A Quantitative Study to Identify Critical Requirement Engineering Challenges in the Context of Small and Medium Software Enterprise

Souhaib Besrour[1], Lukman Bin AB Rahim[2] and P. D. D. Dominic[3]
Department of Computer and Info. Sciences
Universiti Teknologi PETRONAS,
31750 Tronoh, Perak, Malaysia.
[1]souhaib_g02379@utp.edu.my, [2]lukmanrahim@petronas.com.my, [3]dhanapal_d@petronas.com.my

***Abstract*—Requirement engineering field has been intensively studied in last decade. Even though requirement still one of the most critical process in software development. Recent studies shows that 56% of system defects are coming from requirement. Additionally, by one estimate requirement errors cost 10 times more than coding errors. Thus, there is a need to identify those critical requirement engineering challenges. To address this issues, critical challenges have been investigated in the last two decades between the year 1995 and 2016. Where systematic literature review (SLR) has been used to explore previous studies findings and results. Additionally, identified critical challenges have been further investigated using quantitative study through software small and medium enterprises (SME). Where a total of 250 surveys have been collected, after eliminating incomplete surveys 217 completed surveys used for data analysis. After performing all process 12 challenges has been specified as critical challenges in the context of software SME. Where identified critical challenges can benefit software enterprises particularly new enterprises to pay further attention to specified critical challenges.**

Keywords—Requirements challenges, Requirements engineering, Software engineering, Requirement analysis.

I. INTRODUCTION

According to the Standish Group (2013), 39% of developed software solutions were successful. In today's world, software industry is becoming more sophisticated than ever before. Today, more and more companies and organisations consider requirement engineering as one of the principal challenges in system and software development. According to Sankhwar[1], requirement errors are very costly and considered as the most critical issue for the whole software system's life-cycle. Thus, improving the requirement engineering process appears very critical for future software industry success [1-3]. According to Bano[4]and Lehtola [5], studying critical requirement challenges is a significant process which guides software companies through their journey by having a prior knowledge about possible mistakes that can effect developed project. There are few studies in literature that report requirement engineering issues including the following studies [4, 6-12]. However, these studies have some common issues including the following points. First, low number of identified challenges typically happen when the study does not cover all requirement engineering process. Second, low sample size which reflect that achieved results cannot be generalized and used in other organizations. Third, the study designed to covers large scale companies only, reflecting that study results are not applicable in software SME. Moreover, there was no any study conducted in the context of the requirement engineering challenges by practitioners working in software companies in Malaysia. Karlsson [10] and Asghar [6], reported that identifying critical requirement engineering challenges help small and medium software enterprises avoiding unnecessary mistakes which requires additional time to fix and have a negative effects on project delivery time[13, 14]. Moreover occurred requirement engineering mistakes needs further work, resources, effort and additional expenditure to fix [8, 15-20].

II. METHODOLOGY

A. Systematic Literature Review

In this study Systematic literature review (SLR) has been used as primary methodology. Where SLR is the structured pattern of recording the information extracted from the literature [21, 22]. In this study systematic literature review has been conducted in the last two decade in the period between the years 1995 and 2016 as presented in Table I. The systematic literature review primary aim is to identify critical requirement challenges which have negative effect on software industry. Where, systematic literature review aim to explore the maximum pertinent literature using search keywords as presented in Table II. The primary benefit from this process is to explore previous conducted study and to benefit from previous researches findings.

Table I. SLR Articles With Respect To Years

Year	1995-2000	2001-2005	2006-2010	2011-2016	Total
Number of publication	10199	26804	79025	81640	168,680

 978-1-5090-2549-7/16/$31.00 ©2016 IEEE

Table II. SLR Primary Keywords and Alternative Words

Primary Keywords	Alternative words
Requirement	"Needs" OR "demands" OR "request" OR "Matter" OR "Terms" OR "requisite" OR "necessities"
Challenge	"Issues" OR " Trouble" OR "Affair" OR "Obstacle" OR "Disagreement" OR "Challenges" OR "Difficulties" OR "Hitches" OR "Harms" OR "Glitches" OR "Complication"
Software	"computer application" OR "computer program" OR "system" OR "Software system" OR "Software application" OR "Computer system"
Assessment	"Evaluation" OR "estimate" OR "judgment" OR "rating" OR "valuation"
Software companies	"Software industry" OR "global software industry" OR " Software enterprises" OR "SME" OR "Small and Medium" OR "Software firms" OR "Software corporations" OR "Software businesses"
Requirement processes	"Requirement stages" OR "Requirement processes" OR "Software requirement processes" OR "Requirement components"

III. QUANTITATIVE STUDY

Quantitative study has been used as methodology in this study. The most used approach in quantitative method is survey which is more appropriate in data collection for large scale population[23]. Additionally, majority of conducted studies in the area of requirement engineering use quantitative method in data collection process. Surveys is an instruments for collecting information and data from subjects to describe their experience, knowledge, attitude and perspective. Kitchenham[21] report that surveys is the most used research instrument world-wide. On the other hand, specified targeted population are software SME which have a direct interaction with software development including web development, software development, mobile application and outsourcing. This section provides a detailed information of the survey instrument, which includes participants' selection process, the unit of analysis, questionnaire design, and survey approach.

A. Participants' Selection Process.

The surveys were distributed to practitioners who works in software SME in four locations: Kuala Lumpur, Selangor, Johor and Penang. Software enterprises have been selected from the Multimedia Super Corridor (MSC) software company directory and Enterprise Malaysian websites which are available at (*http://www.mtdc.com.my/tenants.php). Additionally, few other websites* have *been used during* companies identifying process including: job seekers (*http://www.jobstreet.com.my*), linked-in (*www.linkedin.com*); (*http://www.monster.com.my);* (*http://my.jobsdb.com/my*), internet search engines (*www.google.com*) and IT researchers and experts in the area of software engineering have been consulted during respondents selection process.

B. Survey Unit Of Analysis

The analysis unit in this research survey was Malaysian software SME. The study population included enterprises which have a direct interaction with software development including web development, software development, mobile application and outsourcing. Lastly, performing this study encompasses interacting with software industry practitioners including software developers, software testers, directors, project team leaders and software project managers.

C. Questionnaire Design

Data collection instrument for this study is research survey. It has been used in this study to validate identified critical requirement challenges. Additionally, this method can help this research study reduce biasness when interpreting quantitative data. A new set of questionnaire has been developed from scratch for the survey, which is a common practice in software engineering research. The questionnaire composes of three primary sections to collect required information. The first survey section is background information about the companies. The second section contains respondent's information including their education level, experience and current position. The last section contains critical requirement engineering challenges which have to be rated and assessed based on software practitioner perspective.

D. Survey Approach

The quantitative method instrument in this study is the survey approach. This approach allows information collection from targeted software practitioners who have industrial experience. To address this empirical study target, 217 surveys have been distributed across Malaysian software SME. The survey data collection process covered a period of 7 month (spanning from January to July 2015). Two primary statistical methods have been used to test collected quantitative data: descriptive statistics and frequent analysis.

IV. RESULT

A. Demographic Information

It is significant to highlight respondents' demographic information in software industry. Table III presents summary of software SME demographic information. The total number of respondents is 217 respondents where 66.3% of respondents are male while 33.6% of respondents presents female. A further detail about respondents' positions is presented in the following section.

Table III. Respondents Demographics

Respondents		Freq.	Valid %
Gender	Male	144	66.3
	Female	73	33.6
Position	Project Manager	19	8.7
	Analyst	16	7.3
	Designer	18	8.2
	Tester	17	7.8
	Programmer	121	55.7
	Other	26	11.9
Working Experience	Less Than a Year	55	25.3
	between 1-2 years	42	19.3
	between 2-3 years	28	12.9
	between 3-5 years	30	13.8
	above 5 years	62	28.5

B. Position

It is significant to highlight the respondent's position where survey has been conducted across software SME. Respondent's positions includes: programmers, analysts, designers, testers and project managers as depicted in Table III. Where121 respondents were programmers, representing the highest number among respondents by 55.7% of the total respondents. Other positions were analysts and designers with 16 and 18 representing 7.3% and 8.2%, respectively. The rest of the positions were testers and project managers, scoring 17 and 19, representing 7.8% and 8.7%, respectively as presented in Figure 1.

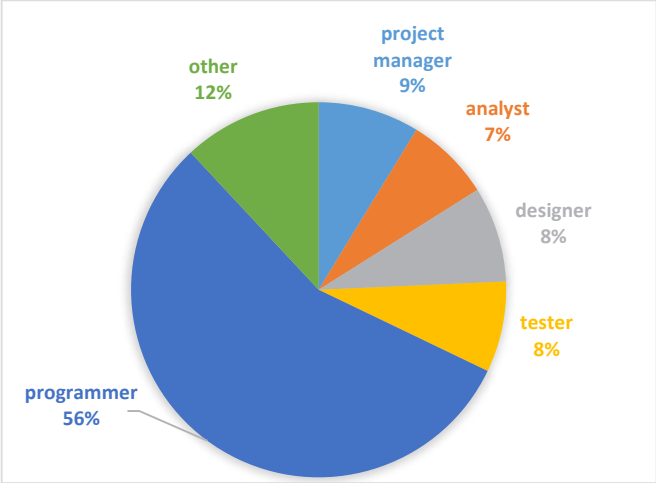

Fig .1. Respondent's Positions

C. Working Experience In The Software Industry

It was significant to consider the working experience of the respondents in the software industry. Respondents working experience includes: less than a year, between 1-2 years, between 2-3 years, between 3-5 years, above 5 years as depicted in Table III. In performed survey, 28 respondents had experience between 2-3 years representing 12.9%. Whilst, 42 respondents had experience between 1-2 years. In contrast, 55 respondents had experience less than a year representing 25.3% of the respondents. On the other hand, 62 respondents had experience more than 5 years representing 28.5% of the respondents. Lastly, 30 respondents had experience between 3-5 years representing13.8% as presented in Figure 2.

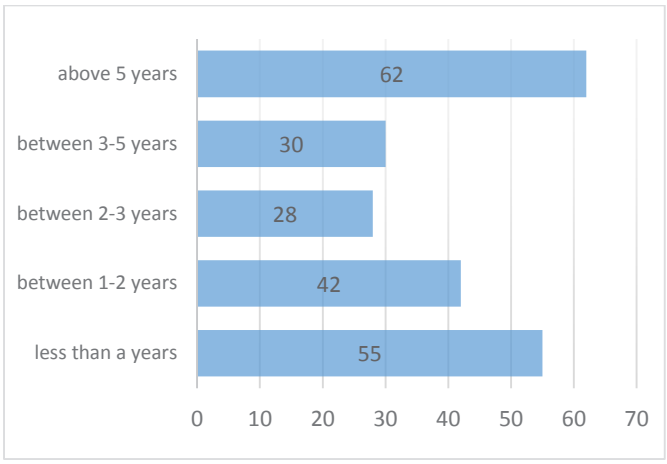

Fig.2. Respondents Working Experience

D. Critical Requirement Challenges

After systematic literature review study has been conducted through the last 2 decade in the context of requirement engineering. A total of 24 critical requirement challenges have been identified as presented in Table IV. The next step was to verify identified challenges using empirical survey, in order to validate the critical requirement engineering challenges that have a negative effect on software companies. Where a total of 217 respondents have been involved in data collection. Note that, majority of the conducted researches in the area of software engineering in the literature use Likert scale level to assess respondents feedback [24, 25]. Thus, a five Likert scale survey has been used to measure the significance of requirement challenges as follow (strongly disagree, disagree, neutral, agree, and strongly agree). The 217 respondents have to rate each challenge base on their experience and software industry perspective. The primary goal of identifying those challenges is to help software companies particularly, less experience companies to avoid those critical challenges. Also, it help software SME to have extra awareness about possible problem that may face in future projects. Thus, identifying critical requirement engineering challenges enhance requirement engineering process by ovoid unnecessary problems.

Table IV. Score Summary of Identified Critical RE Challenges

No	Challenge	Level
1.	Poor stakeholders' identification	1.32
2.	Poor understanding of complex requirement	4.08
3.	Lack of ensuring stakeholders' satisfaction of the requirements	1.87
4.	Collecting vague and ambiguous requirement	4.59
5.	Requirement incorrectness	1.97
6.	Poor Modeling of Functional requirements	4.06
7.	Poor requirement traceability	2.23
8.	Poorly defined specification	4.07
9.	Inconsideration of social and organisational issues	1.97
10.	Requirements Inconsistency	4.26
11.	Sophisticated relation between entities	1.42
12.	Poor understanding of non-functional requireents	4.01
13.	Unprioritized Requirements	1.69
14.	Poor communication during requirement elicitation	4.13
15.	Unclear user preference	2.45
16.	Undocumented Functional requirements and non-functional requirements	4.29
17.	Low idea quantity generation	1.87
18.	Incomplete Requirements	4.08
19.	Poor understanding of domain constraints	1.86
20.	Undocumented relations among requirements and stakeholders	4.16
21.	Ambiguity of defining system scope and boundary	1.78
22.	Undefined functional and non-functional requirements and system constraints	4.03
23.	Lack of evident definitionn	2.17
24.	Poor addressing of security risks in requirements.	4.18

V. DISCUSSION

This section discuss identified critical requirement engineering challenges scores, in order to have better understanding about their support level. A five Likert scale survey has been used to measure the significance of requirement challenges as follow: 1=strongly disagree, 2=disagree, 3=neutral, 4= agree, and 5=strongly agree. Where challenges rated 4.0 and above is considered as critical challenges. Out of 24 identified challenges 12 challenges have been rated as critical. CH4, Collecting vague and ambiguous requirement score 4.59 which is the highest score among all challenges. In contrast CH1, Poor stakeholders' identification scores 1.32 which is below acceptance score of 4.0. Thus CH1 is not considered as critical challenges. On the other hand, CH24, poor addressing of security risks in requirements score 4.18 which is above acceptance score of 4.0. Thus CH24 is considered as critical requirement challenges based on software industry practitioners. CH8 and CH20 scores 4.07 and 4.16 respectively which reflect a high score comparing to other challenges. Table IV summarize the 24 challenges score summary.

VI. CONCLUSION

In conclusion, requirement issues have a negative effect on software project. This study aim to investigate critical requirement issues facing software SME. In order to give insight to software companies particularly companies who have less experience about requirement challenges that may face during the journey of software development. Where software engineering academic resources have been investigated through the last two decade between the years 1995 and 2016 to identify critical challenges in the context of requirement engineering. Additionally, identified challenges have been further investigated by 217 practitioners from Malaysian software industry. 24 challenges have been identified using systematic literature review. However after performing validation process out of 24 challenges 12 challenges have been rated as critical challenges. Identified challenges aim to benefit software small and medium enterprises particularly less experience companies to pay further attention to identified challenges. Lastly, a further study will be performed on order to identify possible solutions for critical requirement engineering challenges in software SME.

REFERENCE

[1] S. Sankhwar, V. Singh, and D. Pandey, "Requirement Engineering Paradigm," *Global Journal Of Multidisciplinary Studies,* vol. 3, 2014.

[2] M. Nawaz, "Review Analysis on Requirement Elicitation and its Issues," 2015.

[3] U. I. Janjua, A. Oxley, and J. B. Jaffer, "Effective Risk Management of Software Projects (ERM): An Exploratory Literature Review of IEEE and Scopus Online Databases," in *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, 2014, pp. 445-452.

[4] M. Bano and N. Ikram, "Issues and challenges of requirement engineering in service oriented software development," in *Software Engineering Advances (ICSEA), 2010 Fifth International Conference on*, 2010, pp. 64-69.

[5] L. Lehtola, M. Kauppinen, and S. Kujala, "Requirements prioritization challenges in practice," in *Product focused software process improvement*, ed: Springer, 2004, pp. 497-508.

[6] S. Asghar and M. Umar, "Requirement engineering challenges in development of software applications and selection of customer-off-the-shelf (COTS) components," *International Journal of Software Engineering,* vol. 1, pp. 32-50, 2010.

[7] J. M. Bhat, M. Gupta, and S. N. Murthy, "Overcoming requirements engineering challenges: Lessons from offshore outsourcing," *Software, IEEE,* vol. 23, pp. 38-44, 2006.

[8] A. Birk and G. Heller, "Challenges for requirements engineering and management in software product line development," in *Requirements Engineering: Foundation for Software Quality*, ed: Springer, 2007, pp. 300-305.

[9] C. J. Davis, R. M. Fuller, M. C. Tremblay, and D. J. Berndt, "Communication challenges in requirements elicitation and the use of the repertory grid technique," *Journal of Computer Information Systems,* vol. 46, pp. 78-86, 2006.

[10] L. Karlsson, Å. G. Dahlstedt, B. Regnell, J. Natt och Dag, and A. Persson, "Requirements engineering challenges in market-driven software development–An interview study with practitioners," *Information and Software technology,* vol. 49, pp. 588-604, 2007.

[11] U. Sajjad and M. Q. Hanif, "Issues and Challenges of Requirement Elicitation in Large Web Projects," *School of computing, Blekinge institute of technology ronneby Sweden,* 2010.

[12] S. Sharma and S. Pandey, "Requirements elicitation: Issues and challenges," in *Computing for Sustainable Global Development (INDIACom), 2014 International Conference on*, 2014, pp. 151-155.

[13] S. Besroura and I. Ghani, "Reduce Security Threaten Using Misuse Case Scenarios," 2012.

[14] B. Souhaib and G. Imran, "Measuring Security in Requirement engineering," *International Journal of Informatics and Communication Technology (IJ-ICT) Vol,* vol. 1, pp. 72-81, 2012.

[15] Damian, Zowghi, and Didar, "RE challenges in multi-site software development organisations," *Requirements engineering,* vol. 8, pp. 149-160, 2003.

[16] J. Fernandes, E. Henriques, A. Silva, and M. A. Moss, "Requirements change in complex technical systems: an empirical study of root causes," *Research in Engineering Design,* pp. 1-19, 2014.

[17] Y. Tang, K. Feng, K. Cooper, and J. Cangussu, "Requirement engineering techniques selection and modeling an expert system based approach," in *Machine Learning and Applications, 2009. ICMLA'09. International Conference on*, 2009, pp. 705-709.

[18] S. Besrour, L. B. Ab Rahim, and P. Dominic, "The Study of Available Techniques for Existing Requirements Engineering Challenges Based on Literature Review Evidences," *Research Journal of Applied Sciences, Engineering and Technology,* vol. 8, pp. 2082-2091, 2014.

[19] S. Besrour, L. Bin Ab Rahim, and P. Dominic, "Assessment and evaluation of requirements elicitation techniques using analysis determination requirements framework," in *Computer and Information Sciences (ICCOINS), 2014 International Conference on*, 2014, pp. 1-6.

[20] S. Besrour and I. Ghani, "Questionnaire based Approach to Measure Security in Requirement Engineering," *International Journal of Computer Applications,* vol. 54, pp. 31-34, 2012.

[21] Kitchenham, Pearl Brereton, Budgen, Turner, Bailey, and Linkman, "Systematic literature reviews in software engineering–A systematic literature review," *Information and Software Technology,* vol. 51, pp. 7-15, 2009.

[22] M. A. T. Almomani, S. Basri, S. Mahamad, and A. O. Bajeh, "Software Process Improvement Initiatives in Small and Medium Firms: A Systematic Review," in *Advanced Computer Science Applications and Technologies (ACSAT), 2014 3rd International Conference on*, 2014, pp. 162-167.

[23] M. Borrego, E. P. Douglas, and C. T. Amelink, "Quantitative, qualitative, and mixed research methods in engineering education," *Journal of Engineering education,* vol. 98, p. 53, 2009.

[24] Canny and George, " ―5 point Vs. 6 point likert scale‖, (Online) Available: http://searchwarp.com/swa69773.htm," 2006.

[25] H. v. Laerhoven, H. v. d. Zaag-Loonen, and B. Derkx, "A comparison of Likert scale and visual analogue scales as response options in children's questionnaires," *Acta paediatrica,* vol. 93, pp. 830-835, 2004.