JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 33, 1519-1530 (2017)
DOI: 10.6688/JISE.2017.33.6.8

Significant Requirements Engineering Practices for Outsourced Mobile Application Development

JAVED IQBAL[1], RODINA AHMAD[2], MOHD HAIRUL NIZAM MD. NASIR[2] AND MUZAFAR KHAN[1]
[1]*Department of Computer Science*
COMSATS Institute of Information Technology
Islamabad, 44000 Pakistan
[2]*Faculty of Computer Science and Information Technology*
University of Malaya
Kuala Lumpur, 50603 Malaysia
E-mail: {javediqbal; muzafar_khan}@comsats.edu.pk; {rodina; hairulnizam}@um.edu.my

The magnitude of Software Development Outsourcing (SDO) is snowballing day by day. However, a considerable proportion of SDO projects cannot acquire the anticipated benefits. In most of such cases, the reasons are often associated to Requirements Engineering (RE) process. The same is true for Mobile Application Requirements Engineering (MARE) process in case of SDO. The aim of this study is to cultivate the MARE process for SDO by identifying and ranking the significant RE practices for the process. To identify the significant MARE practices, we have employed a literature-based list of RE practices. Two online questionnaire surveys have been conducted to complete the two rounds of Delphi method. After the first round, we have identified 86 significant MARE practices for SDO based on 36 responses from SDO experts. The results of the second round have helped us to provide the overall ranking of the practices. The SDO practitioners have termed the identified MARE practices beneficial to address the issues of MARE process in case of SDO and to achieve the outsourcing benefits.

Keywords: mobile application requirements engineering, software development outsourcing, requirements engineering practices, Delphi method, IT outsourcing

1. INTRODUCTION

Information Technology Outsourcing (ITO) is the process of transferring all or some of the software development and associated activities to an external service provider or vendor who performs these activities according to the agreed upon contract [1]. IT outsourcing has various forms. In case of Software Development Outsourcing (SDO), the client outsources all or some of the activities involved in software development lifecycle [2].

Mobile phones and tablets have reshaped the worldwide market [3]. Desire for being connected at anytime and anywhere has increased the usage of mobile applications [4]. The mobile devices market has grown rapidly during the last decade [5]. Nowadays more than 50% population of the world is utilizing mobile applications for the management of everyday activities [6]. Developing versatile mobile applications for large number of users, various operating systems and devices is challenging [7-10]. Furthermore, security breaches provide opportunities for attacking mobile phones and can cause financial losses to mobile application users [6].

Received March 17, 2017; revised July 4, 2017; accepted August 9, 2017.
Communicated by I-Hsien Ting.

Projects are outsourced for software development to achieve the anticipated benefits but only half of such projects are declared successful [11]. Unexpected staff turnover, inertia, unseen cost, scope creeping, implications of law, cultural issues and high tax rates are commonly occurring risks for the outsourced projects [12]. However, Requirements Engineering (RE) related risks are the major source of outsourced projects failure [13, 14]. RE is the most important phase during software development [15] as other development activities are also affected because of the RE [16]. As client and contractor(s) are at distance in case of outsourcing [17], therefore, RE issues are increased manifold [18]. Improper communication, differences in working timings, and knowledge management issues *etc.* augment the problems of RE process for SDO. Because of such obstacles, 40 percent off-shored projects are failed in achieving the anticipated benefits [19] and 50 percent companies attempting GSD cannot acquire the desired benefits [20]. These trends are even more alarming for outsourced mobile application development.

Therefore, to acquire the anticipated benefits of SDO in case of mobile application development outsourcing and to eliminate the issues of Mobile Application Requirements Engineering (MARE) process, significant MARE practices for SDO must be identified. Such RE practices should also be ranked so that they can be employed according to their importance [21]. With this context, this study intends to answer the following Research Questions (RQ):

RQ1: What are significant MARE practices in case of SDO?
RQ2: What is the ranking of the significant MARE practices in case of SDO?

2. RELATED WORK

Mobile application RE has been deliberated in several studies. The challenges of RE, in case of mobile applications for disabled, have been presented in [22]. The role of RE in the development of mobile phone industry has been discussed in [23]. A. Maccri [24] presents the challenges of RE in mobile telephone industry. The RE process for providing mobile services to blind users has been described in [25]. Another study [26] presents case study about the challenges of RE for mobile games development. The mobile commerce user requirements have been identified in [27, 28]. By applying linguistic rules, mobile feature requests have been detected from the users' reviews [29]. Through sentiment analysis, the evolution of mobile application requirements has been investigated in [30]. The study [31] presents a tool to elicit the requirements for mobile social software. The web application RE has been discussed in [32]. The security requirements for mobile device management system have been focused in [33] whereas security engineering process for web application has been discussed in [34].

Thus, several studies focus on the activities of RE process for mobile application development but to the best of our knowledge no study provides a comprehensive list of the significant MARE practices required for a fruitful MARE process in case of SDO. The objective of this research is to explore a comprehensive list of MARE practices for SDO and to rank such practices according to their perceived benefits for SDO.

3. RESEARCH METHODOLOGY

The list of general RE practices has been arranged by choosing RE practices recommended by Sommerville and Sawyer [35], and security RE practices from [36-42]. The practices to address communication, management and coordination, knowledge management, and cultural issues of RE process for SDO have been taken from [21, 43-57]. The practices have been denoted by P_1, P_2, P_3, …, P_{103} and presented in Appendix A. For further research, we have employed the Delphi method [58, 59].

Two online questionnaire surveys have been conducted to complete the two rounds of Delphi method. The questionnaires have been sent to SDO practitioners belonging to Malaysian and Pakistani software development companies. The respondents are project managers, system analysts, requirements engineers, technical managers and senior managers having at least 10 years experience of SDO, including mobile application development experience, as the basic criterion. Out of 200 identified practitioners, only 36 (T) respondents completed both the rounds.

3.1 First Online Questionnaire Survey

The respondents have been solicited to rank the given 103 RE practices according to perceived benefits of the practices for MARE process in case of SDO. Based on 10-point scale, the different categories of the perceived benefits have been formulated as suggested by [20, 60, 61]. The categories are; (i) High Perceived Benefits (H_i, 10): A practice has 'high perceived benefits' if it is mandatory and always used; (ii) Medium Perceived Benefits (M_i, 7): A practice has 'medium perceived benefits' if it is not mandatory but widely used; (iii) Low Perceived Benefits (L_i, 4): A practice has 'low perceived benefits' if it is used only for some particular projects; (iv) Zero Perceived Benefits (Z_i, 1): A practice has 'zero perceived benefits' if it is never or rarely used.

3.1.1 Criteria for the selection of the significant MARE practices for SDO

If according to the opinion of at least 50% participants, the perceived benefits of an RE practice fall in the 'high perceived benefits' and in the 'medium perceived benefits' categories then that practice is considered to be 'significant' for MARE process in case of outsourced software development. The similar method, using the criterion of considering the opinion of 50% or more respondents for decision making, has already been employed effectively in the preceding studies [61-63]. For each practice, the Prominence Level (PL) represents percentage of responses in 'high perceived benefits' and 'medium perceived benefits' categories, and is calculated as given in Eq. (1):

$$PL = \frac{[(H_i + M_i)]}{T} * 100. \quad (1)$$

Where H_i = No. of responses in high perceived benefits category,
M_i = No. of responses in medium perceived benefits category, and
T = Total number of responses considered for the analysis.

3.2 Second Online Questionnaire Survey

After the first survey, we have received back 40 responses. During the second survey, online questionnaires have been sent to those SDO professionals who responded during the first survey. For the second round, we have been successful in receiving back 36 responses that have been selected for the data analysis. The overall research methodology is shown in Fig. 1.

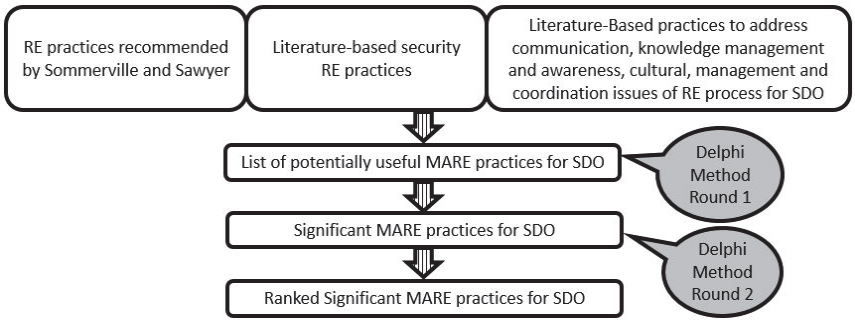

Fig. 1. Steps to identify and rank the significant MARE practices for SDO.

4. RESULTS AND DISCUSSIONS

The results obtained can be divided into two main categories *i.e.*, (i) the identification of significant MARE practices for SDO and (ii) the ranking of the significant MARE practices for SDO. The results are discussed in the following subsections.

4.1 Identification of the Significant MARE Practices for SDO

The significant MARE practices have been identified through the first round of Delphi method.

4.1.1 First round

By providing a literature-based list of 103 practices, we have requested the SDO practitioners for ranking the given practices weighing their benefits in case of SDO. By applying criterion for the significance, we have identified 86 significant MARE practices for SDO.

The PL of 50 or above proves that practice is significant to address the issues of MARE process in case of SDO. In case of the only 17 RE practices, the value of PL is less than 50. This means that according to the SDO practitioners' perception, these 17 RE practices are not significant to address the issues of MARE processes in case of SDO. The RE practices are P_2, P_3, P_{10}, P_{12}, P_{17}, P_{20}, P_{27}, P_{28}, P_{35}, P_{36}, P_{38}, P_{40}, P_{47}, P_{48}, P_{56}, P_{57} and P_{83}. The practices have been underlined and presented in Appendix A. In case of the rest of 86 RE practices, the value of PL is more than 50. All of the remaining 86 RE practices are, therefore, significant according to the defined criterion. This reveals that these practices are considered effective by the SDO industry practitioners to address the

issues of MARE process for SDO. The significant RE practices have also been shown in Appendix A which provides answer to RQ1 of this study.

4.2 Ranking of the Significant MARE Practices for SDO

The significant MARE practices have been ranked during the second round of Delphi method based on the results of first round.

4.2.1 Second round

Before starting the second round of Delphi method, average rankings and standard deviations were calculated in case of each significant MARE practice. In the second round, for each significant MARE practice, the SDO practitioners were provided with their respective individual round 1 rankings and average rankings. The practitioners were requested to reassess their respective individual rankings, for each significant MARE practice, keeping in view the average rankings, if necessary. The average rankings and standard deviations, for each significant MARE, were calculated again after the second round. The average rankings calculated after second round have been shown in Table 1 in descending order**.** This order determines the overall ranks of significant MARE practices for SDO. This provides answer to RQ2 of this study.

Table 1. Average rankings of benefits (descending order) after round 2 of Delphi method.

Sr.	Final Ranks	Signif. Practices	Avg.	Sr.	Final Rank	Signif. Pratices	Avg.	Sr.	Final Rank	Signif. Practices	Avg.
1	1	P_{49}	10.00	30	30	P_{102}	8.75	59	57	P_{32}	8.17
2	2	P_{31}	9.67	31	30	P_{46}	8.75	60	57	P_{5}	8.17
3	2	P_{22}	9.67	32	30	P_{13}	8.75	61	57	P_{89}	8.17
4	4	P_{30}	9.58	33	33	P_{78}	8.67	62	57	P_{81}	8.17
5	5	P_{25}	9.50	34	33	P_{92}	8.67	63	63	P_{59}	8.08
6	6	P_{93}	9.33	35	33	P_{43}	8.67	64	63	P_{51}	8.08
7	7	P_{24}	9.25	36	33	P_{37}	8.67	65	63	P_{61}	8.08
8	7	P_{14}	9.25	37	37	P_{64}	8.58	66	66	P_{73}	8.00
9	7	P_{91}	9.25	38	38	P_{95}	8.50	67	66	P_{72}	8.00
10	10	P_{67}	9.08	39	38	P_{88}	8.50	68	66	P_{101}	8.00
11	11	P_{103}	9.00	40	38	P_{84}	8.50	69	66	P_{86}	8.00
12	11	P_{68}	9.00	41	38	P_{79}	8.50	70	66	P_{74}	8.00
13	11	P_{54}	9.00	42	38	P_{19}	8.50	71	71	P_{62}	7.92
14	11	P_{15}	9.00	43	43	P_{80}	8.42	72	72	P_{66}	7.83
15	11	P_{9}	9.00	44	43	P_{65}	8.42	73	72	P_{18}	7.83
16	11	P_{42}	9.00	45	43	P_{94}	8.42	74	72	P_{71}	7.83
17	11	P_{4}	9.00	46	43	P_{8}	8.42	75	72	P_{60}	7.83
18	18	P_{69}	8.92	47	47	P_{90}	8.33	76	72	P_{7}	7.83
19	18	P_{55}	8.92	48	47	P_{63}	8.33	77	77	P_{76}	7.75
20	18	P_{16}	8.92	49	47	P_{11}	8.33	78	78	P_{23}	7.58
21	18	P_{26}	8.92	50	47	P_{58}	8.33	79	79	P_{29}	7.50
22	22	P_{100}	8.83	51	47	P_{99}	8.33	80	80	P_{97}	7.42
23	22	P_{70}	8.83	52	47	P_{77}	8.33	81	80	P_{44}	7.42
24	22	P_{50}	8.83	53	53	P_{85}	8.25	82	82	P_{52}	7.33
25	22	P_{45}	8.83	54	53	P_{75}	8.25	83	82	P_{82}	7.33
26	22	P_{34}	8.83	55	53	P_{21}	8.25	84	82	P_{53}	7.33
27	22	P_{33}	8.83	56	53	P_{39}	8.25	85	85	P_{96}	7.17
28	22	P_{6}	8.83	57	57	P_{87}	8.17	86	86	P_{98}	7.08
29	22	P_{1}	8.83	58	57	P_{41}	8.17				

5. CONCLUSIONS

This study aims to identify and rank the significant Mobile Application Requirements Engineering (MARE) practices for addressing the issues of MARE process in case of Software Development Outsourcing (SDO). Two rounds of Delphi method have been conducted by involving the SDO practitioners. The respondents have been requested to rank the given practices, extracted from literature, according to the perceived benefits of these practices for SDO. Based on 10-point scale, the different ranks of perceived benefits are: High (10), Medium (7), Low (4) and Zero (1). The study identifies and ranks 86 significant MARE practices for SDO based on the analysis of responses from 36 SDO practitioners. The top 10 practices are about unique identification of requirements, using simple specification language, defining system boundaries, employing standard templates, planning for conflict resolution, informing stakeholders about requirements' changes, facilitating requirements negotiation through software support, using business concerns to drive requirements elicitation, accessibility of all stakeholders to awareness support system and specifying misuse scenarios. Thus, these aspects should be given special attention for a successful MARE process in SDO context, where stakeholders are physically dispersed.

ACKNOWLEDGMENTS

This research was supported financially by the Ministry of Higher Education Malaysia (FRGS-FP068-2015A). We thank the reviewers and editor(s) for their comments to improve this manuscript.

REFERENCES

1. S. Dhar and B. Balakrishnan, "Risks, benefits, and challenges in global IT outsourcing: Perspectives and practices," *Journal of Global Information Management*, Vol. 14, 2006, pp. 59-89.
2. S. U. Khan, M. Niazi, and R. Ahmad, "Barriers in the selection of offshore software development outsourcing vendors: An exploratory study using a systematic literature review," *Information and Software Technology* Vol. 53, 2011, pp. 693-706.
3. D. M. Mahmud and N. A. S. Abdullah, "Mobile application development feasibility studies: a case study in Universiti Teknologi MARA," in *Proceedings of IEEE Conference on Open Systems*, 2014, pp. 30-35.
4. A. Gasimov, C. H. Tan, C. W. Phang, and J. Sutanto, "Visiting mobile application development: What, how and where," in *Proceedings of the 9th IEEE International Conference on Mobile Business and Global Mobility Roundtable*, 2010, pp. 74-81.
5. H. K. Flora, X. Wang, and S. V. Chande, "An investigation into mobile application development processes: challenges and best practices," *International Journal of Modern Education and Computer Science*, Vol. 6, 2014, p. 1.
6. Y. Kouraogo, K. Zkik, and G. Orhanou, "Attacks on Android banking applications," in *Proceedings of IEEE International Conference on Engineering and MIS*, 2016, pp. 1-6.

7. M. Usman, M. Z. Iqbal, and M. U. Khan, "A product-line model-driven engineering approach for generating feature-based mobile applications," *Journal of Systems and Software*, Vol. 12, 2017, pp. 1-32.
8. J. Dehlinger and J. Dixon, "Mobile application software engineering: challenges and research directions," in *Proceedings of Workshop on Mobile Software Engineering*, Vol. 2, 2011, pp. 29-32.
9. T. Vithani and A. Kumar, "Modeling the mobile application development lifecycle," in *Proceedings of International Multi Conference of Engineers and Computer Scientists*, Vol. 1, 2014, pp. 596-600.
10. A. I. Wasserman, "Software engineering issues for mobile application development," in *Proceedings of ACM FSE/SDP Workshop on Future of Software Engineering Research*, 2010, pp. 397-400.
11. D. Gefen, S. Wyss, and Y. Lichtenstein, "Business familiarity as risk mitigation in software development outsourcing contracts," *MIS Quarterly*, Vol. 32, 2008, pp. 531-551.
12. A. Takeoka and P. Wanninayaka, "IT Offshoring risks and governance capabilities," in *Proceedings of the 41st Hawaii International Conference on System Sciences*, 2008, pp. 436-436.
13. M. Niazi, "An instrument for measuring the maturity of requirements engineering process," in *Proceedings of the 6th International Conference on Product Focused Software Process Improvement*, 2005, pp. 574-585.
14. N. V. Oza and T. Hall, "Difficulties in managing offshore outsourcing relationships: an empirical analysis of 18 high maturity Indian software companies," *Journal of Information Technology Cases and Applications Research*, Vol. 7, 2005, pp. 25-41.
15. J. M. Bhat, M. Gupta, and S. N. Murthy, "Overcoming requirements engineering challenges: Lessons from offshore outsourcing," *IEEE Software*, Vol. 23, 2006, pp. 38-44.
16. I. Sommerville and J. Ransom, "An empirical study of industrial requirements engineering process assessment and improvement," *ACM Transactions on Software Engineering and Methodology*, Vol. 14, 2005, pp. 85-117.
17. D. E. Damian and D. Zowghi, "RE challenges in multi-site software development organizations," *Requirements Engineering*, Vol. 8, 2003, pp. 49-160.
18. A. Sharma, A. Serebrenik, and M. Klabbers, "Requirements quality assessment for outsourcing," Master Thesis, Department of Mathematics and Computer Science, Eindhoven University of Technology, 2009.
19. B. Meyer, "The unspoken revolution in software engineering," *Computer*, Vol. 39, 2006, pp. 121-123.
20. M. Niazi, M. El-Attar, M. Usman, and N. Ikram, "GlobReq: a framework for improving requirements engineering in global software development projects: Preliminary results," in *Proceedings of the 16th International Conference on Evaluation and Assessment in Software Engineering*, 2012, pp. 166-170.
21. D. Smite, "Requirements management in distributed projects," *Journal of Universal Knowledge Management*, Vol. 1, 2006, pp. 69-76.
22. M. N. Tahir, S. Khan, and A. Raza, "Challenges in requirements engineering for mobile applications for disabled – Autism," *Journal of Industrial and Intelligent Information*, Vol. 1, 2013, pp. 226-230.

23. H. F. Rasool, N. Saher, Z. Iqbal, M. R. Ajmal, and S. Arshad, "Requirements engineering and its role in mobile telephone industry development," *Journal of Advances in Computer Networks*, Vol. 2, 2014, pp. 218-221.
24. A. Maccari, "The challenges of requirements engineering in mobile telephones industry," in *Proceedings of the 10th International IEEE Workshop on Database and Expert Systems Applications*, 1999, pp. 336-339.
25. S. Hebler, T. Tuunanen, and K. Peffers, "Blind user requirements engineering for mobile services," in *Proceedings of the 15th IEEE International Requirements Engineering Conference*, 2007, pp. 205-208.
26. C. Alves, G. Ramalho, and A. Damasceno, "Challenges in requirements engineering for mobile games development: the meantime case study," in *Proceedings of the 15th IEEE International Requirements Engineering Conference*, 2007, pp. 275-280.
27. G. Büyüközkan, "Determining the mobile commerce user requirements using an analytic approach," *Computer Standards & Interfaces*, Vol. 31, 2009, pp. 144-152.
28. G. Herzwurm, S. Schockert, U. Dowie, and M. Breidung, "Requirements engineering for mobile-commerce applications," in *Proceedings of M-Business*, 2002.
29. M. Nagappan and E. Shihab, "Future trends in software engineering research for mobile apps," in *Proceedings of the 23rd IEEE Software Analysis*, *Evolution*, *and Reengineering*, Vol. 5, 2016, pp. 21-32.
30. N. M. Rizk, A. Ebada, and E. S. Nasr, "Investigating mobile applications' requirements evolution through sentiment analysis of users' reviews," in *Proceedings of the 11th IEEE International Computer Engineering Conference*, 2015 pp. 123-130.
31. N. Seyff and F. Graf, "User-driven requirements engineering for mobile social software," in *Proceedings of Software Engineering Workshops*, 2010, pp. 503-512.
32. M. J. Escalona and N. Koch, "Requirements engineering for web applications-a comparative study," *Journal of Web Engineering*, Vol. 2, 2004, pp. 193-212.
33. K. Rhee, W. Jeon, and D. Won, "Security requirements of a mobile device management system," *International Journal of Security and Its Applications*, Vol. 6, 2012, pp. 353-358.
34. P. Salini and S. Kanmani, "Security requirements engineering process for web applications," *Procedia Engineering*, Vol. 38, 2012, pp. 2799-2807.
35. I. Sommerville and P. Sawyer, "Requirements engineering: a good practice guide," John Wiley & Sons, Inc., 1997, p. 31.
36. N. R. Mead, E. Hough, and T. Stehney, "Security quality requirements engineering (SQUARE) methodology," Report No. CMU/SEI-2005-TR-009, Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, 2005.
37. M. U. A. Khan and M. Zulkernine, "On selecting appropriate development processes and requirements engineering methods for secure software," in *Proceedings of the 33rd Annual Conference on Computer Software and Applications*, Vol. 2, 2009, pp. 353-358.
38. OWASP, https://www.owasp.org/index.php/Category:OWASP_CLASP_Project, 2017.
39. C. B. Haley, J. D. Moffett, R. Laney, and B. Nuseibeh, "A framework for security requirements engineering," in *Proceedings of International Workshop on Software Engineering for Secure Systems*, 2006, pp. 35-42.

40. H. Mouratidis, P. Giorgini, and G. Manson, "When security meets software engineering: a case of modelling secure information systems," *Information Systems*, Vol. 30, 2005, pp. 609-629.
41. J. Viega, "Building security requirements with CLASP," *ACM SIGSOFT Software Engineering Notes*, Vol. 30, 2005, pp. 1-7.
42. D. Mellado, E. Fernández-Medina, and M. Piattini, "A common criteria based security requirements engineering process for the development of secure information sysems," *Computer Standards & Interfaces*, Vol. 29, 2007, pp. 244-253.
43. R. D. Gibbs, "Project management with the IBM rational unified process: Lessons from the trenches," *IBM Press*, 2006, pp. 7-20.
44. D. Damian, "An empirical study of requirements engineering in distributed software projects: is distance negotiation more effective?" in *Proceedings of the 8th Asia-Pacific Software Engineering Conference*, 2001, pp. 149-152.
45. D. Damian, F. Lanubile, and T. Mallardo, "On the need for mixed media in distributed requirements negotiations," *IEEE Transactions on Software Engineering*, Vol. 34, 2008, pp. 116-132.
46. D. Damian, "The study of requirements engineering in global software development: as challenging as important," in *Proceedings of Global Software Development Workshop*, Vol. 9. 2002.
47. F. Lanubile, "A P2P toolset for distributed requirements elicitation," in *Proceedings of the 2nd International Workshop on Global Software Development*, 2003, pp. 12-15.
48. D. Damian, "Stakeholders in global requirements engineering: Lessons learned from practice," *IEEE Software*, Vol. 24, 2007, pp. 21-27.
49. B. Decker, E. Ras, J. Rech, P. Jaubert, and M. Rieth, "Wiki-based stakeholder participation in requirements engineering." *IEEE Software*, Vol. 24, 2007, pp. 28-35.
50. D. E. Damian and D. Zowghi, "An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003, p. 10.
51. M. Heindl, F. Reinisch, and S. Biffl, "Requirements management infrastructures in global software development–towards application lifecycle management with role-based in-time notification," in *Proceedings of Workshop on Tool Support and Requirements Management in Distributed Projects*, 2007, pp. 46-52.
52. D. Damian, J. Chisan, P. Allen, and B. Corrie, "Awareness meets requirements management: awareness needs in global software development," in *Proceedings of International Conference on Software Engineering*, 2003, pp. 7-11.
53. L. D. Catledge and C. Potts, "Collaboration during conceptual design," in *Proceedings the 2nd International Conference on Requirements Engineering*, 1996, pp. 182-189.
54. T. Illes-Seifert, A. Herrmann, M. Geisser, and T. Hildenbrand, "The challenges of distributed software engineering and requirements engineering: Results of an online survey," in *Proceedings of the 1st international Global Requirements Engineering Workshop*, 2007, pp. 55-66.
55. E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," *IEEE Software*, Vol. 18, 2001, pp. 22-29.

56. S. Deshpande, I. Richardson, V. Casey, and S. Beecham, "Culture in global software development-a weakness or strength?" in *Proceedings of the 5th IEEE International Conference on Global Software Engineering*, 2010, pp. 67-76.
57. B. Berenbach, "Impact of organizational structure on distributed requirements engineering processes: lessons learned," in *Proceedings of International Workshop on Global Software Development for the Practitioner*, 2006, pp. 15-19.
58. C. Okoli and S. D. Pawlowski, "The Delphi method as a research tool: an example, design considerations and applications," *Information and Management*, Vol. 42, 2004, pp. 15-29.
59. G. Skulmoski, F. Hartman, and J. Krahn, "The Delphi method for graduate research," *Journal of Information Technology Education: Research*, Vol. 6, 2007, pp. 1-21.
60. R. T. Nakatsu and C. L. Iacovou, "A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: a two-panel Delphi study," *Information & Management*, Vol. 46, 2009, pp. 57-68.
61. K. Cox, M. Niazi, and J. Verner, "Empirical study of Sommerville and Sawyer's requirements engineering practices," *IET Software*, Vol. 3, pp. 339-355.
62. A. Rainer and T. Hall, "Key success factors for implementing software process improvement: a maturity-based analysis," *Journal of Systems and Software*, Vol. 62, 2002, pp. 71-84.
63. M. Niazi, D. Wilson, and D. Zowghi, "A maturity model for the implementation of software process improvement: an empirical study," *Journal of Systems and Software*, Vol. 74, 2005, pp. 155-172.

Javed Iqbal is an Assistant Professor at Department of Computer Science, COMSATS Institute of IT, Islamabad, Pakistan. He received his Ph.D. in Computer Science from University of Malaya, Malaysia in 2016. His areas of interest are software process improvement, requirements engineering, and software development outsourcing.

Rodina Ahmad is an Associate Professor at the Software Engineering Department, Faculty of Computer Science and IT, University of Malaya, KL, Malaysia. Her research interests include requirements engineering and organizational analysis. She received her Ph.D. in Information Systems Management from National University of Malaysia in 2006.

Mohd Hairul Nizam MD Nasir is a Sr. Lecturer at Software Engineering Department, University of Malaya, KL, Malaysia. His research interests include software process models, software process improvement, software quality and project management. He received his Ph.D. in Computer Science from Universiti Teknologi Malaysia.

Muzafar Khan is currently working as an Assistant Professor at Department of Computer Science, COMSATS Institute of IT, Islamabad, Pakistan. He completed his Ph.D. in Human Computer Interaction from Universiti Teknologi PETRONAS, Malaysia in 2012. He has keen interest in interdisciplinary research related to human computer interaction and software engineering.

APPENDIX A: List of RE Practices and Results of Delphi Method Round 1

P_1: Define and follow a standard document structure, P_2: Include a section in introduction part of document to explain how to use it, P_3: Include a summary of the requirements, P_4: Make a business case for the system, P_5: Define specialized terms, P_6: Lay out the document for readability, P_7: Help readers to find desired information, P_8: Make document easy to change, P_9: Assess system feasibility, P_{10}: Be sensitive to organizational and political considerations, P_{11}: Identify and consult system stakeholders, P_{12}: Record requirements originating sources. P_{13}: Define the system's operating environment, P_{14}: Use business concerns to drive requirements elicitation, P_{15}: Look for domain constraints, P_{16}: Record requirements rationale, P_{17}: Collect requirements from multiple viewpoints, P_{18}: Prototype the poorly understood requirements, P_{19}: Use scenarios to elicit requirements, P_{20}: Define operational processes, P_{21}: Reuse requirements from already developed similar systems, P_{22}: Define system boundaries, P_{23}: Use checklists for requirements analysis, P_{24}: Provide software to support negotiations, P_{25}: Plan for conflicts identification and resolution, P_{26}: Prioritize requirements, P_{27}: Classify requirements using a multi-dimensional approach, P_{28}: Use interaction matrices to find requirements conflicts and overlaps, P_{29}: Assess requirements risks, P_{30}: Define and use standard templates for requirements description, P_{31}: Use simple, consistent and concise language to describe requirements, P_{32}: Use diagrams appropriately, P_{33}: Supplement natural language with other descriptions of requirements when appropriate, P_{34}: Specify requirements quantitatively where appropriate, P_{35}: Develop complementary system models, P_{36}: Model the system's environment, P_{37}: Model the system's architecture, P_{38}: Use Structured methods for system modeling, P_{39}: Use a data dictionary, P_{40}: Document the association between stakeholder requirements and system models, P_{41}: Check that the requirements document meets your standards, P_{42}: Organize requirements inspections, P_{43}: Use multi-disciplinary teams to review requirements, P_{44}: Define requirements validation checklists, P_{45}: Use prototyping to animate requirements, P_{46}: Write a user manual draft, P_{47}: Propose requirements test cases, P_{48}: Paraphrase system models into natural language, P_{49}: Uniquely identify each requirement, P_{50}: Define policies for requirements management, P_{51}: Define requirements traceability policies, P_{52}: Maintain traceability manual, P_{53}: Use a database to manage requirements, P_{54}: Define requirements change management policies, P_{55}: Identify global system requirements, P_{56}: Identify volatile requirements, P_{57}: Record rejected requirements, P_{58}: Agreeing on definitions, P_{59}: Selecting techniques for elicitation of security requirements, P_{60}: Specifying high level functional requirements of software, P_{61}: Identifying assets and resources of the software, P_{62}: Evaluating assets and resources of the software, P_{63}: Identifying potential attackers of the software, P_{64}: Identifying possi-

ble interests that attackers may have about the resources or assets, P_{65}**:** Identifying competencies of potential attackers, P_{66}**:** Specifying use cases, P_{67}**:** Specifying misuse scenarios, P_{68}**:** Identifying potential threats, P_{69}**:** Developing security requirements by analyzing potential threats, P_{70}**:** Identifying goals of security by involving all the stakeholders, P_{71}**:** Specifying high level security requirements, P_{72}**:** Specifying security policy and constraints on the software by involving all the stakeholders, P_{73}**:** Specifying low level functional requirements containing requirements of security mechanisms, P_{74}**:** Specifying low level security requirements, P_{75}**:** Performing cost and benefits analysis, P_{76}**:** Classifying and prioritizing low level security requirements, P_{77}**:** Introducing selected low level security requirements in the requirement specifications, P_{78}**:** Managers should arrange the computer-mediated negotiations, involving all the appropriate stakeholders, for resolution of the requirements conflicts, P_{79}**:** Face to face start-off meeting should be scheduled at the start of the project in order to establish personal relationships among key stake holders, P_{80}**:** Synchronous communication should be supported by asynchronous communication (mixed media) in order to provide time for information processing and sifting through the issues which help to resolve outstanding issues and building common grounds, P_{81}**:** Special considerations should be given to persuade the professionals that revealing of the issues will not negatively affect their organizational positions, and instead will assist in overcoming the problems and enhancing the performance, P_{82}**:** A peer-to-peer workshop tool can substitute traditional face to face workshops during which stakeholders work together, P_{83}**:** Open lines should be maintained for communication among all the stakeholders, P_{84}**:** Video conferences or teleconferences should be scheduled daily, weekly, bimonthly, monthly *etc.* so that there are no or minimal inconvenient hours for all the stakeholders, P_{85}**:** Requirements documents should be prepared collaboratively by the remote stakeholders, P_{86}**:** Try to find natural overlapping of working hours, P_{87}**:** Assess 'around-the-clock' capability of working, P_{88}**:** During videoconferencing meetings share the agenda of meeting and list of issues, and designate a facilitator from each stakeholder, P_{89}**:** In order to facilitate the communication with the system users, personnel should be appointed for field support, P_{90}**:** Requirements Management(REQM) tool support providing these features: (i) Continuous access to requirements related information (like history, who is working on requirement, decisions made *etc.*); (ii) On occurrence of certain events (like change in requirements) timely notifying relevant stakeholders. P_{91}**:** By using an awareness support system, all the stakeholders should be able to access following information: (i) Requests' descriptions, rationale and priorities, (ii) Dependencies among the requirements and with design, coding and testing; (iii) Each team member's responsibilities with respect to particular requirement(s) and contact information like email, phone number; (iv) Requirements' initiators; (v) Issues' initiators, status of the resolution of issues and decisions taken due to issues; (vi) Meetings date, time and location, stakeholders that are involved, discussed problems and decisions taken; (vii) Initiators of change request, decisions made about the request of change and people involved in taking decisions, P_{92}**:** After every meeting, a team member or facilitator should summarize that which issues have been raised during the meeting, what has been decided about each issue, which issues are pending, whose responsibility is to find out further information and whose advice should be sought in case of each issue, P_{93}**:** In case of requirement(s) change(s), relevant stakeholder can be informed about change: (i) Through the telephone calls, emails, NetMeeting , messengers, skype; (ii) By generating automatic notifications through the system, P_{94}**:** A standard language should be decided and used for communication, P_{95}**:** A common glossary should be developed and/or used to create consensus on terminologies, P_{96}**:** If all the stakeholders cannot express in standard language then cultural liaisons should be appointed to enable fruitful communication. The individuals, who are familiar with the cultures of key stake holders and keep traveling between their locations, should be given priority for this purpose, P_{97}**:** Through cultural trainings team members are informed about stakeholders' regional and religious values, communication style, English dialect and techniques to deal with the consequences of cultural differences, P_{98}**:** Preparing backup teams at different locations to handle the unforeseen circumstances because of cultural differences. These teams can also be used to provide 24hours-7days support, P_{99}**:** Process standards (like standard terminology, standard language, standard templates *etc.*) should be defined and used, P_{100}**:** Clearly defined responsibilities of each individual and group, P_{101}**:** Authoritative leadership at the level of project managers and team heads, P_{102}**:** Explicit sequence of commands, P_{103}**:** For requirements meetings: (i) Engaging a human facilitator and using a rich communication media that supports integration of data, videos and audios, (ii) Preparation of agenda and then following it, (iii) Relevant participant selection to take part in requirements meetings, (iv) Timely exchange of supporting documents to give participants enough time to read the relevant material, (v) Participants should be able to access the resources (like chatting messages, emails, documents provided by the clients *etc.*) that contain information about the requirements.