Tracking system problem	Allow copy on tracking code	Track button is not working
	Tracking system : choose type of courier and tracking number	Update tracking number is slow	Track system is not available
	Misleading shipping information	Make copy tracking code available	Can't track parcel status
	Poor tracking system	Problem with the tracking system	Cannot track item
	Add system tracking to check parcel	Tacking system is poor	Timing for delivery need to be improve
	Track system not updated	Tracking system needs to be up to date	No estimated delivery date before purchase. Need to time plan
	Set a preferable range date and time to deliver package	Tracking system shows order received but is actually not	
	Tracked code are not emailed /given	Tracking information make easy	
	Cannot deliver on the date given(wrong information on tracking system)	Delivery not reach but system says it is completed	
		Option to copy tracing number	

Fig. 1 Sample of Feature from Crowd’s Sentiments

After listing and categorizing the feedbacks, ten (10) main features are chosen that represent the requirements needed to be prioritized. Each of it is tagged with indicator (as requirement ID) as listed in Figure 2. It also shows the interface to insert new feature that need to be prioritized. The feature’s description and code are captured and recorded into the database.



Features

Options chosen by participants in Kano Questionnaire is listed and their score is calculated

Code

Enter Code

Description

Enter Kano Category Name

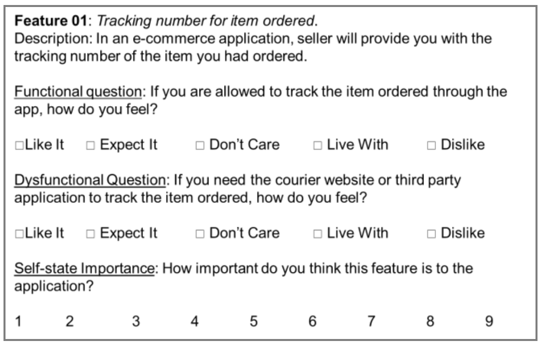
Save

No	Code	Description
1	F01	Tracking nummber for item order
2	F02	Filteration on Search Engine
3	F03	Report Button
4	F04	Incoming notification for the application
5	F05	Payment Option
6	F06	Product Description
7	F07	Item Legitimacy
8	F08	Coupons and vouchers
9	F09	Option to choose courier
10	F10	Wish List

Fig. 2 Screenshot for Feature Input and Listing

B. Kano-based questionnaire

The main objective of this questionnaire is to obtain and determine how the crowd feels over the features, in which, Kano-based questionnaire is used. The questionnaire was distributed to crowd (likely the end users of the product) via email to get their opinion about the features. The questionnaire was designed into two forms of questions that could be translated into a functional (positive) and dysfunctional (negative) questions. The basic concept to these two forms of question is the presence of the feature. For the functional question, it would ask what the user would feel if the mentioned feature would be in presence. For the dysfunctional question, the same choice would be presented but for the absence of the feature. Each of the questions was made clear and only stand for a single feature. In order to be able to tell customers what the product or application can do, the questions are phrased in terms of benefit to the users. The choice of the answers for both of the questions is fixed of 5 choices, which are: *Like It*, *Expect It*, *Don’t Care*, *Live With* and *Dislike*. These two forms of question were justified by Kano, in order to find out the category of the feature, we need to find out the relativity of the questions and the choice of answer that the respondents had provided. The result of the category will provide us with data of the features and it determines whether the questions lay out was understandable and properly executed. Figure 3 shows snippet of the questionnaire as a sample.



Feature 01: *Tracking number for item ordered.*
Description: In an e-commerce application, seller will provide you with the tracking number of the item you had ordered.

Functional question: If you are allowed to track the item ordered through the app, how do you feel?

□Like It □ Expect It □ Don't Care □ Live With □ Dislike

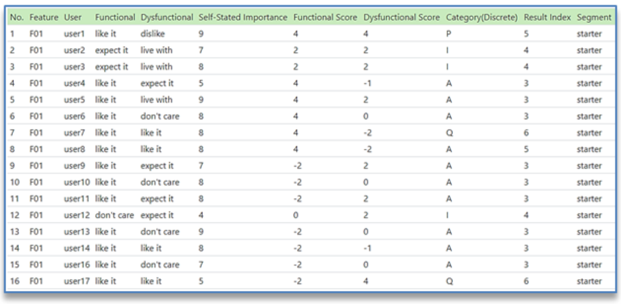
Dysfunctional Question: If you need the courier website or third party application to track the item ordered, how do you feel?

□Like It □ Expect It □ Don't Care □ Live With □ Dislike

Self-state Importance: How important do you think this feature is to the application?

1 2 3 4 5 6 7 8 9

Fig. 3 Snippet of the Questionnaire



No.	Feature	User	Functional	Dysfunctional	Self-Stated Importance	Functional Score	Dysfunctional Score	Category(Discrete)	Result Index	Segment
1	F01	user1	like it	dislike	9	4	4	P	5	starter
2	F01	user2	expect it	live with	7	2	2	I	4	starter
3	F01	user3	expect it	live with	8	2	2	I	4	starter
4	F01	user4	like it	expect it	5	4	-1	A	3	starter
5	F01	user5	like it	live with	9	4	2	A	3	starter
6	F01	user6	like it	don't care	8	4	0	A	3	starter
7	F01	user7	like it	like it	8	4	-2	Q	6	starter
8	F01	user8	like it	like it	8	4	-2	A	5	starter
9	F01	user9	like it	expect it	7	-2	2	A	3	starter
10	F01	user10	like it	don't care	8	-2	0	A	3	starter
11	F01	user11	like it	expect it	8	-2	2	A	3	starter
12	F01	user12	don't care	expect it	4	0	2	I	4	starter
13	F01	user13	like it	don't care	9	-2	0	A	3	starter
14	F01	user14	like it	like it	8	-2	-1	A	3	starter
15	F01	user16	like it	don't care	7	-2	0	A	3	starter
16	F01	user17	like it	like it	5	-2	4	Q	6	starter

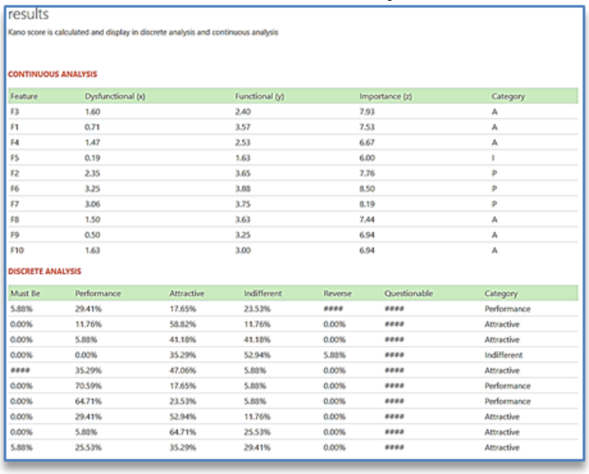
Fig. 4 Sample of Recorded data for first feature

Another essential part of the questionnaire is the self-stated importance of the feature to determine the importance of the feature being present in the product. This information is useful to distinguish the features among each other and knowing which are the most relevant to the users. This question is crafted using a scale system of 1 to 9 with 1 being the least importance and 9 with the very important. The result of the questionnaire is translated into satisfaction levels that are divided into categories as shown in Figure 4.

C. Analyse the Results

The results of the questionnaire were translated into satisfaction levels and divided into categories. Kano rules are explained through basic scores and the categories of the requirements. This will aid the analyst to refer back to Kano's table for any misconducted calculations. Each of the options that the users have provided through the functional and dysfunctional questions is referred to this table. Each of the options translated to numerical value with a satisfaction level from -2 to 4. The larger number reflects that the feature should be available in the point of view of the customers. For the dysfunctional scale, even though that the *dislike* option is referred to the highest number, it also concludes to the same result. This is because; having to *dislike* or to *disagree* with the dysfunctional question would mean to disagree to the absence of the feature. Therefore, there would be more satisfaction if it is included. With the prospect of 'not understanding the questions' and 'underrated feature', the result would then be categorized as 'Questionable' and 'Reverse' respectively. These two categories hold the negative quadrant and are much weaker than the positive quadrant categorized as 'Must-Be' and 'Performance'. We are focusing on the positive quadrants where it holds a stronger response.

When all of the responses has been scored and tabulated, then calculate the results through discrete and continuous analysis [21]. In which, it calculate the average Functional, Dysfunctional and Importance values over all answers and the standard deviation for the scores. Where, the average values of the scores will place the features to their categorization quadrant and the error bars represent the notion of how on or off target the categorization are, which are obtain through standard deviation. The importance score is added by visualising it through the circle size to easily compare the features among similar positioning. But, remain the general prioritization rule of thumb presented in the discrete analysis: Must-be > Performance > Attractive > Indifferent. As for a self-stated importance ranking, average the scores for each feature. Figure 5 shows sample of the results for continuous and discrete analysis.



results

Kano score is calculated and display in discrete analysis and continuous analysis

CONTINUOUS ANALYSIS

Feature	Dysfunctional (x)	Functional (y)	Importance (z)	Category
F3	1.60	2.40	7.93	A
F1	0.71	3.57	7.53	A
F4	1.47	2.53	6.67	A
F5	0.19	1.63	6.00	I
F2	2.35	3.65	7.76	P
F6	3.25	3.88	8.50	P
F7	3.06	3.75	8.19	P
F8	1.50	3.63	7.44	A
F9	0.50	3.25	6.94	A
F10	1.63	3.00	6.94	A

DISCRETE ANALYSIS

Must Be	Performance	Attractive	Indifferent	Reverse	Questionable	Category
5.88%	29.41%	17.65%	23.53%	####	####	Performance
0.00%	11.76%	58.82%	11.76%	0.00%	####	Attractive
0.00%	5.88%	41.18%	41.18%	0.00%	####	Attractive
0.00%	0.00%	35.29%	52.94%	5.88%	####	Indifferent
####	35.29%	47.06%	5.88%	0.00%	####	Attractive
0.00%	70.59%	17.65%	5.88%	0.00%	####	Performance
0.00%	64.71%	23.53%	5.88%	0.00%	####	Performance
0.00%	29.41%	52.94%	11.76%	0.00%	####	Attractive
0.00%	5.88%	64.71%	25.53%	0.00%	####	Attractive
5.88%	25.53%	35.29%	29.41%	0.00%	####	Attractive

Fig. 5 Sample of Results

While, the categorization plane (as shown in Figure 6) helps to visualise the results to better understand the importance of each feature that has been prioritized. The circle represents the feature. From the categorization plane, all of the features belong to the *performance* quadrant. Each feature has about the same amount of importance and marginal errors. Therefore, more solid prioritization according to the rank listed was produced.

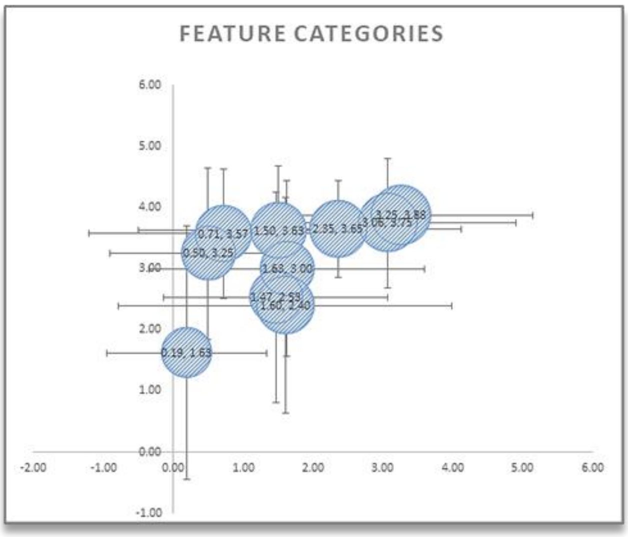


Fig. 6 Results in Categorization Plane

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

IV. FINDINGS AND DISCUSSION

About six developers with various years of experience in software or mobile application development field were interviewed for the feasibility of the proposed approach. Based on the given responses about the current practices, most companies tend to follow the clients' requirement lists and implement them according to the constraints exists, for example, time, cost, budget and resources. They also depend greatly on the particular module that the clients need based on the timeline.

While, there are varying responses regarding the feasibility of the proposed approach. The case study used to illustrate the proposed approach was unable to show clearly on the marginal errors and importance of each feature, since they are all clustered in one place. However, by providing a ranking list based on calculated scores on the user's opinion on functional, dysfunctional and importance features, they found it easily understandable to pick the requirements to be implemented. They also mentioned that the proposed approach would satisfy the quality attributes of product requirements if the feedbacks and reviews from users are properly extracted and categorized in their respective description. Besides, to allow time saving, tools for extraction could be used. Additionally, Kano-based questionnaire should also distribute to the right stakeholders for the software product, and questions that are well structured will determine the best chosen choices of answer. Yet, some application vendors and developers provide recommendations to improve the quality and features of the tool. It is mentioned that the tool is still lack of a good display of the data. It is not flexible for entertaining multiple different developments. The use of dashboard display would help to properly adjusted and view the data without clustering everything in one place. Since the tool provide a mean of automated calculation of data obtained from Kano-based questionnaire, it would be easier to have a .cvs file upload instead of manually entering the data.

V. CONCLUSION

It is a challenge to prioritize requirements that are elicited from wide scope of stakeholders (i.e., crowd), particularly, to ensure their satisfaction of the software product-to-be. There are many prioritization techniques but choosing the right prioritization technique is important as the product quality is heavily influenced by the choice of the crowd's preferences. The proposed approach aims to help developer to acknowledge and make decision on which requirements should be developed first. It is also concern on collecting responses from the crowd through Kano-based questionnaire to make certain their preferences. In which, Kano's model qualifies the desirable aspects and is suitable to be used if the customers' preference is the main focus. Kano's model version of [20] utilized to give values to the requirements based on the attributes and determine which requirements satisfy the customer the most.

In general, the proposed approach does help developer to identify and prioritize users' preferences towards the software product development. However, there is a limitation in which the requirements elicitation from crowd is done manually. Thus, it took a lot of time and tedious job too. Moreover, there were a lot of confusion to properly manage and categorize each feedbacks and reviews. Therefore, for future work, natural language processing and data mining tool will be useful to properly extract the features in fine grain and saves more time on the process. In addition, the proposed tool could be made more interactive by allowing flexibility in inserting, viewing and editing data. For example, just by dragging a *.cvs* file and automatically fill in all the fields needed for the prioritization.

ACKNOWLEDGMENT

We wish to acknowledge support from Universiti Putra Malaysia.

REFERENCES

1. Kepes, Ben. "How to Utilize Cloud Computing, Big Data and Crowdsourcing for an Agile Enterprise." Gigaom Research, 2014, pp.1-17.
2. E. Guzman and W. Maalej, "How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews," *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, Karlskrona, 2014, pp. 153-162. doi: 10.1109/RE.2014.6912257
3. D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," *2013 21st IEEE International Requirements Engineering Conference (RE)*, Rio de Janeiro, 2013, pp. 125-134.
4. S. Hassan, T.A.T.M. Amin, N. Admodisastro, N., & A. Kamaruddin, *Morphological Approach in Creative Requirements Elicitation from Crowdsourcing, J. of Telecommunication Electronic and Computer Engineering (JTEC)*, vol. 9, no. 3-5, pp. 31-35, 2017
5. Babar, M. I., Ramzan, M., & Ghayyur, S. A. K. (2011). Challenges and future trends in software requirements prioritization. International Conference on Computer Networks and Information Technology, (July), 319–324. https://doi.org/10.1109/ICCNIT.2011.6020888
6. Lehtola L., Kauppinen M., Kujala S. (2004) Requirements Prioritization Challenges in Practice. In: Bomarius F., Iida H. (eds) Product Focused Software Process Improvement. PROFES 2004. Lecture Notes in Computer Science, vol 3009. Springer, Berlin, Heidelberg, pp 497-508.
7. Shehzad, K., Awan, M. D., & Rizvi, S. S. (2014). A Hybrid Technique based on Standard SRS Modules for Software Requirement Prioritization A Hybrid Technique based on Standard SRS Modules for Software Requirement Prioritization, (July).
8. Rami Hasan AL-Ta'ani, Rozilawati Razali (2013). Prioritizing Requirements in Agile Development : A Conceptual Framework. Procedia Technology, vol.11, pp.733–739. https://doi.org/10.1016/j.protcy.2013.12.252
9. Firesmith, D. (2004). Prioritizing Requirements, Journal of Object Technology, 3(8), pp.35–47.
10. Hudaib, A., Masadeh, R., Haj Qasem, M., & Alzaqebah, A. (2018). Requirements Prioritization Techniques. Modern Applied Science, 12(2), 3283–3285. https://doi.org/10.1007/978-3-642-04425-0_55
11. J. A. Khan, I. U. Rehman, Y. H. Khan, I. J. Khan, S. Rashid, 2015. Comparison of requirement prioritization techniques to find best prioritization technique, International Journal of Modern Education and Computer Science, vol. 7, no. 11, pp. 53.Berander P., P. Jönsson, Hierarchical Cumulative Voting (HCV) - Prioritization of Requirements in Hierarchies, Int'l J of SEKE. 16(6) 2006, pp. 819-849.
12. F. Sher, D. N. A. Jawawi, R. Mohamad and M. I. Babar, 2014. Requirements prioritization techniques and different aspects for prioritization a systematic literature review protocol, In: 2014 8th. Malaysian Software Engineering Conference (MySEC), Langkawi, 2014, pp.31-36. doi: 10.1109/MySec.2014.6985985
13. Vestola, M. (2010). "A Comparison of Nine Basic Techniques for Requirements Prioritization", Helsinki University of Technology, Helsinki (2010).

14. Ma, Q. (2009). The effectiveness of requirements prioritization techniques for a medium to large number of requirements: a systematic literature review. Diss. Auckland University of Technology, (November). Retrieved from http://aut.researchgateway.ac.nz/bitstream/10292/833/3/MaQ.pdf
15. Narendhar, M., & Anuradha, K. (2016). Different Approaches of Software Requirement Prioritization. International Journal of Engineering Science Invention, 5(9), pp.38–43.
16. Kano, Noriaki; Nobuhiku Seraku; Fumio Takahashi; Shinichi Tsuji (April 1984). Attractive quality and must-be quality. Journal of the Japanese Society for Quality Control (in Japanese). 14 (2): 39–48. ISSN 0386-8230.
17. Violante, M. G., & Vezzetti, E. (2017). Computers in Industry Kano qualitative vs quantitative approaches : An assessment framework for products attributes analysis. Computers in Industry, 86, pp.15–25. https://doi.org/10.1016/j.compind.2016.12.007
18. Lee, Y., & Huang, S. (2009). Expert Systems with Applications A new fuzzy concept approach for Kano' s model. Expert Systems With Applications, 36(3), 4479–4484. https://doi.org/10.1016/j.eswa.2008.05.034
19. Walden, D. (1999). CENTER FOR QUALITY OF M ANAGEMENT From the Chairman of the Editorial Board Introduction to Kano' s Methods, (617).
20. Berger, Charles; Blauth, Robert; Boger, David; Bolster, Christopher; Burchill, Gary; DuMouchel, William; Pouliot, Fred; Richter, Reinhard; Rubinoff, Allan; Shen, Diane; Timko, Mike; Walden, David. "Kano's Methods for Understanding Customer-defined Quality", In: Center for Quality Management Journal, Vol. 4 (Fall 1993), pp. 3 - 36.
21. William DuMouchel, 1993. "Thoughts on Graphical and Continuous Analysis" on"Kano's Methods for Understanding Customer-defined Quality", Center for Quality of Management Journal, Fall 1993
22. Joachim Karlsson and Kevin Ryan. 1997. A Cost-Value Approach for Prioritizing Requirements. IEEE Softw. 14, 5 (September 1997), 67-74. DOI=http://dx.doi.org/10.1109/52.605933
23. Kaur, G., & Bawa, S. (2013). A Survey of Requirement Prioritization Methods. International Journal of Engineering Research & Technology (IJERT), 2(5), 958–962.
24. R. Qaddoura, A. Abu-Srhan, M. H. Qasem and A. Hudaib, "Requirements Prioritization Techniques Review and Analysis," 2017 International Conference on New Trends in Computing Sciences (ICTCS), Amman, 2017, pp. 258-263. doi: 10.1109/ICTCS.2017.55
25. Google play, (2018). Retrieved fromhttps://play.google.com/store/apps.