Decision Support for Requirements Prioritization Using Data Analysis

[1]Hadeel E. Elsherbeiny, [2]A. A. Abd El-Aziz, [2]Nagy Ramadan
[1]Center of Environmental Planning and Development, INP, Cairo, Egypt
[2]Information Systems and Technology Department, ISSR, Cairo University, Giza, Egypt
hadel.ahmed@inp.edu.eg, a.ahmed@cu.edu.eg, nagyrd@hotmail.com

Abstract

Requirements Engineering (RE) is a very critical phase in System Development Life Cycle (SDLC). There are several attempts to enhance the quality and performance of this phase and therefore enhancing the industry of producing software systems. One of the most important issues in requirements is prioritizing them. Every software system has limited budget and time but stakeholders have large number of requirements so, there is a big need to prioritize those requirements to implement the most essential functions and get stakeholders' satisfaction. In this paper, we are aiming to prioritizing requirements using statistical analysis for large scale software systems and supporting requirements engineers in taking decisions about what to implement first. Large scale here, meaning software systems with large number of stakeholders.

Keywords: *Requirements Engineering, System Development Life Cycle, SPSS, Prioritization.*

1. Introduction

Requirements Engineering (RE) is the first phase in software development lifecycle; it is the process of capturing requirements from stakeholders. It is the most critical and important phase in the development process as it is hard to decide what precisely should be built [1]. RE is the science which facilitate developing the right software for the stakeholders and getting their satisfaction as the degree of success of any software systems depends on to which extent it meets the intended purpose[2,3]. By having agreed requirements, later phases of SDLC can be start. Any error in this phase may enforce developers to repeat their work and therefore increasing the cost.

The Standish Group surveyed over 350 companies about their over 8000 software projects, to find out how well they were faring. The results are sobering. Thirty-one percent of the software projects were canceled before they were completed. Moreover, in large companies, only 9% of the projects were delivered on time and within budget; 16% met those criteria in small companies similar results have been reported since then; the bottom line is that developers have trouble delivering the right system on time and within budget. Standish asked the survey respondents to explain the causes of the failed projects. The main factors were reported to be; Incomplete requirements, Lack of user involvement, Unrealistic expectations, Lack of executive support, Changing requirements and specifications and Lack of planning system. Some part of the requirements elicitation, definition, and management process is noticed to be involved in almost all of these causes [4]. Boehm and Papaccio reported that if it costs $1 to find and fix a requirements- based problem during the requirements definition process, it can cost $5 to repair it during design, $10 during coding, and $20 during unit testing, and as much as $200 after delivery of the system! So it pays to take time to understand the problem and its context and to get the requirements right the first time [5]**.**The process of RE consists of four sub-phases as shown in Fig.1:

1- Elicitation: collecting and elaborating requirements from stakeholders.
2- Analysis: analyzing and modeling the initial set of requirements that have been elicited, requirements engineer can return back to the elicitation phase to be sure that he has analyzed all elicited requirements.
3- Specification: Documenting the requirements, the output of RE process.
4- Validation: ensure that the specification matches user requirements, so the arrow is pointing to the elicitation phase. [4].

After validating the specification, Software Requirements Specification (SRS), the output of RE process, will be generated which acts as a contract between the stakeholders.

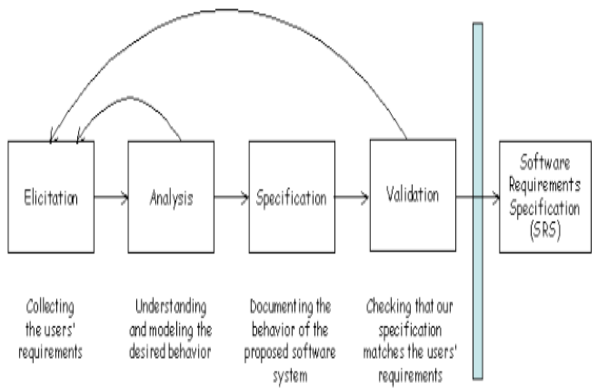

Fig.1. Requirements Engineering Process

Requirements elicitation and analysis which concerning with capturing and understanding requirements from stakeholdersis one of the most important and critical activities in RE phase especially in large scale software projects [6]. Now, There are large scale software projects, meaning here by large scale; projects with large number of stakeholders.

Stakeholders often have different, conflicting and non-organized requirements[7]. Requirements engineer is responsible for organizing and solving conflicts between those requirements. There is a need to prioritize those requirements to implement the most significant ones by the earliest product releases. In 2012, Soo Ling Lim and Anthony Finkelstein used social network to prioritize requirements and collaborative filtering (KNN algorithm) to predict requirements to stakeholders. They used weka tool to run KNN algorithm on the data set. In this work, we will use statistical analysis SPSS to prioritize the same dataset and get the correlation between requirements to predict the ratings of the stakeholders.

SPSS Statistics can take data from almost any type of file and use them to generate tabulated reports, charts, and plots of distributions and trends, descriptive statistics, and complex statistical analyses.

SPSS Statistics makes statistical analysis more accessible for the beginner and more convenient for the experienced user. Simple menus and dialog box selections make it possible to perform complex analyses without typing a single line of command syntax. The Data Editor offers a simple and efficient spreadsheet-like facility for entering data and browsing the working data file [8].

2. Related Work

Donald Fire smith defined the term of "priority", listed the purpose of requirements prioritization and its benefits. He defined the challenges and risks which face requirements engineers while prioritizing requirements. Various prioritization techniques are also defined. Then the reasons of naming it a critical phase are listed [9].

Qiao Ma presented a Systematic Literature Review on the effectiveness of requirements prioritization techniques on medium to large numbers of requirements. It is found that the strength of evidence for effectiveness is weak for most prioritization techniques for large numbers of requirements. She thought that more studies on prioritization techniques for large numbers of requirements are needed. For medium sized numbers of requirements, the techniques were more mature. She suggested a future work as further investigating the identified prioritization techniques for large numbers of requirements to get more evidence on the effectiveness of these techniques [10].

Lim and Finkelstein proposed Stake Rare method which helps projects with large scale stakeholders to elicit requirements accurately using social network and collaborative filtering.

Stake Rare addresses and solves three problems in eliciting requirements:

1- Inadequate stakeholder input.
2- Overloading stakeholders with information.
3- Biased prioritization of requirements.

To avoid the problem of inadequate stakeholder input; StakeRare uses StakeNet which builds a social network by asking each stakeholder to recommend other stakeholders and prioritizes them using social network measures.

To ovoid overloading stakeholders with information; StakeRare uses collaborative filtering to present only requirements that are relevant to them. It asks each stakeholder to rate an initial list of requirements and based on his rating it identifies a neighborhood for each stakeholder.

To avoid biased prioritization of requirements; StakeRare produces a prioritized list of requirements based on each stakeholder's rating and their influence in the project.

Our work uses the same dataset which collected in this paper. We have used statistical analysis to prioritize requirements which are elicited from 76 stakeholders. In this paper there are three elicitation methods; RateP, RankP, PointP. **In RateP**, Respondents rate a predefined list of requirements, from 0 (not important)to 5 (very important), and –1 for requirements they actively do not want. Respondents are also asked to add requirements not in the predefined list and rate those requirements**. In RankP**, Respondents provide their requirements with numeric priorities (1 for most important) and X for requirements they actively do not want. **In PointP**, Respondents are allocated 100 points to distribute among the requirements they want from RateP. The requirements include both the predefined ones and the additional ones they provide. Respondents are asked to allocate more points to the requirements that are more important to them.In this research, we have used the dataset of RateP elicitation method as RateP received high ratings from respondents when asking them about the elicitation method they preferred most. We have prioritized the requirements using statistical analysis and got spearman correlation coefficient between requirements to predict the ratings of unrated requirements [7].

Khari and Kumar took a closer look at six prioritization techniques and put them in a controlled experiment with the objective of understanding differences regarding ease of use, total time taken, scalability, accuracy, and total number of comparisons required to make decisions. These five criteria combined will indicate which technique is more suitable. The result from the experiment shows that Value oriented Prioritization (VOP) yields an accurate result, can scale up, and requires the least amount of time. VOP is supposed to be the best prioritization method according to the result of the experiment. They ordered the six prioritization techniques according to these five criteria. The worst candidate according to result is NAT [11].

Achimugu et al. identified and analyzed existing prioritization techniques in the context of the formulated research questions. They discovered that, although there is a lot of existing prioritization techniques, they still suffer from a number of limitations which includes: lack of scalability, methods of dealing with rank updates during requirements evolution, coordination among stakeholders and requirements dependency issues so, improvements still required. Also, the applicability of existing techniques in complex and real setting has not been reported yet. Some studies that dealt with the method of enhancing existing prioritization techniques were identified and synthesized [12].

[Khan et al., 2015] described an assessment of different requirements prioritization techniques and concluded that AHP is the best of them. They provide an examination of different requirement prioritization methods: analytic network process (ANP), analytic hierarchy process (AHP), hierarchy AHP, spanning tree matrix, bubble sort, binary search tree and priority groups. They categorized the requirement prioritization techniques from a user's perspective according to a number of criteria such as ease of use, required completion time, reliability of results [13].

[Khan et al., 2016] studied limitations and problems of some requirements prioritization techniques (binary search tree, AHP, hierarchy AHP, spanning tree matrix, priority group and bubble sort). They identified some problems and limitation in these techniques. None of the requirements prioritization techniques prioritize dependent requirements and performance of requirements prioritization techniques is not good. Therefore, new technique for prioritizing dependent and independent requirements is developed that is known as ANP. They conducted an experiment to evaluate the performance of ANP against existing requirements prioritization techniques (binary search tree, AHP, hierarchy AHP, spanning tree matrix, priority group and bubble sort). Experiment proves that main advantage of ANP is the prioritization of dependent requirements [14].

Table 1. Summary of Related Work

Author	Method	Result
Donald Firesmith(2004)	Definitions and discussion about requirements prioritization.	Defining the meaning, importance, risks, challenges and various techniques of requirements prioritization.
Qiao Ma(2009)	Systematic Literature Review on effectiveness of prioritization techniques for a medium to large number of requirements.	The effectiveness of Prioritization techniques is weak for software systems with large number of requirements.
Lim and Finkelstein(2012)	Social network and Collaborative filtering.	Identify and prioritize stakeholders and their requirements for large scale software systems.
Khari and Kumar(2013)	Characterization and evaluation the six prioritization techniquesaccording toease of use, total time taken, scalability, accuracy, and total number of comparisons required to make decisions.	Value Oriented Prioritization (VOP) yields an accurate result, can scale up, and requires the least amount of time.
Achimugu et al. (2014)	Identification and analysis of the existing prioritization techniques in the context of the formulated research questions.	Existing prioritization techniques still suffer from a number of limitations which includes: lack of scalability, methods of dealing with rank updates during requirements evolution, coordination among stakeholders and requirements dependency issues so, improvements still required
Khan et al.(2015)	Assessment and Examination of different requirements prioritization techniques.	AHP is the best requirements prioritization technique amongst all the requirements prioritization techniques.
Khan et al.(2016)	Evaluation of seven software requirements prioritization based on a case study.	ANP is the most successful prioritization technique.

3. The Proposed Technique

We have used the same dataset which used in [Lim and Finkelstein, 2012]. There are three elicitation methods used while eliciting requirements from stakeholders which are RateP, RankP and PointP. In this paper, we are interested in RateP method. In RateP, Respondents rate a predefined list of requirements, from 0 (not important) to 5 (very important), and –1 for requirements they actively do not want. The predefined list is collected using the normal methods of eliciting requirements; Interviews, Questionnaire, Brainstorming, Focus Group, etc. Respondents also were asked to add requirements which are not exist in the predefined list and rate them. The dataset consists of 76 stakeholders, 10 project objectives, 48 requirements and 104 specific requirements. Every project objective consists of several requirements and every requirement consists of several specific requirements. We have noticed that every project objective with its requirements and specific requirements took the same rate, so every project objective represent its requirements and specific requirement.

3.1 Methodology

In this section, we present the suggested steps for prioritizing requirements.

Input→ list of non-prioritized requirements obtained from stakeholders.

Output→ Recommended Prioritized list of requirements.

Begin

1- Give a symbol for each requirement.
2- Get the frequency for each requirement(requirements which stakeholders gave rate).
3- Compute threshold value for requirements frequencies.
4- Exclude requirements which got frequencies less than the threshold value.
5- Get the mean of the ratings for each requirement.
6- Prioritize the remaining requirement.
7- Compute the correlation coefficient between requirements to get the association between them.

End

3.2 Experiment and Results

We have used the same dataset which used in [Lim and Finkelstein, 2012]. There are three elicitation methods used while eliciting requirements from stakeholders which are RateP, RankP and PointP. In this paper, we are interested in RateP method. In RateP, Respondents rate a predefined list of requirements, from 0 (not important) to 5 (very important), and –1 for requirements they actively do not want. The predefined list is collected using the normal methods of eliciting requirements. Respondents also were asked to add requirements which are not exist in the predefined list and rate them. The dataset consists of 76 stakeholders, 10 project objectives, 48 requirements and 104 specific requirements. Every project objective consists of some requirements and every requirement consists of some specific requirements. In this analysis, we found that stakeholders gave each requirement and its specific requirement the same rate of their project objective.

Step1. We got the requirements with high frequencies and ignored those which got frequencies less than 41 after computing global thresholding. In table (2), frequencies and percentages for each requirement is presented.

Table 2. Frequencies and percentages for each requirement

Req.	d	c	g	a	h	b	j	i	f	e
Freq.	75	74	74	73	65	20	17	15	13	7
%	98.7	97.4	97.4	96.1	85.5	26.4	22.4	19.7	15.8	10.5

Step2. After ignoring requirements with frequencies less than 41, we prioritized the remaining requirements based on stakeholders' ratings. In table (3) and Fig (2), mean rate for the selected requirements, requirements with high ratings, is presented.

Table 3. Mean rate for the selected requirements

Req.	a	d	g	c	h
Rating (Mean)	4.71	4.66	4.59	4.53	3.41

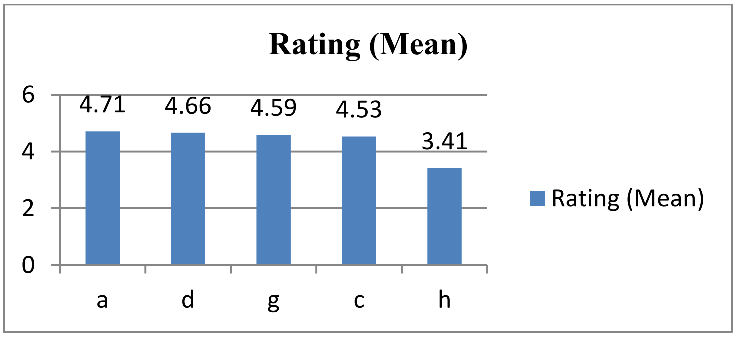

Fig. 2. Mean rate for the selected requirements

Step3. The correlation coefficient between the requirements is computed to get the association between the requirements so that we can predict the rating of a requirement if a stakeholder doesn't rate based on his other ratings. In table (4), the correlations between the selected requirements are presented.

Table 4. The Spearman Correlations between the selected requirements

			a	c	d	g	h
Spearman's rho	a	Correlation Coefficient	1.000	.392**	.409**	.246*	.135
		Sig. (2-tailed)	.	.001	.001	.048	.318
		N	66	66	66	65	57
	c	Correlation Coefficient	.392**	1.000	.442**	.265*	.148
		Sig. (2-tailed)	.001	.	.000	.030	.263
		N	66	68	68	67	59
	d	Correlation Coefficient	.409**	.442**	1.000	.342**	.196
		Sig. (2-tailed)	.001	.000	.	.005	.137
		N	66	68	68	67	59

	g	Correlation Coefficient	.246*	.265*	.342**	1.000	.197
		Sig. (2-tailed)	.048	.030	.005	.	.135
		N	65	67	67	68	59
	h	Correlation Coefficient	.135	.148	.196	.197	1.000
		Sig. (2-tailed)	.318	.263	.137	.135	.
		N	57	59	59	59	59

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed)

It is clear that there is a significant correlation between all requirements except h which is not significant with all requirements.

We can predict that when a stakeholder gives high rating to a specific requirement, he will give also a high rating to the requirements which are positive correlated with. If there is a negative correlate; this will mean that if a stakeholder gives high rating to specific requirement, we will predict law rating to the requirement.

4. Conclusion

RE is one of the most important phases in SDLC as it's hard to decide what precisely to build. Every software system has limited time, budget and resources. Requirements prioritization is needed to begin implementing the most important functions first and therefore gaining stakeholders' satisfaction. In this paper, statistical tools as descriptive statistics and spearman correlation coefficient are used to prioritize requirements for software systems with large stakeholders. Statistical packages are easy to use and accurate way to obtain the results. In the future work, we can compare between the results obtained by statistical methods and other methods.

References

[1]. Amira A. Alshazlya , Ahmed M. Elfatatrya and Mohamed S. Abougabalb., "Detecting defects in software requirements specification", Alexandria Engineering Journal. Vol. 53, 513–527. 2014.

[2]. VerónicaCastañed, Luciana Ballejo, Ma. Laura Caliusco and Ma. Rosa Galli, "The Use of Ontologies in Requirements Engineering". Global Journal of Researches in Engineering, Vol. 10, Issue 6 (Ver 1.0). November 2010.

[3]. Elizabeth Bjarnason." Integrated Requirements Engineering –Understanding and Bridging Gaps within Software Development".DoctoralDissertation.Department of Computer Science Lund University.2013.

[4]. HadeelElsherbeiny, A. A. Abd El-Aziz and Nagy Ramadan . "A Survey on Attempts to Enhance Requirements Engineering Process", CIIT, 2016.

[5]. Shari Lawrence Pfleeger and Joanne M. Atlee "*Software Engineering Theory an Practice*", Fourth Edition, Pearson Education. 2010.

[6]. BeniSuranto, "Software Prototypes: Enhancing the Quality of Requirements Engineering Process", IEEE. 2015.

[7]. Soo Ling Lim and Anthony Finkelstein." StakeRare: Using Social Networks ndCollaborative Filtering for Large-Scale Requirements Elicitation". IEEE Transactions on Software Engineering,2012.

[8]. IBM Corporation, "IBM SPSS Statistics 21 Brief Guide", 2012.

[9]. Donald Firesmith, "Prioritizing Requirements", Journalof Object Technology, Vol. 3, No.8, September-October 2004.

[10]. Qiao Ma, "The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements: A Systematic Literature Review", Master thesis, Auckland University, November 2009.

[11]. ManjuKhari and Nikunj Kumar, "Comparison of Six Prioritization Techniques for Software Requirements", Journal of Global Research in Computer Science, Volume 4, No. 1, January 2013

[12]. Philip Achimugu, Ali Selamat, Roliana Ibrahim andMohdNaz'riMahrin "A systematic literature review of software requirements prioritization research", Information and Software Technology, 2014.

[13]. Javed Ali Khan,Izaz Ur Rehman,Yawar Hayat Khan,IftikharJaved Khan and Salman Rashid. "Comparison of Requirements Prioritization Techniques to Find Best Prioritization Technique", I.J. Modern Education and Computer Science. 2015.

[14]. Javedali Khan,Izaz-ur-Rehman,Shah Poor Khan, IqbalQasim and Yawar Hayat Khan. "An Evaluation of Requirement Prioritization Techniques with ANP", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 7, 2016.