AN IMPROVED REQUIREMENT CHANGE MANAGEMENT MODEL FOR AGILE SOFTWARE DEVELOPMENT

NAJIA SAHER

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2021

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Keperluan perniagaan untuk projek pembangunan perisian adalah tidak menentu dan memerlukan penambahbaikan secara berterusan. Oleh yang demikian, kaedah tangkas menjadi semakin popular kerana ia menyokong perubahan keperluan semasa pembangunan perisian tangkas (ASD). Walau bagaimanapun, model sedia ada hanya memberi tumpuan kepada perubahan keperluan fungsian yang tidak mencukupi untuk mencapai kemampanan dan menyokong proses perubahan keperluan. Oleh itu, kajian ini mencadangkan sebuah Model Pengurusan Perubahan Keperluan Tangkas (ARCM) yang dipertingkatkan di mana ia menyediakan sokongan yang lebih baik terhadap perubahan keperluan bukan-fungsian dalam ASD bagi mencapai kemampanan perisian. Kajian ini dilaksanakan dalam empat fasa. Fasa pertama adalah kajian teori yang mengkaji isu dan amalan penting bagi perubahan keperluan dalam ASD. Seterusnya, dalam fasa kedua, kajian penerokaan telah dijalankan untuk menyiasat amalan semasa bagi perubahan keperluan dalam ASD. Kajian ini melibatkan 137 pengamal perisian dari Pakistan. Manakala dalam fasa ketiga, dapatan daripada fasa sebelumnya telah digunakan untuk membina model ARCM. Model ini dibina dengan mengadaptasi kaedah rancang-buat-semak-tindakan (PDCA) yang terdiri daripada 4 fasa. Setiap fasa ini menyediakan matlamat, proses, aktiviti dan amalan yang jelas. Akhirnya, model ini telah dinilai menggunakan pendekatan ulasan pakar dan kajian kes. Seramai enam pakar telah terlibat untuk menentusahkan model ini dan dua kajian kes yang melibatkan dua syarikat perisian di Pakistan telah dijalankan untuk mengesahkan kebolehgunaan model cadangan. Kajian ini mencadangkan model ARCM yang terdiri daripada tiga komponen utama: ciri kemampanan bagi mengendalikan keperluan bukan-fungsian, kaedah analisis kemampanan untuk melaksanakan analisis kesan dan risiko dan, mekanisma penaksiran ARCM menggunakan kaedah metrik matlamat soalan (GQM). Keputusan penilaian menunjukkan bahawa Model ARCM mendapat kepuasan di kalangan pengamal perisian dan mampu dilaksanakan dalam persekitaran sebenar. Dari perspektif teori, kajian ini telah memperkenalkan Model ARCM yang menyumbang kepada bidang Pengurusan Keperluan Tangkas, serta penemuan empirikal yang menjurus kepada isu-isu semasa, cabaran dan amalan RCM. Selain itu, model ARCM menyediakan penyelesaian untuk mengendalikan perubahan keperluan bukan-fungsian dalam ASD. Oleh itu, penemuan ini memberi manfaat kepada pengamal perisian Tangkas dan juga penyelidik bagi memastikan kemampanan perisian dipenuhi yang seterusnya membantu syarikat dalam meningkatkan nilai serahan perisian.

Katakunci: Pembangunan perisian tangkas, Model Pengurusan Perubahan Keperluan Tangkas, Keperluan bukan-fungsian, Analisis kemampanan, Ciri kemampanan.

Abstract

Business requirements for software development projects are volatile and continuously need improvement. Hence, popularity of Agile methodology increases as it welcomes requirement changes during the Agile Software Development (ASD). However, existing models merely focus on change of functional requirements that are not adequate to achieve software sustainability and support change requirement processes. Therefore, this study proposes an improved Agile Requirement Change Management (ARCM) Model which provides a better support of non-functional requirement changes in ASD for achieving software sustainability. This study was carried out in four phases. Phase one is a theoretical study that examined the important issues and practices of requirement change in ASD. Then, in phase two, an exploratory study was conducted to investigate current practices of requirement changes in ASD. The study involved 137 software practitioners from Pakistan. While in phase three, the findings from the previous phases were used to construct the ARCM model. The model was constructed by adapting Plan-Do-Check-Act (PDCA) method which consists of four 4 stages. Every stage provides well-defined aims, processes, activities, and practices. Finally, the model was evaluated by using expert review and case study approaches. There were six experts involved to verify the model and two case studies which involved two software companies from Pakistan were carried out to validate the applicability of the proposed model. The study proposes the ARCM model that consists of three main components: sustainability characteristics for handling non-functional requirements, sustainability analysis method for performing impact and risk analysis and assessment mechanism of ARCM using Goal Question Metrics (GQM) method. The evaluation result shown that the ARCM Model gained software practitioners' satisfaction and able to be executed in a real environment. From the theoretical perspective, this study introduces the ARCM Model that contributed to the field of Agile Requirement Management, as well as the empirical findings that focused on the current issues, challenges and practices of RCM. Moreover, the ARCM model provides a solution for handling the non-functional requirements changes in ASD. Consequently, these findings are beneficial to Agile software practitioners and researchers to ensure the software sustainability are fulfilled hence empowers the companies to improve their value delivery.

Keywords: Agile software development, Agile Requirement Change Management Model, Non-functional requirement, Sustainability analysis, Sustainability characteristics.

Acknowledgement

In the name of Allah, the Most Gracious and the Most Merciful.

I would like to extend my thanks and gratitude to Almighty Allah for His blessings and help to complete this research.

First and foremost, I would like to thank my supervisor Assoc. Prof. Dr. Fauziah Baharom for her relentless guidance, support, motivation, as well as insightful contribution during all the stages of my PhD study. Without her assistance, encouragement, support and valuable comments, the accomplishment of my research would have not been completed. She indeed is an insightful mentor, expertly guiding and supporting me throughout my PhD journey. In addition, my sincere appreciation also goes to my second supervisor Assoc. Prof. Dr. Rohaida Romli, who has been an active, foresighted and also guided me in every stage of the research process, and making the necessary resources available to support this research.

I also thank all Software companies and IT professionals who participated in the study and for the generous giving of their time and the careful completion of the questionnaires and the companies for giving me opportunities to conduct my case studies for this I will always be grateful.

I would like to express my gratitude to my beloved family members for their continuous prayers, patience and supports throughout this journey. I am particularly grateful for all my siblings for their kind love and prayers. I am very thankful to my brothers Dr. Khurram Shahzad, Amir Shahzad and Dr. Faisal Shahzad those are the source of inspiration and encouragement throughout my life. My husband Dr. Waqas Mahmood has an endless source of support and love, and this journey cannot be possible without his assistance. A special thanks to my elder brother, Dr. Khurram Shahzad for his encouragement and sustained support at the challenging part of my PhD endeavour. The last but not the least I would like to dedicate all my work to my late father, S.M. Quddus, and my late mother, Khurshida Quddus, those are the source of every accomplishment in my life.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

AHP	Analytic Hierarchy Process
AM	Agile Modelling
APM	Agile Project Management
ARCM	Agile Requirement Change Management
ARE	Agile Requirement Engineering
ASD	Agile Software Development
BST	Binary Search Tree
CCB	Change Control Board
CHAM	Change Analysis And Management
CMQF	Change Management Quality Factors
SCF	Critical Success Factor
CV	Cumulative Voting
FR	Functional Requirement
GSD	Global Software Development
JAD	Joint Application Design
NFR	Non Functional Requirement
NRM	Need-Based Requirements Management
PEOU	Perceived Ease of Use
PU	Perceived Usefulness
PG	Planning Game
PO	Product Owner
PM	Project Manager
QA	Quality Assurance
QFD	Quality Function Deployment
RC	Requirement Change
RCM	Requirement Change Management
RE	Requirement Engineering
RM	Requirement Management
RO	Research Objective
RP	Requirement Prioritization
RQ	Research Question
SCM	Software Configuration Management
SDLC	Software Development Life Cycle
SME	Small Medium Enterprise
SPSS	Software Package For Social Sciences
SWOT	Strength, Weakness, Opportunity and Threats
TAM	Technology Acceptance Model
V & V	Verification And Validation
VOP	Value-Oriented Prioritization

CHAPTER ONE

INTRODUCTION

1.1 Overview

This research investigates the challenges faced by Agile Software Development (ASD) practitioners in managing continuous change of requirements specifically dealing with sustainability characteristics as Non-Functional Requirement (NFR). This chapter describe an introduction to the study which begins with the background of the study and problem statement. Afterwards, research questions are formulated that is the basis to develop the objectives of the study. Finally, this chapter presents the scope as well as the significance of the research which follows the overview of the remaining chapters of this proposal.

1.2 Background of the Study

Due to changing business landscape and competitive business strategies, organizations are considering the ways to meet their changing business needs through rapid technology implementations. Keeping the business benefits in view, the fast paced and ever changing business requirements has led organizations to consider Agile methodology as alternatives to traditional approaches (Deloitte, 2010; Harb, Noteboom, & Sarnikar, 2015; Hsu & Lin, 2018; Mills, Berthon, & Pitt, 2018). Since the publication of the Agile Manifesto in 2001, ASD has gained significant adoption (Kent Beck et al., 2001). Indeed, ASD has become the default way of building and integrating software solutions. Several studies have been conducted to investigate the current practice of Agile methodology and depicted that Agile is helping organizations around the world in producing high quality software (Shafinah, Fauziah, & Deraman, 2014; Wagner, Fernández, Felderer, & Kalinowski, 2017). Furthermore,

REFERENCES

Abdul Rahman, A., Islam, S., Kalloniatis, C., & Gritzalis, S. (2017). A Risk Management Approach for a Sustainable Cloud Migration. *Journal of Risk and Financial Management*, *10*(4), 20. https://doi.org/10.3390/jrfm10040020

Achimugu, P., Selamat, A., Ibrahim, R., & Mahrin, M. N. (2014). A systematic literature review of software requirements prioritization research. *Information and Software Technology*, *56*(6), 568–585. https://doi.org/10.1016/j.infsof.2014.02.001

Ahl, V. (2005). An experimental comparison of five prioritization methods : Investigating ease of use, accuracy and scalability. Retrieved from http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A833611&dswid=-1024#sthash.LS4e9Nrm.dpbs

Ahmad, R., Baharom, F., & Azham, H. (2015). A Systematic Review on Characteristic and Sub-Characteristic for Software Development towards Software Sustainability. *World Scientific and Engineering Academy and Society (WSEAS) 2015*, 190–199. Retrieved from http://www.wseas.org/cms.action %0A

Ahmad, R., Baharom, F., & Azham, H. (2016). Goal Oriented Measurement for Software Sustainable Evaluation Metric Focused on Environmental Dimension. *Knowledge Management International Conference (KMICe)*, (August), 405–410. Retrieved from http://www.kmice.cms.net.my/

Ahmad, R., Baharom, F., & Azham, H. (2017). Software Sustainability Development : Impactibility Characteristic Focuses on Social Approach. In *Proceedings of the 6th International Conference on Computing and Informatics, ICOCI 2017* (pp. 595–600).

Ahmad, Z., Hussain, M., Rehman, A., Qamar, U., & Afzal, M. (2015). Impact minimization of requirements change in software project through requirements classification. In *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication - IMCOM '15* (pp. 1–5). New York, New York, USA: ACM Press. https://doi.org/10.1145/2701126.2701174

Ahmed, A., Ahmad, S., Ehsan, N., Mirza, E., & Sarwar, S. Z. (2010). Agile software development: Impact on productivity and quality. In *2010 IEEE International Conference on Management of Innovation & Technology* (pp. 287–291). IEEE. https://doi.org/10.1109/ICMIT.2010.5492703

Ahmed, H., Azham, H., & Fauziah, B. (2016). Current Challenges of Requirement Change Management. *Journal of Telecomunication Electronics and Computer Engineering*, *8*(10), 173–176. Retrieved from http://journal.utem.edu.my/index.php/jtec/article/view/1390

Ajila, S. (2002). Change Management: Modelling Software Product Lines Evolution. *Proceeding of the 6 Th World Multi-Conference on Systemic, Cybernetics and Informatics*, *7*, 492–497. Retrieved from http://www.sce.carleton.ca/faculty/ajila/Change Management in Software Product Line.pdf

Akhtar, M. J., Ahsan, A., & Sadiq, W. Z. (2010). Scrum adoption, acceptance and

implementation (a case study of barriers in Pakistan's IT industry and mandatory improvements). In *2010 IEEE 17Th International Conference on Industrial Engineering and Engineering Management* (pp. 458–461). IEEE. https://doi.org/10.1109/ICIEEM.2010.5646573

AL-Ta'ani, R. H., & Razali, R. (2013). Prioritizing Requirements in Agile Development: A Conceptual Framework. *Procedia Technology*, *11*, 733–739. https://doi.org/10.1016/j.protcy.2013.12.252

Al-Tarawneh, F. H. (2014). *A Framework for Cots Software Evaluation and Selection for Cots Mismatches Handling and Non- Functional Requirements*. (Unpublished Doctoral Dissertation). Universiti Utara Malaysia, Kedah, Malaysia.

Al-Zewairi, M., Biltawi, M., Etaiwi, W., & Shaout, A. (2017). Agile Software Development Methodologies : Survey of Surveys. *Journal of Computer and Communications*, *5*, 74–97. https://doi.org/10.4236/jcc.2017.55007

Alan Koch. (2010). *12 Advantages of Agile Software Development*. Retrieved from https://www.globalknowledge.com/ca-en/resources/resource-library/white-paper/12-advantages-of-agile-software-development/

Albertao, F., Xiao, J., Tian, C., Lu, Y., Zhang, K. Q., & Liu, C. (2010). Measuring the Sustainability Performance of Software Projects. In *2010 IEEE 7th International Conference on E-Business Engineering* (pp. 369–373). IEEE. https://doi.org/10.1109/ICEBE.2010.26

Ali, M. A. (2012). Survey on the State of Agile Practices implementation in Pakistan. *International Journal of Information and Communication Technology Research*, *2*(2223–4985), 6. https://doi.org/10.1212/01.wnl.0000187074.27586.d1

Aljallabi, B. M., & Mansour, A. (2015). Enhancement approach for non-functional requirements analysis in Agile environment. In *2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE)* (pp. 428–433). IEEE. https://doi.org/10.1109/ICCNEEE.2015.7381407

Alnoukari, M., Alzoabi, Z., & Hanna, S. (2008). Applying adaptive software development (ASD) agile modeling on predictive data mining applications: ASD-DM methodology. In *2008 International Symposium on Information Technology* (pp. 1–6). IEEE. https://doi.org/10.1109/ITSIM.2008.4631695

Alsalemi, A. M., & Yeoh, E.-T. (2015). A survey on product backlog change management and requirement traceability in agile (Scrum). In *2015 9th Malaysian Software Engineering Conference (MySEC)* (pp. 189–194). IEEE. https://doi.org/10.1109/MySEC.2015.7475219

Alshomrani, S., & Qamar, S. (2012). Hybrid SWOT-AHP Analysis of Saudi Arabia E-Government. *International Journal of Computer Applications*, *48*(2), 1–7. https://doi.org/10.5120/7317-0065

Altarawneh, M. H. A. (2016). *Monitoring Oriented Agile Based Web Applications Development Methodology for Small Software Firms in Jordan*. (Unpublished Doctoral

Dissertation). Universiti Utara Malaysia, Kedah, Malaysia.

Ambler, S. (2014). Agile Requirements Change Management. Retrieved February 20, 2017, from http://agilemodeling.com/essays/changeManagement.htm

Asghar, A. R., Tabassum, A., Bhatti, S. N., & Jadi, A. M. (2017). Impact and challenges of requirements elicitation & prioritization in quality to agile process: Scrum as a case scenario. In *2017 International Conference on Communication Technologies (ComTech)* (pp. 50–55). IEEE. https://doi.org/10.1109/COMTECH.2017.8065749

Asif, M., & Searcy, C. (2014). Towards a standardised management system for corporate sustainable development. *The TQM Journal*, *26*(5), 411–430. https://doi.org/10.1108/TQM-08-2012-0057

Asif, M., Searcy, C., Garvare, R., & Ahmad, N. (2011). Including sustainability in business excellence models. *Total Quality Management & Business Excellence*, *22*(7), 773–786. https://doi.org/10.1080/14783363.2011.585784

Asif, M., Searcy, C., Zutshi, A., & Ahmad, N. (2011). An integrated management systems approach to corporate sustainability. *European Business Review*, *23*(4), 353–367. https://doi.org/10.1108/09555341111145744

Austin, J. E. (2002). *Managing in Developing Countries: Strategic Analysis and Operating Techniques* (5th ed.). Collier Macmilllan Publishers, London.

Awad, M. A. (2005). A Comparison between Agile and Traditional Software Development Methodologies. *The University of Western Australia*, *1*, 1–300. https://doi.org/10.1145/130840.130843

Azar, J., Smith, R., & Cordes, D. (2007). Value-Oriented Requirements Prioritization in a Small Development Organization. *IEEE Software*, *24*(1), 32–37. https://doi.org/10.1109/MS.2007.30

Baharom, F., Deraman, A., Hamdan, A. R., & Shafinah, F. P. (2012). A Case Study of Software Process Assessment and Certification (SPAC) Model Implementation. Retrieved from https://www.semanticscholar.org/paper/A-Case-Study-of-Software-Process-Assessment-and-Baharom-Deraman/728a5a013a3b2c710131b2829088b58ec7dfb1b1

Baharom, F., Yahaya, J., Deraman, A., & Hamdan, A. R. (2011). SPQF: Software Process Quality Factor. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics* (pp. 1–7). IEEE. https://doi.org/10.1109/ICEEI.2011.6021526

Bailey, K. (2014). *Methods of social research, 4th edition*. New York: Free Press.

Bano, M., Imtiaz, S., Ikram, N., Niazi, M., & Usman, M. (2012). Causes of requirement change - a systematic literature review. In *16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012)* (pp. 22–31). IET. https://doi.org/10.1049/ic.2012.0003

Baruah, N. (2015). Requirement Management in Agile Software Environment. *Procedia*

Computer Science, *62*, 81–83. https://doi.org/10.1016/J.PROCS.2015.08.414

Basili, V., Cladiera, G., & Rombach, H. D. (1994). Goal Question Metric Paradigm. *Encyclopedia of Software Engineering, Vol 2, 1*, 529–532.

Basri, S., Kama, N., Haneem, F., & Ismail, S. A. (2016). Predicting effort for requirement changes during software development. In *Proceedings of the Seventh Symposium on Information and Communication Technology - SoICT '16* (pp. 380–387). New York, New York, USA: ACM Press. https://doi.org/10.1145/3011077.3011096

Beck, K., & Andres, C. (2005). *Extreme programming explained : Embrace change*. Boston, USA: Addison-Wesley.

Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., … Thomas, D. (2001). Principles behind the Agile Manifesto. Retrieved February 22, 2017, from http://agilemanifesto.org/principles.html

Becker, C., Betz, S., Chitchyan, R., Duboc, L., Easterbrook, S. M., Penzenstadler, B., … Venters, C. C. (2016). Requirements: The Key to Sustainability. *IEEE Software*, *33*(1), 56–65. https://doi.org/10.1109/MS.2015.158

Becker, C., Chitchyan, R., Duboc, L., Easterbrook, S., Penzenstadler, B., Seyff, N., & Venters, C. C. (2015). Sustainability design and software: the karlskrona manifesto. *Proceedings of the 37th International Conference on Software Engineering - Volume 2*. IEEE Press. Retrieved from http://dl.acm.org/citation.cfm?id=2819082

Behutiye, W., Karhapää, P., Costal, D., Oivo, M., & Franch, X. (2017). Non-functional Requirements Documentation in Agile Software Development: Challenges and Solution Proposal (pp. 515–522). Springer, Cham. https://doi.org/10.1007/978-3-319-69926-4_41

Berander, P., & Andrews, A. (2005). Requirements Prioritization. In A. Aurum and C. Wohlin eds (Ed.), *Engineering and Managing Software Requirements* (pp. 69–94). Berlin/Heidelberg: Springer-Verlag. https://doi.org/10.1007/3-540-28244-0_4

Bergmann, T., & Karwowski, W. (2019). Agile Project Management and Project Success: A Literature Review (pp. 405–414). Springer, Cham. https://doi.org/10.1007/978-3-319-94709-9_39

Berntsson-Svensson, R., & Aurum, A. (2006). Successful software project and products. In *Proceedings of the 2006 ACM/IEEE international symposium on International symposium on empirical software engineering - ISESE '06* (p. 144). New York, New York, USA: ACM Press. https://doi.org/10.1145/1159733.1159757

Bhatti, W., Hayat, F., Ehsan, N., Ishaque, A., Ahmed, S., & Mirza, E. (2010). A methodology to manage the changing requirements of a software project. In *2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)* (pp. 319–322). IEEE. https://doi.org/10.1109/CISIM.2010.5643642

Bhavsar, R., Thakkar, A., Sanghavi, P., & Tanwar, S. (2019). Resolving Conflicts in Requirement Engineering Through Agile Software Development: A Comparative Case Study (pp. 349–357). Springer, Singapore. https://doi.org/10.1007/978-981-13-2324-

9_35

Binti Arbain, A. F., Ghani, I., & Wan Kadir, W. M. N. (2014). Agile non functional requiremnents (NFR) traceability metamodel. In *2014 8th. Malaysian Software Engineering Conference (MySEC)* (pp. 228–233). IEEE. https://doi.org/10.1109/MySec.2014.6986019

Bjarnason, E., Wnuk, K., & Regnell, B. (2011a). A case study on benefits and side-effects of agile practices in large-scale requirements engineering. In *Proceedings of the 1st Workshop on Agile Requirements Engineering - AREW '11* (pp. 1–5). New York, New York, USA: ACM Press. https://doi.org/10.1145/2068783.2068786

Bjarnason, E., Wnuk, K., & Regnell, B. (2011b). Requirements are slipping through the gaps — A case study on causes & effects of communication gaps in large-scale software development. In *2011 IEEE 19th International Requirements Engineering Conference* (pp. 37–46). IEEE. https://doi.org/10.1109/RE.2011.6051639

Bohner. (1996). Impact analysis in the software change process: a year 2000 perspective. In *Proceedings of International Conference on Software Maintenance ICSM-96* (pp. 42–51). IEEE. https://doi.org/10.1109/ICSM.1996.564987

Bohner, S. A., & Arnold, R. S. (1996). *Software change impact analysis*. IEEE Computer Society Press.

Bolarinwa, O. A. (2015). Principles and methods of validity and reliability testing of questionnaires used in social and health science researches. *The Nigerian Postgraduate Medical Journal*, *22*(4), 195–201. https://doi.org/10.4103/1117-1936.173959

Brundtland, G. H. (1987). *Our Common Future: Report of the World Commission on Environment and Development*. *United Nations CommissionUnited Nations World Commission on Environment and Development* (Vol. 4). https://doi.org/10.1080/07488008808408783

Bukhsh, F. A., Bukhsh, Z. A., & Daneva, M. (2020). A systematic literature review on requirement prioritization techniques and their empirical evaluation. *Computer Standards & Interfaces*, *69*, 103389. https://doi.org/10.1016/J.CSI.2019.103389

Butt, S. aziz, & Jamal, T. (2017). Frequent Change Request From User to Handle Cost on Project in Agile Model. *Asia Pacific Journal of Multidisciplinary Research*, *5*(2), 26–42. Retrieved from https://www.academia.edu/32873984/Frequent_Change_Request_From_User_to_Handle_Cost_on_Project_in_Agile_Model

Calero, C., Bertoa, M. F., & Moraga, M. A. (2013). A systematic literature review for software sustainability measures. In *2013 2nd International Workshop on Green and Sustainable Software (GREENS)* (pp. 46–53). IEEE. https://doi.org/10.1109/GREENS.2013.6606421

Calero, C., Moraga, M. angeles, Bertoa, M. F., & Duboc, L. (2014). Quality in use and software greenability. *CEUR Workshop Proceedings*, *1216*, 28–36. Retrieved from http://ceur-ws.org/Vol-1216/paper6.pdf

Cao, L., & Ramesh, B. (2008). Agile Requirements Engineering Practices: An Empirical Study. *IEEE Software*, *25*(1), 60–67. https://doi.org/10.1109/MS.2008.1

Ceschi, M., Sillitti, A., Succi, G., & De Panfilis, S. (2005). Project Management in Plan-Based and Agile Companies. *IEEE Software*, *22*(3), 21–27. https://doi.org/10.1109/MS.2005.75

CHAOS Report. (2014). The Standish Group: the CHAOS Report. *Project Smart*, 16. https://doi.org/10.1016/S0895-7061(01)01532-1

Chatzipetrou, P., Angelis, L., Rovegard, P., & Wohlin, C. (2010). Prioritization of Issues and Requirements by Cumulative Voting: A Compositional Data Analysis Framework. In *2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications* (pp. 361–370). IEEE. https://doi.org/10.1109/SEAA.2010.35

Chen, Y.-L. (2010). *Analysis of the Agile Deployment*. Retrieved from https://gupea.ub.gu.se/handle/2077/21954

Chow, T., & Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, *81*(6), 961–971. https://doi.org/10.1016/j.jss.2007.08.020

Chung, L., & do Prado Leite, J. C. S. (2009). On Non-Functional Requirements in Software Engineering (pp. 363–379). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-02463-4_19

Cleland-Huang, J. (2012). Traceability in Agile Projects. In *Software and Systems Traceability* (pp. 265–275). London: Springer London. https://doi.org/10.1007/978-1-4471-2239-5_12

Cleland-Huang, J., Gotel, O. C. Z., Huffman Hayes, J., Mäder, P., & Zisman, A. (2014). Software traceability: trends and future directions. In *Proceedings of the on Future of Software Engineering - FOSE 2014* (pp. 55–69). New York, New York, USA: ACM Press. https://doi.org/10.1145/2593882.2593891

CMMI Version 1.3. (2006). *CMMI For Development, Version 1.3, CMMI® Institute*. Retrieved from https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9661

Cockburn, A. (2005). *Crystal clear : a human-powered methodology for small teams*. Addison-Wesley.

Cohn, M. (2006). *Agile estimating and planning*. Prentice Hall Professional Technical Reference.

Cohn, M., & Ford, D. (2003). Introducing an Agile Process to an Organization. *Computer*, *36*(6), 74–78. https://doi.org/10.1109/MC.2003.1204378

Cooper, D. R., & Schindler, P. S. (2003). *Business research methods*. McGraw-Hill/Irwin.

COSO. (2004). *Enterprise Risk Management — Integrated Framework. New York* (Vol. 3). https://doi.org/10.1504/IJISM.2007.013372

Creswell, J. W. (2014). *Research design : qualitative, quantitative, and mixed methods approaches*. SAGE Publications.

Croitoru, H. (2018). Agile Project Management. In *Agile Office 365* (pp. 31–46). Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-4081-6_2

Crow, K. (2009). Customer-focused development with QFD. *Annual Quality Congress Proceedings-American Society for Quality Control*.

DAMA UK Working Group. (2013). *The six primary dimensions for data quality assessment: defining data quality dimensions*.

Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, *13*(3), 319. https://doi.org/10.2307/249008

De Lucia, A., & Qusef, A. (2010). Requirements engineering in agile software development. *Journal of Emerging Technologies in Web Intelligence*, *2*(3), 212–220. https://doi.org/10.4304/jetwi.2.3.212-220

Deloitte. (2010). *Don't fear change , embrace it: Advancing the case for agile methods in systems integration*. Retrieved from https://www2.deloitte.com/ie/en/pages/technology/articles/methods-in-systems-integration.html

Deming, W. E. (William E. (2000). *The new economics : for industry, government, education*. MIT Press. Retrieved from https://mitpress.mit.edu/books/new-economics-industry-government-education

Demuth, A., Riedl-Ehrenleitner, M., Kretschmer, R., & Egyed, A. (2016). Towards efficient risk-identification in risk-driven development processes. In *Proceedings of the International Workshop on Software and Systems Process - ICSSP '16* (pp. 36–40). New York, New York, USA: ACM Press. https://doi.org/10.1145/2904354.2904364

Differding, C., Hoisl, B., & Lott, C. M. (1996). Technology Package for the Goal Question Metric Paradigm. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.5631

Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*. https://doi.org/10.1016/j.jss.2012.02.033

Domah, D., & Mitropoulos, F. J. (2015). The NERV methodology: A lightweight process for addressing non-functional requirements in agile software development. In *SoutheastCon 2015* (pp. 1–7). IEEE. https://doi.org/10.1109/SECON.2015.7133028

Dooley, J. F. (2017). Software Process Models. In *Software Development, Design and Coding* (pp. 7–27). Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-3153-1_2

Downey, J. (2007). *Strategic Analysis Tools Topic Gateway Series Strategic Analysis Tools Topic Gateway Series No. 34*. Retrieved from www.cimaglobal.com

Duan, C., Laurent, P., Cleland-Huang, J., & Kwiatkowski, C. (2009). Towards automated

requirements prioritization and triage. *Requirements Engineering*, *14*(2), 73–89. https://doi.org/10.1007/s00766-009-0079-7

Eberlein, & Leite, J. (2002). Agile Requirements Definition: A View from Requirements Engineering.

Ernst, N. A., Borgida, A., Mylopoulos, J., & Jureta, I. J. (2012). Agile Requirements Evolution via Paraconsistent Reasoning (pp. 382–397). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-31095-9_25

Etikan, I., Musa, S. A., & Alkassim, R. S. (2016). Comparison of Convenience Sampling and Purposive Sampling. *American Journal of Theoretical and Applied Statistics*, *5*(1), 1. https://doi.org/10.11648/j.ajtas.20160501.11

Farid, W. M. (2012). The NORMAP Methodology: Lightweight Engineering of Non-functional Requirements for Agile Processes. In *2012 19th Asia-Pacific Software Engineering Conference* (pp. 322–325). IEEE. https://doi.org/10.1109/APSEC.2012.23

Farid, W. M., & Mitropoulos, F. J. (2012). NORMATIC: A visual tool for modeling Non-Functional Requirements in agile processes. In *2012 Proceedings of IEEE Southeastcon* (pp. 1–8). IEEE. https://doi.org/10.1109/SECon.2012.6196989

Farid, W. M., & Mitropoulos, F. J. (2013). NORPLAN: Non-functional Requirements Planning for agile processes. In *2013 Proceedings of IEEE Southeastcon* (pp. 1–8). IEEE. https://doi.org/10.1109/SECON.2013.6567463

Fauziah, B., Jamaiah H., Y., Aziz, D., & Abdul Razak, H. (2013). Software Process Certification: A Practical Model for Maintaining Software Quality. *International Journal of Information Processing and Management*, *4*(3), 51–61. https://doi.org/10.4156/ijipm.vol4.issue3.5

Fehlmann, T. M. (2008). New Lanchester Theory for Requirements Prioritization. In *2008 Second International Workshop on Software Product Management* (pp. 35–40). IEEE. https://doi.org/10.1109/IWSPM.2008.6

Fernández, D. M., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetrò, A., … Wieringa, R. (2017). Naming the pain in requirements engineering. *Empirical Software Engineering*, *22*(5), 2298–2338. https://doi.org/10.1007/s10664-016-9451-7

Forte, F., & Kloppenborg, T. (2018). The Agile Mindset for Project Management. In *The Modern Project: Mindsets, Toolsets, and Theoretical Frameworks*. UTS ePRESS. https://doi.org/10.5130/pmrp.irnop2017.5740

Fowler, M., & Beck, K. (1999). *Refactoring : improving the design of existing code*. Addison-Wesley.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*, *9*(August), 28–35. https://doi.org/10.1177/004057368303900411

Franch, X., Gómez, C., Jedlitschka, A., López, L., Martínez-Fernández, S., Oriol, M., & Partanen, J. (2018). Data-Driven Elicitation, Assessment and Documentation of Quality Requirements in Agile Software Development (pp. 587–602). Springer, Cham.

https://doi.org/10.1007/978-3-319-91563-0_36

Fu, Y., Li, M., & Chen, F. (2012). Impact propagation and risk assessment of requirement changes for software development projects based on design structure matrix. *International Journal of Project Management*, *30*(3), 363–373. https://doi.org/10.1016/J.IJPROMAN.2011.08.004

Ghayyur, S. A. K., Ahmed, S., Ullah, S., & Ahmed, W. (2018). The Impact of Motivator and Demotivator Factors on Agile Software Development. *International Journal of Advanced Computer Science and Applications*, *9*(7). https://doi.org/10.14569/IJACSA.2018.090712

Ghosh, S., Ramaswamy, S., & Jetley, R. P. (2013). Towards Requirements Change Decision Support. In *2013 20th Asia-Pacific Software Engineering Conference (APSEC)* (pp. 148–155). IEEE. https://doi.org/10.1109/APSEC.2013.30

Gidey, E., Jilcha, K., Beshah, B., & Kitaw, D. (2014). The Plan-Do-Check-Act Cycle of Value Addition. *Industrial Engineering & Management*, *3*(1). https://doi.org/10.4172/2169-0316.1000124

Gopichand, M., & Rao, A. A. (2012). Four Layered Approach to Non-Functional Requirements Analysis. Retrieved from http://arxiv.org/abs/1201.6141

Gorenflo, G., & Moran, J. W. (2010). The ABCs of PDCA. Retrieved from http://www.phf.org/resourcestools/Pages/The_ABCs_of_PDCA.aspx

Grady, R. B., & B., R. (1992). *Practical software metrics for project management and process improvement*. Prentice Hall. Retrieved from https://dl.acm.org/citation.cfm?id=140207

Grobbelaar, S. S. (2012). *R & D in the National system of innovation : A system dynamics model By*. *Research Policy*. Doctoral dissertation, University of Pretoria. https://doi.org/10.1016/S0166-4972(01)00035-9

Gupta, P. (2006). Beyond PDCA: A new process management model. *Quality Progress*, *39*(7), 45–52. Retrieved from https://dbproxy.udallas.edu/login?url=http://search.proquest.com/docview/214765105?accountid=7106

Gurel, E. (2017). SWOT ANALYSIS: A THEORETICAL REVIEW. *Journal of International Social Research*, *10*(51), 994–1006. https://doi.org/10.17719/jisr.2017.1832

Hallowell, M. R., & Gambatese, J. A. (2010). Qualitative Research: Application of the Delphi Method to CEM Research. *Journal of Construction Engineering and Management*, *136*(1), 99–107. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000137

Hameed, T., Latif, M., & Kholief, S. (2016). Identify and Classify Critical Success Factor of Agile Software Development Methodology Using Mind Map. *International Journal of Advanced Computer Science and Applications*, *7*(5). https://doi.org/10.14569/IJACSA.2016.070513

Hamzah, M. H. (2018). *A Service-Oriented Architecture Adoption Maturity Model Focuses on Cross Evaluation Between IT and Business Benefits*. (Unpublished Doctoral Dissertation). Universiti Utara Malaysia, Kedah, Malaysia.

Hannay, J. E., Benestad, H. C., & Strand, K. (2017). Benefit Points: The Best Part of the Story. *IEEE Software*, *34*(3), 73–85. https://doi.org/10.1109/MS.2017.66

Harb, Y., Noteboom, C., & Sarnikar, S. (2015). Evaluating Project Characteristics for Selecting the Best-fit Agile Software Development Methodology. *Journal of the Midwest Association for Information Systems*, *1*(1), 33–52. https://doi.org/10.17705/3jmwa.00004

Harker, S. D. P., Eason, K. D., & Dobson, J. E. (1993). The change and evolution of requirements as a challenge to the practice of software engineering. In *[1993] Proceedings of the IEEE International Symposium on Requirements Engineering* (pp. 266–272). IEEE Comput. Soc. Press. https://doi.org/10.1109/ISRE.1993.324847

Hatton, S. (2008). Choosing the Right Prioritisation Method. In *19th Australian Conference on Software Engineering (aswec 2008)* (pp. 517–526). IEEE. https://doi.org/10.1109/ASWEC.2008.4483241

Heales, J. (2000). Factors affecting information system volatility. *Proceedings of the Twenty First International Conference on Information Systems*. Association for Information Systems.

Heck, P., & Zaidman, A. (2017). A framework for quality assessment of just-in-time requirements: the case of open source feature requests. *Requirements Engineering*, *22*(4), 453–473. https://doi.org/10.1007/s00766-016-0247-5

Heikkila, V. T., Damian, D., Lassenius, C., & Paasivaara, M. (2015). A Mapping Study on Requirements Engineering in Agile Software Development. In *2015 41st Euromicro Conference on Software Engineering and Advanced Applications* (pp. 199–207). IEEE. https://doi.org/10.1109/SEAA.2015.70

Hoff, G., Fruhling, A., & Ward, K. (2008). Requirement prioritization decision factors for agile development environments. Retrieved from https://nebraska.pure.elsevier.com/en/publications/requirement-prioritization-decision-factors-for-agile-development

Hoorn, J., Konijn, E., Vliet, H., & Veer, G. (2007). Requirements change: Fears dictate the must haves; desires the won't haves. *Journal of Systems and Software*, *80*(3), 328–355. https://doi.org/10.1016/j.jss.2006.02.064

Hsu, H., & Lin, Y. (2018). How Agile Impacts a Software Corporation: An Empirical Study. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (pp. 20–25). IEEE. https://doi.org/10.1109/COMPSAC.2018.10197

Hussain, W. (2016). *Requirements Change Management in Global Software Development : A Multiple Case Study*. (Unpublished Doctoral Dissertation). School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology.

Hussain, W., & Clear, T. (2012). GRCM: a model for global requirements change management. Retrieved from http://aut.researchgateway.ac.nz/handle/10292/4565

Hussain, W., Zowghi, D., Clear, T., MacDonell, S., & Blincoe, K. (2016). Managing Requirements Change the Informal Way: When Saying "No" is Not an Option. In *2016 IEEE 24th International Requirements Engineering Conference (RE)* (pp. 126–135). IEEE. https://doi.org/10.1109/RE.2016.64

Iberle, K. (2018). Talking About Quality: How to Define Quality Targets in an Agile World - PNSQC. In *36th Software Quality Conference*. Retrieved from https://www.pnsqc.org/talking-about-quality-how-to-define-quality-targets-in-an-agile-world/

Ibrahim, S., Munro, M., & Deraman, A. (2005). A Requirements Traceability to Support Change Impact Analysis. *International Conference on Information Technology in Asia*, *4*(4), 345–355. Retrieved from https://www.academia.edu/19303509/A_REQUIREMENTS_TRACEABILITY_TO_SUPPORT_CHANGE_IMPACT_ANALYSIS

Imtiaz, S., Ikram, N., & Imtiaz, S. (2008). A process model for managing requirement change. *Proceedings of the Fourth IASTED International Conference on Advances in Computer Science and Technology*, 121–128.

Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, *51*, 915–929. https://doi.org/10.1016/j.chb.2014.10.046

Ince, D. (Darrel). (1994). *An introduction to software quality assurance and its implementation*. McGraw-Hill.

Ismail, W., Abedlazeez, N., & Hussin, Z. (2011, December 18). Epistemological Beliefs of Students at High Schools: A Survey Study in Malaysia. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1974094

ISO/IEC-9126-1. (2001). Information technology - Software quality characteristics and metrics - Part 1: Quality characteristics and sub-characteristics, ISO, Int. Electrotechnical Commission., *2001*, 1–8. https://doi.org/10.1109/IEEESTD.2015.7106438

ISO/IEC 25012. (2008). *ISO/IEC 25012:2008 - Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) - Data quality model*. Retrieved from https://www.iso.org/standard/35736.html

Javed, T., Maqsood, M. e, & Durrani, Q. S. (2004). A study to investigate the impact of requirements instability on software defects. *ACM SIGSOFT Software Engineering Notes*, *29*(3), 1. https://doi.org/10.1145/986710.986727

Jayatilleke, S., & Lai, R. (2018). A systematic review of requirements change management. *Information and Software Technology*, *93*, 163–185. https://doi.org/10.1016/J.INFSOF.2017.09.004

JingFeng, N., Zhiyu, C., & Gang, L. (2010). PDCA process application in the continuous improvement of software quality. In *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*. IEEE. https://doi.org/10.1109/CMCE.2010.5609635

Jones, C. (2016). COMPARING SIX SOFTWARE DEVELOPMENT METHODOLOGIES: SCHEDULES, PRODUCTIVITY, AND QUALITY. Retrieved from www.Namcook.com

Jun, L., Qiuzhen, W., & Lin, G. (2010). Application of Agile Requirement Engineering in Modest-Sized Information Systems Development. In *2010 Second World Congress on Software Engineering* (pp. 207–210). IEEE. https://doi.org/10.1109/WCSE.2010.105

Kapyaho, M., & Kauppinen, M. (2015). Agile requirements engineering with prototyping: A case study. In *2015 IEEE 23rd International Requirements Engineering Conference (RE)* (pp. 334–343). IEEE. https://doi.org/10.1109/RE.2015.7320450

Karlsson, J., & Ryan, K. (1997). A cost-value approach for prioritizing requirements. *IEEE Software*, *14*(5), 67–74. https://doi.org/10.1109/52.605933

Karlsson, J., Wohlin, C., & Regnell, B. (1998). An evaluation of methods for prioritizing software requirements. *Information and Software Technology*, *39*(14–15), 939–947. https://doi.org/10.1016/S0950-5849(97)00053-0

Kassab, M. (2014). An Empirical Study on the Requirements Engineering Practices for Agile Software Development. In *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications* (pp. 254–261). IEEE. https://doi.org/10.1109/SEAA.2014.77

Kassab, M., Daneva, M., & Ormandjieva, O. (2007, June 18). Early Quantitative Assessment of Non-Functional Requirements. Centre for Telematics and Information Technology (CTIT). Retrieved from https://research.utwente.nl/en/publications/early-quantitative-assessment-of-non-functional-requirements

Kaur, G., & Bawa, S. (2013). A Survey of Requirement Prioritization Methods. *International Journal of Engineering Research & Technology (IJERT)*, *2*(5), 958–962.

Kelle, E. van, Visser, J., Plaat, A., & Wijst, P. van der. (2015). An Empirical Study into Social Success Factors for Agile Software Development. In *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering* (pp. 77–80). IEEE. https://doi.org/10.1109/CHASE.2015.24

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, … Dave Thomas. (2001). Manifesto for Agile Software Development. Retrieved February 2, 2017, from http://agilemanifesto.org/

Khan, A. A. (2017). *Software Process Improvement Implementation and Management Model (SPIIMM) for Global Software Development*. Ph.D Dissertation, City University of Hong Kong. Retrieved from https://pdfs.semanticscholar.org/8ebb/cf1862a7f96a0516b47d96a3e3b3bb8c43cf.pdf

Khan, A. A., Basri, S., & Dominic, P. D. D. (2012). A propose framework for requirement Change Management in Global Software Development. In *2012 International Conference on Computer & Information Science (ICCIS)* (pp. 944–947). IEEE. https://doi.org/10.1109/ICCISci.2012.6297161

Khan, A. A., Basri, S., Dominic, P. D. D., & Fazal-e-Amin. (2012). A process model for Requirements Change Management in collocated software development. In *2012 IEEE Symposium on E-Learning, E-Management and E-Services* (pp. 1–6). IEEE. https://doi.org/10.1109/IS3e.2012.6414949

Khan, J. A., Rehman, I., Hayat Khan, Y., Javed Khan, I., & Rashid, S. (2015). Comparison of Requirement Prioritization Techniques to Find Best Prioritization Technique. *International Journal of Modern Education and Computer Science*, *7*(11), 53–59. https://doi.org/10.5815/ijmecs.2015.11.06

Khari, M., & Kumar, N. (2013). Comparison of six prioritization techniques for software requirements. *Journal of Global Research in Computer Science*, *4*(1), 38–43. Retrieved from http://www.jgrcs.info/index.php/jgrcs/article/view/568

Kitchenham, B. A., & Pfleeger, S. L. (2002). Principles of survey research part 2. *ACM SIGSOFT Software Engineering Notes*, *27*(1), 18–20. https://doi.org/10.1145/566493.566495

Klassen, M. (2018). *What Is Change Management Process?* Retrieved from https://www.cherwell.com/library/blog/what-is-change-management-process/

Kobayashi, A., & Maekawa, M. (2001). Need-based requirements change management. In *Proceedings. Eighth Annual IEEE International Conference and Workshop On the Engineering of Computer Based Systems-ECBS 2001* (pp. 171–178). IEEE Comput. Soc. https://doi.org/10.1109/ECBS.2001.922419

Koçak, S., Alptekin, G., & Bener, A. (2014). Evaluation of Software Product Quality Attributes and Environmental Attributes using ANP Decision Framework. *RE4SuSy@ RE*. Retrieved from http://ceur-ws.org/Vol-1216/paper7.pdf

Kotonya, G., & Sommerville, I. (1998). *Requirements engineering : processes and techniques*. J. Wiley.

Koziolek, H., & Heiko. (2011). Sustainability evaluation of software architectures: A Systematic Review. In *Proceedings of the joint ACM SIGSOFT conference -- QoSA and ACM SIGSOFT symposium -- ISARCS on Quality of software architectures -- QoSA and architecting critical systems -- ISARCS - QoSA-ISARCS '11* (p. 3). New York, New York, USA: ACM Press. https://doi.org/10.1145/2000259.2000263

Krejcie, R. V., & Morgan, D. W. (1970). Determining Sample Size for Research Activities. *Educational and Psychological Measurement*, *30*(3), 607–610. https://doi.org/10.1177/001316447003000308

Krishna Kaiser, A. (2018). Managing Changes in a DevOps Project. In *Reinventing ITIL® in the Age of DevOps* (pp. 231–270). Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-3976-6_9

Kumar, A., & Sharma, A. (2015). SWOT ANALYSIS OF REQUIREMENTS ENGINEERING FOR WEB APPLICATIONS. Retrieved from https://www.semanticscholar.org/paper/SWOT-ANALYSIS-OF-REQUIREMENTS-ENGINEERING-FOR-WEB-Kumar-Sharma/7cf1fa172b8a7db363aba1cd68ac161994cd42d1

Kunda, D. (2003). STACE: Social Technical Approach to COTS Software Evaluation (pp. 64–84). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-45064-1_4

Lam, W., Shankararaman, V., Jones, S., Hewitt, J., & Britton, C. (1998). Change analysis and management: a process model and its application within a commercial setting. In *Proceedings. 1998 IEEE Workshop on Application-Specific Software Engineering and Technology. ASSET-98 (Cat. No.98EX183)* (pp. 34–39). IEEE Comput. Soc. https://doi.org/10.1109/ASSET.1998.688230

Leffingwell, D. (2011). *Agile software requirements : lean requirements practices for teams, programs, and the enterprise*. Addison-Wesley.

Leffingwell, D., & Widrig, D. (2000). *Managing software requirements : a unified approach*. Addison-Wesley.

Lehnert, S., & Steffen. (2011). A taxonomy for software change impact analysis. In *Proceedings of the 12th international workshop and the 7th annual ERCIM workshop on Principles on software evolution and software evolution - IWPSE-EVOL '11* (p. 41). New York, New York, USA: ACM Press. https://doi.org/10.1145/2024445.2024454

Lehtola, L., Kauppinen, M., & Kujala, S. (2004). Requirements Prioritization Challenges in Practice (pp. 497–508). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-24659-6_36

Li Xu-Xiang, & Zhang, W.-N. (2010). The PDCA-based software testing improvement framework. In *The 2010 International Conference on Apperceiving Computing and Intelligence Analysis Proceeding* (pp. 490–494). IEEE. https://doi.org/10.1109/ICACIA.2010.5709948

Lim, S. L., & Finkelstein, A. (2011). Anticipating Change in Requirements Engineering. In *Relating Software Requirements and Architectures* (pp. 17–34). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-21001-3_3

Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., … Zelkowitz, M. (2002). Empirical Findings in Agile Methods (pp. 197–207). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45672-4_19

Liu, J., & Cai, J. (2016). Construction Cost Control Based on PDCA Cycle. *International Journal of Simulation - Systems, Science & Technology*, *17*(37), 1–9. https://doi.org/10.5013/IJSSST.a.17.37.19

Lloyd, D., Moawad, R., & Kadry, M. (2017). A supporting tool for requirements change management in distributed agile development. *Future Computing and Informatics Journal*, *2*(1), 1–9. https://doi.org/10.1016/J.FCIJ.2017.04.001

Lock, S., & Kotonya, G. (1999). An Integrated, Probabilistic Framework for Requirement Change Impact Analysis. *Australasian Journal of Information Systems*, *6*(2). https://doi.org/10.3127/ajis.v6i2.292

Lopez, L., Partanen, J., Rodriguez, P., & Martinez-Fernandez, S. (2018). How Practitioners Manage Quality Requirements in Rapid Software Development: A Survey. In *2018 IEEE 1st International Workshop on Quality Requirements in Agile Projects (QuaRAP)* (pp. 14–17). IEEE. https://doi.org/10.1109/QuaRAP.2018.00009

Maiti, R. R., & Mitropoulos, F. J. (2015). Capturing, eliciting, predicting and prioritizing (CEPP) non-functional requirements metadata during the early stages of agile software development. In *SoutheastCon 2015* (pp. 1–8). IEEE. https://doi.org/10.1109/SECON.2015.7133007

Maiti, R. R., & Mitropoulos, F. J. (2017). Prioritizing Non-Functional Requirements in Agile Software Engineering. In *Proceedings of the SouthEast Conference on - ACM SE '17* (pp. 212–214). New York, New York, USA: ACM Press. https://doi.org/10.1145/3077286.3077565

Mäkäräinen, M. (2000). *Software change management process in the development of embedded software*. *Expoo 2000, Technical Research Center of Finland, VTT Publications*. (Unpublished Doctoral Dissertation), University of Oulu, Finland.

Mathisen, E., Ellingsen, K., & Fallmyr, T. (2009). Using business process modelling to reduce the effects of requirements changes in software projects. In *2009 2nd International Conference on Adaptive Science & Technology (ICAST)* (pp. 14–19). IEEE. https://doi.org/10.1109/ICASTECH.2009.5409754

McCall, J. A., Richards, P. K., & Walters, G. F. (1977). Factors in Software Quality. Volume I. Concepts and Definitions of Software Quality. Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA049014

McGee, S., & Greer, D. (2009). A Software Requirements Change Source Taxonomy. In *2009 Fourth International Conference on Software Engineering Advances* (pp. 51–58). IEEE. https://doi.org/10.1109/ICSEA.2009.17

McGee, S., & Greer, D. (2011). Software requirements change taxonomy: Evaluation by case study. In *2011 IEEE 19th International Requirements Engineering Conference* (pp. 25–34). IEEE. https://doi.org/10.1109/RE.2011.6051641

McGee, S., & Greer, D. (2012). Towards an understanding of the causes and effects of software requirements change: two case studies. *Requirements Engineering*, *17*(2), 133–155. https://doi.org/10.1007/s00766-012-0149-0

Medeiros, J. D. R. V, Alves, D. C. P., Vasconcelos, A., Silva, C., & Wanderley, E. (2015). Requirements engineering in agile projects: A systematic mapping based in evidences of industry. *CibSE*, 460–473. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84936074918&partnerID=tZOtx3y1

Méndez Fernández, D., & Wagner, S. (2015). Naming the pain in requirements engineering: A design for a global family of surveys and first results from Germany. *Information and Software Technology*, *57*, 616–643. https://doi.org/10.1016/J.INFSOF.2014.05.008

Méndez Fernández, D., Wagner, S., Lochmann, K., Baumann, A., & de Carne, H. (2012). Field study on requirements engineering: Investigation of artefacts, project parameters, and execution strategies. *Information and Software Technology*, *54*(2), 162–178. https://doi.org/10.1016/J.INFSOF.2011.09.001

Miller, G. (2001). The Characteristics of Agile Software Processes. *International Conferences and Exhibition on Technology of Object-Oriented Languages and Systems (TOOLS)*, 10–12. https://doi.org/10.1016/j.catena.2014.05.025

Mills, A. J., Berthon, P. R., & Pitt, C. (2018). Agile authorship: Evolving models of innovation for information-intensive offerings. *Journal of Business Research*. https://doi.org/10.1016/J.JBUSRES.2018.05.010

Minhas, N. M., Qurat-ul-Ain, Zafar-ul-Islam, & Zulfiqar, A. (2014). An Improved Framework for Requirement Change Management in Global Software Development - ProQuest. *Journal of Software Engineering and Applications*, *7*(9), 779–790. Retrieved from http://search.proquest.com/openview/0923b999cb49cf710a9cd351b7f62a37/1?pq-origsite=gscholar&cbl=51909

Mitchell, C. (2016). Model Validation: Four Elements of Determining the Accuracy of Your Models | BBA. Retrieved August 19, 2017, from https://www.bba.org.uk/news/insight/model-validation-four-elements-of-determining-the-accuracy-of-your-models/#.WZgKRygjFPY

Mnkandla, E., & Dwolatzky, B. (2004). A survey of agile methodologies. *SAIEE Africa Research Journal*, *95*(4), 236–247.

Morkos, B., Shankar, P., & Summers, J. (2012). Predicting Requirement Change Propagation Using Higher Order Design Structure Matrices: An Industry Case Study. *Journal of Engineering Design*, *23*(12). Retrieved from http://tigerprints.clemson.edu/cedar_pubs/4

Nagabhushan, S. ., & Swarnalatha, K. . (2010). Requirement Change Management. *Global Journal of Management And Business Research*, *10*(9).

Niazi, M. (2004). *A Framework for Assisting the Design of Effective Implementation Strategies for Software Process Improvement*. Ph.D. dissertation, Faculty Inf. Technol., University of Technology Sydney, Ultimo, NSW, Australia.

Niazi, M., Hickman, C., Ahmad, R., & Ali Babar, M. (2008). A Model for Requirements Change Management: Implementation of CMMI Level 2 Specific Practice. In *Product-Focused Software Process Improvement* (pp. 143–157). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-69566-0_14

Nurmuliani, N., Zowghi, D., & Powell, S. (2004). Analysis of requirements volatility during software development life cycle. In *2004 Australian Software Engineering Conference. Proceedings.* (pp. 28–37). IEEE. https://doi.org/10.1109/ASWEC.2004.1290455

Nurmuliani, N., Zowghi, D., & Williams, S. P. (2006). Requirements volatility and its impact on change effort: Evidence-based research in software development projects. *Proceedings of the Eleventh Australian Workshop on Requirements Engineering*.

ODwyer, L. M., & Bernauer, J. A. (2014). *Quantitative research for the qualitative researcher*. Retrieved from https://uk.sagepub.com/en-gb/asi/quantitative-research-for-the-qualitative-researcher/book236699

Olsen, N. C. (1993). The software rush hour (software engineering). *IEEE Software*, *10*(5), 29–37. https://doi.org/10.1109/52.232394

Olson, D. (2014). Matrix Prioritization | BAwiki. Retrieved August 10, 2017, from http://www.bawiki.com/wiki/techniques/matrix-prioritization/

Oriol, M., Seppänen, P., Behutiye, W., Farré, C., Kozik, R., Martínez-Fernández, S., … Partanen, J. (2019). Data-Driven Elicitation of Quality Requirements in Agile Companies (pp. 49–63). Springer, Cham. https://doi.org/10.1007/978-3-030-29238-6_4

Paetsch, F., Eberlein, A., & Maurer, F. (2003). Requirements engineering and agile software development. In *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.* (pp. 308–313). IEEE Comput. Soc. https://doi.org/10.1109/ENABL.2003.1231428

Paganini, P. (2015). Companies Worldwide Are Adopting Agile Development Techniques | Veracode. Retrieved April 8, 2018, from https://www.veracode.com/blog/2015/03/companies-worldwide-are-adopting-agile-development-techniques-sw

Pellissier, R. (2007). *Business research made easy* (Ist Editio). Juta.

Penzenstadler, B., & Femmer, H. (2013). A generic model for sustainability with process- and product-specific instances. In *Proceedings of the 2013 workshop on Green in/by software engineering - GIBSE '13* (p. 3). New York, New York, USA: ACM Press. https://doi.org/10.1145/2451605.2451609

Perumal, S. A., & Kavitha, G. (2011). Changing Requirements – Correlated to Risk or Quality? *International Journal of Engineering and Technology*, *3*(1), 22–25. https://doi.org/10.7763/IJET.2011.V3.194

Petersen, K., & Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of Systems and Software*, *82*(9), 1479–1490. https://doi.org/10.1016/j.jss.2009.03.036

Phadermrod, B., Crowder, R. M., & Wills, G. B. (2019). Importance-Performance Analysis based SWOT analysis. *International Journal of Information Management*, *44*, 194–203. https://doi.org/10.1016/J.IJINFOMGT.2016.03.009

PMBOK® Guide. (2013). *A guide to the project management body of knowledge, Project Management Institute.* (Fifth Edit). Newtown Square, Pennsylvania: Project Management Institute, Inc.

Popli, R., & Chauhan, N. (2014). Cost and effort estimation in agile software development. In

2014 International Conference on Reliability Optimization and Information Technology (ICROIT) (pp. 57–61). IEEE. https://doi.org/10.1109/ICROIT.2014.6798284

Popli, R., Chauhan, N., & Sharma, H. (2014). Prioritising user stories in agile environment. In *2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)* (pp. 515–519). IEEE. https://doi.org/10.1109/ICICICT.2014.6781336

Poppendieck, M. (Mary B. ., & Poppendieck, T. D. (2003). *Lean software development : an agile toolkit*. Addison-Wesley.

Pressman, R. S. (2005). *Software engineering : a practitioner's approach*.

PSEB. (2005). *Pakistan's Software Industry Best Practices & Strategic Challenges, An Exploratory Analysis*. Retrieved from http://www.pseb.org.pk/

Quaglia, E. J., & Tocantins, C. A. (2011). Simulation projects management using Scrum. In *Proceedings of the 2011 Winter Simulation Conference (WSC)* (pp. 3421–3430). IEEE. https://doi.org/10.1109/WSC.2011.6148038

Racheva, Z., Daneva, M., & Buglione, L. (2008). Supporting the Dynamic Reprioritization of Requirements in Agile Development of Software Products. In *2008 Second International Workshop on Software Product Management* (pp. 49–58). IEEE. https://doi.org/10.1109/IWSPM.2008.7

Racheva, Z., Daneva, M., Herrmann, A., & Wieringa, R. J. (2010). A conceptual model and process for client-driven agile requirements prioritization. In *2010 Fourth International Conference on Research Challenges in Information Science (RCIS)* (pp. 287–298). IEEE. https://doi.org/10.1109/RCIS.2010.5507388

Racheva, Z., Daneva, M., Sikkel, K., Herrmann, A., & Wieringa, R. (2010). Do We Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study. In *2010 18th IEEE International Requirements Engineering Conference* (pp. 147–156). IEEE. https://doi.org/10.1109/RE.2010.27

Raharjo, H., Xie, M., & Brombacher, A. C. (2006). Prioritizing quality characteristics in dynamic quality function deployment. *International Journal of Production Research*, *44*(23), 5005–5018. https://doi.org/10.1080/00207540600547414

Rahman, A. (2015, March 23). *An Empirical Study on the Role of Requirement Engineering in Agile Method and Its Impact on Quality*. Retrieved from https://yorkspace.library.yorku.ca/xmlui/handle/10315/30059

Rahman, M. A., Razali, R., & Singh, D. (2014). A Risk Model of Requirements Change Impact Analysis. *Journal of Software*, *9*(1), 76–81. https://doi.org/10.4304/jsw.9.1.76-81

Raju, S., & Uma, G. V. (2014). A Quantitative Measurement of Software Requirement Factors using Goal Question Metric (GQM) Approach. *IOSR Journal of Computer Engineering Ver. VIII*, *16*(3), 2278–661. Retrieved from www.iosrjournals.org

Ramesh, B., Cao, L., & Baskerville, R. (2010). Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*, *20*(5), 449–480.

https://doi.org/10.1111/j.1365-2575.2007.00259.x

Rao, N. G. (2012). *Research Methodology and Quantitative Methods*. BS Publications.

Rauhamäki, J., & Kuikka, S. (2015). Strategies for hazard management process II. In *Proceedings of the 20th European Conference on Pattern Languages of Programs - EuroPLoP '15* (pp. 1–7). New York, New York, USA: ACM Press. https://doi.org/10.1145/2855321.2855325

Rebelo, M. F., Santos, G., & Silva, R. (2016). Integration of management systems: towards a sustained success and development of organizations. *Journal of Cleaner Production*, *127*, 96–111. https://doi.org/10.1016/J.JCLEPRO.2016.04.011

Rida, A., Nazir, S., Tabassum, A., & Asim, S. (2017). The Impact of Analytical Assessment of Requirements Prioritization Models: An Empirical Study. *International Journal of Advanced Computer Science and Applications*, *8*(2). https://doi.org/10.14569/IJACSA.2017.080240

Rogers, M. R., & Lopez, E. C. (2002). Identifying Critical Cross-Cultural School Psychology Competencies. *Journal of School Psychology*, *40*(2), 115–141. https://doi.org/10.1016/S0022-4405(02)00093-6

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, *14*(2), 131–164. https://doi.org/10.1007/s10664-008-9102-8

Sahota, M., Appelo, J., & Kniberg, H. (2012). *An agile adoption and transformation survival guide : working with organizational culture*.

Sara, A., & Aftab, S. (2017). Scrum with the Spices of Agile Family : A Systematic Mapping, (November), 58–72. https://doi.org/10.5815/ijmecs.2017.11.07

Schön, E.-M., Thomaschewski, J., & Escalona, M. J. (2017). Agile Requirements Engineering: A systematic literature review. *Computer Standards & Interfaces*, *49*, 79–91. https://doi.org/10.1016/j.csi.2016.08.011

Schön, E.-M., Winter, D., Escalona, M. J., & Thomaschewski, J. (2017). Key Challenges in Agile Requirements Engineering (pp. 37–51). Springer, Cham. https://doi.org/10.1007/978-3-319-57633-6_3

Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum*. Prentice Hall.

Sebega, Y., & Mnkandla, E. (2017). Exploring Issues in Agile Requirements Engineering in the South African Software Industry. *The Electronic Journal of Information Systems in Developing Countries*, *81*(1), 1–18. https://doi.org/10.1002/j.1681-4835.2017.tb00597.x

Sekaran, U., & Bougie, R. (2013). *Research methods for business : a skill-building approach*. Wiley.

Shafinah, M. (2015). *A Process Based Approach Software Certification Model for Agile And Secure Environment*. (Unpublished Doctoral Dissertation). Universiti Utara Malaysia, Kedah, Malaysia.

Shafinah, M., Fauziah, B., & Deraman, A. (2014). An exploratory study on agile based software development practices. *International Journal of Software Engineering and Its Applications*, *8*(5), 85–114. https://doi.org/10.14257/IJSEIA.2014.8.5.09

Shahir, H. Y., Daneshpajouh, S., & Ramsin, R. (2008). Improvement Strategies for Agile Processes: A SWOT Analysis Approach. In *2008 Sixth International Conference on Software Engineering Research, Management and Applications* (pp. 221–228). IEEE. https://doi.org/10.1109/SERA.2008.33

Shariatmadari, M., Sarfaraz, A. H., Hedayat, P., & Vadoudi, K. (2013). Using SWOT Analysis and Sem to Prioritize Strategies in Foreign Exchange Market in Iran. *Procedia - Social and Behavioral Sciences*, *99*, 886–892. https://doi.org/10.1016/J.SBSPRO.2013.10.561

Sheffield, J., & Lemétayer, J. (2013). Factors associated with the software development agility of successful projects. *International Journal of Project Management*, *31*(3), 459–472. https://doi.org/10.1016/j.ijproman.2012.09.011

Shneiderman, B. (1998). *Designing the user interface : strategies for effective human-computer-interaction*. Addison Wesley Longman. Retrieved from https://dl.acm.org/citation.cfm?id=523237

Sillitti, A., & Succi, G. (2005). Requirements Engineering for Agile Methods. In *Engineering and Managing Software Requirements* (pp. 309–326). Berlin/Heidelberg: Springer-Verlag. https://doi.org/10.1007/3-540-28244-0_14

SMEDA. (2010). *Pre-Feasibility Study:Small and Medium Enterprises Development Authority*. Retrieved from www.smeda.org.pk

Soares, H. F., Alves, N. S. R., Mendes, T. S., Mendonca, M., & Spinola, R. O. (2015). Investigating the Link between User Stories and Documentation Debt on Software Projects. In *2015 12th International Conference on Information Technology - New Generations* (pp. 385–390). IEEE. https://doi.org/10.1109/ITNG.2015.68

Solingen, R. van., & Berghout, E. (1999). *The goal/question/metric method : a practical guide for quality improvement of software development*. McGraw-Hill.

Sommerville, I. (2004). *Software engineering*. Pearson/Addison-Wesley.

Sommerville, I. (2007). *Software engineering*. Addison-Wesley. Retrieved from http://dl.acm.org/citation.cfm?id=1196763

Sommerville, I. (2011). *Software engineering* (9th ed.). Pearson Education.

Stålhane, T., Katta, V., & Myklebust, T. (2014). Change Impact Analysis in Agile Development. *SAFECOMP 2014 Workshops: ASCoMS, DECSoS, DEVVARTS, ISSE, ReSA4CI, SASSUR*.

Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.-B. (2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *Journal of Systems and Software*, *86*(6), 1663–1678. https://doi.org/10.1016/j.jss.2013.02.027

Stapleton, J., & DSDM Consortium. (2003). *DSDM : business focused development*. Addison-Wesley.

Sultana, S., Motla, Y. H., Asghar, S., Jamal, M., & Azad, R. (2014). A hybrid model by integrating agile practices for Pakistani software industry. In *2014 International Conference on Electronics, Communications and Computers (CONIELECOMP)* (pp. 256–262). IEEE. https://doi.org/10.1109/CONIELECOMP.2014.6808600

Tate, K., & Kevin. (2005). *Sustainable software development : an agile perspective*. Addison Wesley. Retrieved from http://dl.acm.org/citation.cfm?id=1076970

Tavakol, M., & Dennick, R. (2011). Making sense of Cronbach's alpha. *International Journal of Medical Education*, *2*, 53–55. https://doi.org/10.5116/ijme.4dfb.8dfd

Terwee, C. B., Bot, S. D. M., de Boer, M. R., van der Windt, D. A. W. M., Knol, D. L., Dekker, J., … de Vet, H. C. W. (2007). Quality criteria were proposed for measurement properties of health status questionnaires. *Journal of Clinical Epidemiology*, *60*(1), 34–42. https://doi.org/10.1016/j.jclinepi.2006.03.012

Thakurta, R. (2013). A framework for prioritization of quality requirements for inclusion in a software project. *Software Quality Journal*, *21*(4), 573–597. https://doi.org/10.1007/s11219-012-9188-5

Thayer, R. H., Dorfman, M. (Merlin), & Bailin, S. C. (1997). *Software requirements engineering*. IEEE Computer Society Press.

Thomas, S. (2008). Agile Change Management | It's a Delivery Thing. Retrieved February 22, 2017, from http://itsadeliverything.com/agile-change-management.

Venters, C. C., Jay, C., Lau, L., Griffiths, M., Holmes, V., Ward, R., … Xu, J. (2014). Software Sustainability: The Modern Tower of Babel. In *RE4SuSy: Third International Workshop on Requirements Engineering for Sustainable Systems*. CEUR. Retrieved from http://eprints.whiterose.ac.uk/84941/

Venters, C. C., Lau, L., Griffiths, M. K., Holmes, V., Ward, R. R., Jay, C., … Xu, J. (2014). The Blind Men and the Elephant: Towards an Empirical Evaluation Framework for Software Sustainability. *Journal of Open Research Software*, *2*(1). https://doi.org/10.5334/jors.ao

VersionOne. (2017). VERSIONONE: 11th Annual State of Agile Report. Retrieved from https://www.versionone.com/about/press-releases/versionone-releases-11th-annual-state-of-agile-report/

Wagner, S., Fernández, D. M., Felderer, M., & Kalinowski, M. (2017). Requirements engineering practice and problems in agile projects: results from an international survey. In *Ibero-American Conference on Software Engineering (CIbSE)*. PeerJ Inc. https://doi.org/10.7287/PEERJ.PREPRINTS.2038V1

Wan, J., & Wang, R. (2010). Empirical Research on Critical Success Factors of Agile Software Process Improvement. *Journal of Software Engineering and Applications*, *3*(12), 1131–1140. https://doi.org/10.4236/jsea.2010.312132

Wang, X., Zhao, L., Wang, Y., & Sun, J. (2014). The Role of Requirements Engineering Practices in Agile Development: An Empirical Study. *The Inaugural Asia Pacific Requirements Engineering Symposium*, *432*, 195–209. https://doi.org/10.1007/978-3-662-43610-3_15

Waters, K. (2009). Prioritization using MoSCoW.

Weihrich, H. (2017). SWOT Analysis: Bringing Internal and External Factors Together. Retrieved February 23, 2021, from https://www.business-to-you.com/swot-analysis/

West, D., & Grant, T. (2010). Agile Development: Mainstream Adoption Has Changed Agility. *Forrester Research, Inc.*, 1–20. Retrieved from http://www.ca.com/us/~/media/Files/IndustryResearch/forrester-agile-development-mainstream-adoption.pdf

Wiegers, K. (2017). Best Practices for Change Impact Analysis | Jama Software. Retrieved April 13, 2017, from http://www.jamasoftware.com/blog/change-impact-analysis-2/

Wiegers, K. E. (2003). *Software requirements*. Microsoft Press.

Wiegers, K. E., & Beatty, J. (2013). *Software requirements* (3rd Ed.). Microsoft Press, Washington 98052-6399.

Williams, L. (2007). A Survey of Agile Development Methodologies. *Univercsty of the West England*, 209–227. Retrieved from http://www.cems.uwe.ac.uk/~pchatter/2011/isd_hk/AgileMethods.pdf

Wohlin, C., & Aurum, A. (2015). Towards a decision-making structure for selecting a research design in empirical software engineering. *Empirical Software Engineering*, *20*(6), 1427–1455. https://doi.org/10.1007/s10664-014-9319-7

Yin, R. K. (2014). *Case study research : design and methods*. Retrieved from https://uk.sagepub.com/en-gb/asi/case-study-research/book237921

Yu, Y., & Sharp, H. (2011). Analysing requirements in a case study of pairing. In *Proceedings of the 1st Workshop on Agile Requirements Engineering - AREW '11* (pp. 1–6). New York, New York, USA: ACM Press. https://doi.org/10.1145/2068783.2068787

Zacarias, D. (2016). 20 Product Prioritization Techniques : Retrieved July 11, 2017, from https://foldingburritos.com/product-prioritization-techniques/

Zhu, J., Liang, Y., & Gu, Y. (2008). The Requirements Change Analysis for Different Level Users. In *2008 International Symposium on Intelligent Information Technology Application Workshops* (pp. 987–989). IEEE. https://doi.org/10.1109/IITA.Workshops.2008.204

Zhu, Y. (2009). *Requirements Engineering in an Agile Environment*. Institutionen för informationsteknologi. Retrieved from http://www.teknat.uu.se/student

Zikmund, W. G., Babin, B. J., Carr, J. C., & Griffin, M. (2013). *Business research methods* (9th editio). South-Western: Cengage Learning.

Appendix A

The Objective and Sources for Instrument Development

SECTION I. Demographic Information				
Objectives	**Variables**	**Questions**	**Contents**	**Sources**
To assess the qualification of respondents	Respondent details	1	Position in company	Shafinah (2015)
		2	Years of experience in Software Development	Shafinah (2015)
To study the organizational background	Organization details	3	Sector of organization	Fernández et al. (2017)
To investigate the respondents basic knowledge about Agile	Agile Knowledge and Experience	4	Agile experience	Shafinah (2015)
		5	Agile team member number	Shafinah (2015)
		6	Agile methodologies	VersionOne (2010)

SECTION II. Agile Requirement Engineering and Agile RCM Issues and Challenges				
Objectives	**Variables**	**Questions**	**Contents**	**Sources**
To determine the issues and challenges related to Agile RE	Agile RE issues and challenges	7(1)-7(11)	Common Issues in Agile RE	Wagner, Fernández, Felderer, and Kalinowski (2017)
		8	Common Challenges in Agile RE	Literature Review (Section 2.2.5, Table 2.5)
To determine the challenges and issues related to Agile RE To investigate the challenges and issues related to Agile RCM	Agile RE issues and challenges Agile RCM issues	9	Opinion on handling change	Literature Review (Section 2.3.2)
		10(1)-10(11)	Common issues in Agile RCM	Wagner, Fernández, Felderer, and Kalinowski (2017), Literature Review (Table 2.7)
		11	Common Challenges in Agile RCM	Literature Review - (Section 2.3.3, Table 2.7)

SECTION III. Current practices of Agile RCM				
Objectives	**Variables**	**Questions**	**Contents**	**Sources**
To investigate the current practices of Agile RCM.	Agile RCM practices	12	Dealing with changing requirement	Wagner et al. (2017)
		13	Opinion on change request	-
		14	Change requirement recorded	Alsalemi & Yeoh (2015)
		15	Change management regarding RE	Fernández et al. (2017)
		16	Product Backlog Change reasons	Alsalemi & Yeoh (2015)

SECTION IV. Sustainability Characteristics as a Non-Functional Requirement				
Objectives	**Variables**	**Questions**	**Contents**	**Sources**
To Investigate the method of elicitation and management of Non-functional requirement	Elicitation and Management	17	Elicitation and Management of Non-functional requirement	Literature Review - (Section 2.3.4)
To access the documentation of non-functional requirements	Non-functional requirement	18	Non-functional requirement document	Fernández et al. (2017)
To identify the Sustainability Characteristics as a Non-functional requirement.	Sustainability Characteristics	19(1)-19(8)	Sustainability Characteristics as a Non-functional requirement	(ISO/IEC 25012, 2008)
To investigate the importance of Sustainability characteristics and there sub- characteristics as a Non-functional requirement	Sustainability characteristics and sub-characteristics	20(1)-20(27)	Sustainability Characteristics and sub-characteristics as a Non-functional requirement	(ISO/IEC 25012, 2008)

SECTION V. Sustainability Analysis				
Objectives	**Variables**	**Questions**	**Contents**	**Sources**
To assess the impact of change	Impact Analysis	21	Opinion on Impact Analysis	(Wiegers, 2017)
	Analyse change effect	22	Requirement change analysis	Wagner et al. (2017)
	Change Impact	23(1)-23(3)	Change impact on effort, cost and duration	Literature Review - (Section 2.3.6.1)
	Estimation	24	Estimation to determine the impact of change	-
To investigate the risk of requirement change.	Risk analysis	25	Opinion on Risk Analysis	Literature Review - (Section 2.3.6.2)
	Risk type	26	Risks due to requirement change	Literature Review - (Section 2.3.6.2)
		27(1)-26(13)	Common risk factors	Literature Review - (Table 2.17)
		28(1)-27(13)	Risk factors related to sustainability characteristics	-
To map the sustainability characteristic with sustainable dimension.	sustainable dimension.	29(1)-28(8)	sustainability characteristic with sustainable dimension	-

SECTION V. Requirement prioritization technique used in ASD				
Objectives	**Variables**	**Questions**	**Contents**	**Sources**
To assess the decision-makers in the process prioritization.	Decision makers	30	Handling the process of prioritization	Racheva, Daneva, Sikkel, Herrmann, & Wieringa (2010)
To investigate the Prioritization techniques used after change request.	Prioritization techniques	31	Prioritization techniques used in ASD	(Achimugu et al., 2014) Literature Review - (Table 2.21)
To examine the evaluation criteria to perform requirement prioritization.	Evaluation criteria	32	Consideration criteria of requirement prioritization	Khan et al. (2015)
To investigate the prioritization criteria with techniques.	Evaluation criteria with techniques	33	Mapping prioritization criteria with requirement prioritization techniques.	Khari & Kumar (2013), Racheva, Daneva, Sikkel, Herrmann, & Wieringa (2010), Rida et al. (2017)

Appendix B

The Instrument

COLLEGE OF ARTS AND SCIENCES
UNIVERSITI UTARA MALAYSIA

SURVEY ON AGILE REQUIREMENT CHANGE MANAGEMENT

Dear Respected Respondent,

We are conducting a survey regarding ASD practitioner's expertise focusing on RCM, as part of our research work. Because you are one of the software practitioners, thus we would like to invite you to participate in this survey. Basically the objectives of this survey are:

(1) To investigate the issues and challenges of Agile RE and Agile RCM.

(2) To determine the current practices of Agile RCM.

(3) To investigate the software practitioner opinion on the sustainability characteristics as a non-functional requirement.

(4) To examine the software practitioner opinion on the importance of adopting sustainability analysis considering impact and risk factor.

(5) To select the suitable technique of prioritization after the change of requirement in ASD.

What you need to do:

Please tick (√) the most appropriate answer or **write** your rating accordingly. You are advised to answer the questions based on your knowledge and experience. We would appreciate it very much if you could answer the questions carefully as the information you provide will influence the accuracy and the success of this research. It will take

around 25-30 minutes to complete the questionnaire. All answers will be treated as strictly confidential and will be used for the purpose of the study only.

This survey consists of 4 sections:

SECTION I: DEMOGRAPHIC INFORMATION.
SECTION II: AGILE RE AGILE RCM (ARCM) ISSUES AND CHALLENGES.
SECTION III: CURRENT PRACTICES OF ARCM.
SECTION IV: SUSTAINABILITY CHARACTERISTICS AS A NON-FUNCTIONAL REQUIREMENT.
SECTION V: SUSTAINABILITY ANALYSIS.
SECTION VI: REQUIREMENT PRIORITIZATION TECHNIQUE USED IN ASD.

Thank you for your cooperation and the time taken in answering this questionnaire. If you have any questions regarding this research, you may address them to us at the contact details below.

Researcher Contact Detail: Najia Saher
Phone: Malaysia: +601112226491
Pakistan: +923336387864
Email: najiasaher@gmail.com

Research Supervisor Contact Detail:
Assoc. Prof. Dr. Fauziah Baharom
School of Computing, College of Arts and Sciences
Universiti Utara Malaysia,
(m): +60194741666 (o): +6049285093
Email: fauziah@uum.edu.my
Head of Software Technology (SofTech) Special Research Group

Respondent Details

Respondent's Name:	
Organization:	
Telephone No:	Fax No:
e-Mail:	Date:
Company's Website: (Optional)	

SECTION I: DEMOGRAPHIC INFORMATION

A. Respondent's Background

1. What best describes your current position in your company?

[ ] Project Manager
[ ] Product Owner
[ ] Quality Assurance/Tester/Auditor
[ ] Team Leader
[ ] Other (please specify):_____

[ ] Developer
[ ] Requirement Engineer
[ ] System Analyst
[ ] Consultant

2. How long have you been participating in software development?

[ ] < 3 year
[ ] 3-5 years
[ ] 6 – 10 years
[ ] 11-20 years
[ ] 21-29 years
[ ] 30 years and above

B. Organization's Background

3. What is the main business area of your company?

[ ] Software development (custom software)
[ ] Consulting / Project management support
[ ] Software development (standard software)
[ ] Others (please specify):_____

[ ] IT Consulting & Services
[ ] Telecommunication
[ ] Embedded Software Systems

C. Experience in Agile

4. How long have you been practicing Agile development methods?

[ ] < 2 year [ ] 2-3 years [ ] 3-5 years [ ] > 5 years

5. How many members does a team possess in your organization as an Agile team?

[ ] < 5 [ ] 5- 10 [ ] 11-20 [ ] 21-40 [ ] > 40

6. Which Agile methods you have practicing or you are familiar with?
(Check all that apply):

[ ] Scrum
[ ] Extreme Programming (XP)
[ ] Agile Modelling
[ ] Dynamic Systems Development Method (DSDM)
[ ] Other (please specify):_____

[ ] Future-Driven Development (FDD)
[ ] Crystal Family
[ ] Kanban

SECTION II: AGILE RE & AGILE RCM ISSUES AND CHALLENGES

A. Agile RE Issues and Challenges

7. Considering your personal experiences, how do the following problems in Requirements Engineering apply to your projects? (from 1: Strongly disagree to 7: Strongly agree).

Ranking:

1= Strongly Disagree, 2 = Disagree, 3 = More or Less Disagree, 4 = Neutral, 5 = More or less Agree, 6 = Agree, 7 = Strongly Agree

Common Issues in Agile RE	**Strongly Disagree ⟷ Strongly Agree**
Underspecified requirements that are too abstract and allow for various interpretations	1 2 3 4 5 6 7
Communication flaws between developers and the customer	1 2 3 4 5 6 7
Incomplete and/or hidden requirements	1 2 3 4 5 6 7
Implicit requirements not made explicit	1 2 3 4 5 6 7
Inconsistent requirements	1 2 3 4 5 6 7
Insufficient support by project lead	1 2 3 4 5 6 7
Insufficient support by customer	1 2 3 4 5 6 7
"Gold plating" (implementation of features without corresponding requirements)	1 2 3 4 5 6 7
Discrepancy between high innovation and need for formal acceptance of requirements	1 2 3 4 5 6 7
Unclear responsibilities	1 2 3 4 5 6 7
Technically unfeasible requirements	1 2 3 4 5 6 7

8. Considering your personal experience, what challenges do you face in Agile RE (ARE)?

No.	**Challenges in Agile RE**

B. Agile Requirement Change Management (RCM) Issues and Challenges

9. Considering the literature review it is determined that handling of requirement change without planning and analysis can influence the quality of software.

[ ] Disagree

[ ] Agree, (If you Agree, How your company perform the planning and analysis (please

specify_____

_____

10. Considering your personal experiences, how do the following problems in Requirements Change Management (RCM) apply to your projects? (From 1: Strongly disagree to 7: Strongly agree).

Ranking:

1= Strongly Disagree, 2 = Disagree, 3 = More or Less Disagree, 4 = Neutral, 5 = More or less Agree, 6 = Agree, 7 = Strongly Agree

Common Issues in Agile RCM	**Strongly Disagree** ⟷ **Strongly Agree**						
Project Scope change							
Moving targets (changing goals, business processes and/or requirements)	1	2	3	4	5	6	7
Budget and schedule estimation							
Due to the nature of incorporating Requirement Change(s) in subsequent iterations, it is not possible to make upfront estimations, which can result in budget and schedule overruns.	1	2	3	4	5	6	7
Requirement Prioritization							
Prioritization on single dimension (business value)	1	2	3	4	5	6	7
Evaluating the consequences of change							
Lack of method of evaluating the consequences of changes.	1	2	3	4	5	6	7
Customer involvement							
Weak access to customer needs and /or (internal) business information	1	2	3	4	5	6	7
Weak knowledge of customer's application domain	1	2	3	4	5	6	7
Weak relationship between customer and project lead	1	2	3	4	5	6	7
Volatile customer's business domain regarding, e.g., changing points of contact, business processes or requirements	1	2	3	4	5	6	7
Face-to-face communication							
Unavailability and unwillingness of the required customer	1	2	3	4	5	6	7
Iterative development							
Time boxing/Not enough time in general	1	2	3	4	5	6	7
Non-Functional Requirement							
Unclear/unmeasurable non-functional requirements	1	2	3	4	5	6	7

11. Considering your personal experience, what challenges do you face in Agile RCM?

No.	**Challenges in Agile RCM**

SECTION III: CURRENT PRACTICES OF ARCM

---

12. How do you deal with changing requirements after the initial release in Agile?

[ ] Freeze baseline when the iteration is under process
[ ] Introduces high-priority changes immediately and update product backlog
[ ] Only work with change request
[ ] Regularly change the requirement specification
[ ] Other (please specify):_____

13. How do you serve request of changes in Agile software development?

[ ] Change request form [ ] Ad hoc basis
[ ] Other (please specify):_____

14. How the requests of changed requirements are recorded in Agile?

[ ] Modify existing requirements (No Record) [ ] Record as new requirement
[ ] Make version of existing requirement [ ] Other (please specify):_____

15. How is the change management defined regarding Requirements Engineering in Agile?

[ ] Continuous change management as part of Agile RE approach
[ ] Change management approach that applies after formally accepting a requirements specification
[ ] Change management that applies during RE
[ ] Do not consider a change management in RE

16. What are the common reasons of change in Product Backlog?

[ ] Defect Fixing [ ] Scope Reduction [ ] Missing Requirement
[ ] Redundant functionality [ ] Functionality Enhancement [ ] Obsolete Functionality
[ ] Product Strategy [ ] Design Improvement [ ] Erroneous Requirements
[ ] Resolving Conflicts [ ] Clarifying Requirements
[ ] Other (please specify):_____

SECTION IV: SUSTAINABILITY CHARACTERISTICS AS A NON-FUNCTIONAL REQUIREMENT

Note: Sustainable software refers to the software that developed with the resources use aims to meet the needs of present generation until future with integrating the aspects of environment, economic and social towards long living software. One way to achieve sustainable software development is to link sustainability with the quality of the software product as a Non-Functional Requirement.

17. Does your company use any specific technique for elicitation and management of non-functional requirements during Agile software development?

[ ] Yes [ ] No

18. How do you document Non-functional requirements?

[ ] Quantified textual requirements [ ] Non-quantified textual requirements

[ ] Other (please specify):_____

19. Please indicate your perception on the **importance** of the following Sustainability characteristics as a Non-functional requirement during the process of requirement change. (From 1: Not at all important to 7: Extremely important).

Ranking:

1= Not at all important, 2 = Low importance, 3 = Slightly important, 4= Neutral, 5 = Moderately important, 6 = Very important, 7 = Extremely important

Sustainability Characteristics as a Non-Functional Requirement	**Not at all Important ⟷ Extremely Important**
Performance Efficiency *(the performance relative to the amount of resources used under stated conditions)*	1 2 3 4 5 6 7
Compatibility (a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.)	1 2 3 4 5 6 7
Usability (Degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.)	1 2 3 4 5 6 7
Reliability (Degree to which a system, product or component performs specified functions under specified conditions for a specified period of time.)	1 2 3 4 5 6 7
Security (degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.)	1 2 3 4 5 6 7
Maintainability (This characteristic represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in environment, and in requirements.)	1 2 3 4 5 6 7
Portability (Degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.)	1 2 3 4 5 6 7
Functional Suitability *(the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions)*	1 2 3 4 5 6 7
Other Important **Sustainability characteristics** to consider (please identify, if any):	1 2 3 4 5 6 7

20. Please indicate your perception on the **importance** of the following Sustainability characteristics and there sub- characteristics as a Non-functional requirement during the process of requirement change.

Performance Efficiency	**Not at all Important ⟷ Extremely Important**
Time behaviour (*The capacity of a product while performing its functions to meet requirements regarding response time, throughput and processing time)*	1 2 3 4 5 6 7
Resource utilization (*When performing its functions, the amounts and types of resources used by a product or system to meet requirements.*	1 2 3 4 5 6 7
Other Important sub-characteristics related to **Performance Efficiency** (please identify, if any):	
Compatibility	**Not at all Important ⟷ Extremely Important**
Interoperability *(degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.)*	1 2 3 4 5 6 7
Communication commonality (*The degree to which software is dependent on its associated hardware.)*	1 2 3 4 5 6 7
Data commonality (*The use of standard data representation*.)	1 2 3 4 5 6 7
Other Important sub-characteristics related to **Compatibility** (please identify, if any):	
Usability	**Not at all Important ⟷ Extremely Important**
Learnability *(degree to which a product or system enables the user to learn how to use it with effectiveness, efficiency in emergency situations.)*	1 2 3 4 5 6 7
Understandability *(The ability to recognize whether the software is appropriate for the user needs.)*	1 2 3 4 5 6 7
Operability (*The capacity of a product that make it easy to operate and control.)*	1 2 3 4 5 6 7
Attractiveness (*User interface assists satisfying and pleasing interaction for the user.)*	1 2 3 4 5 6 7
Other sub-characteristics related to **Usability** (please identify, if any):	
Reliability	**Not at all Important ⟷ Extremely Important**
Availability (*The capacity of a product or component to be accessible and operational at the time of use.)*	1 2 3 4 5 6 7
Fault tolerance (*The capacity of a product to be operated as required in spite of the existence of hardware or software faults.)*	1 2 3 4 5 6 7
Recoverability (*The capacity of a product to recover the data and restore the state of the system during the event of an interruption or a failure.)*	1 2 3 4 5 6 7
Other sub-characteristics related to **Reliability** (please identify, if any):	
Security	**Not at all Important ⟷ Extremely Important**
Confidentiality *(degree to which the prototype ensures that data are accessible only to those authorized to have access.)*	1 2 3 4 5 6 7
Integrity *(degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data.)*	1 2 3 4 5 6 7
Authenticity *(degree to which the identity of a subject or resource can be proved to be the one claimed.)*	1 2 3 4 5 6 7
Other sub-characteristics related to **Security** (please identify, if any):	

Maintainability	**Not at all Important** ⟷ **Extremely Important**
Modularity (*A discrete component that does not have an impact on other components during change.)*	1 2 3 4 5 6 7
Analysability (*A discrete component that does not have an impact on other components during change. Diagnose a product for deficiencies or causes of failures due to change.)*	1 2 3 4 5 6 7
Modifiability (*Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.)*	1 2 3 4 5 6 7
Extensibility (*The level of effort required to implement the extension.)*	1 2 3 4 5 6 7
Reusability (*The capacity of a module or component that can be re-used in other software system.)*	1 2 3 4 5 6 7
Testability (*The capacity to construct the test criteria with effectiveness and perform tests to meet those criteria.)*	1 2 3 4 5 6 7
Other sub-characteristics related to **Maintainability** (please identify, if any):	
Portability	**Not at all Important** ⟷ **Extremely Important**
Adaptability (*The capacity of product or system to be adopted in different hardware, software or other operational environments.)*	1 2 3 4 5 6 7
Installability (*The capacity of a product to be successfully installed or uninstalled in any environment.)*	1 2 3 4 5 6 7
Replaceability (*The capacity of replacement of a product by another software product in the same environment.)*	1 2 3 4 5 6 7
Other sub-characteristics related to **Portability** (please identify, if any):	
Functional Suitability	**Not at all Important** ⟷ **Extremely Important**
Functional Completeness *(degree to which the set of functions covers all the specified tasks and user objectives.)*	1 2 3 4 5 6 7
Functional Correctness *(degree to which the functions provides the correct results with the needed degree of precision.)*	1 2 3 4 5 6 7
Functional Appropriateness (degree to which the functions facilitate the accomplishment of specified tasks and objectives.)	1 2 3 4 5 6 7
Other sub-characteristics related to **Functional Suitability** (please identify, if any):	

SECTION V: SUSTAINABILITY ANALYSIS

***Sustainability Analysis**:* The sustainability analysis has been performed by incorporating risk analysis and impact analysis that systematically assesses and manages risks and its impact to support the decision making related to change. The sustainability analysis will go around the risk and impact factor.

A. Impact Analysis

21. Impact Analysis should be performed to analyze the effect of requirement changes. (If you Disagree, Please give justification).

[ ] Agree

[ ] Disagree, please justify:_____

_____

22. How do you analyse the effect of changes to requirements? (You can select more than one option if applicable)

[ ] Impact analysis between requirement. [ ] Risk analysis between requirement.

[ ] Do not analyse the effect of changes to requirement.

[ ] Other (please specify):_____

_____

23. Please indicate your perception on the importance of the following significant impact due to requirement change. (From 1: Not at all important to 7: Extremely important).

Ranking:

1= Not at all important, 2 = Low importance, 3 = Slightly important, 4= Neutral, 5 = Moderately important, 6 = Very important, 7 = Extremely important

Impact of change	**Not at all Important** ⟷ **Extremely Important**						
Development Effort (Cost)	1	2	3	4	5	6	7
Project duration	1	2	3	4	5	6	7
Quality of a project/product	1	2	3	4	5	6	7
Other Important Significant Impact**(s)** you encounter during requirement change (please identify, if any) :							

24. Do you perform estimation to determine the impact of change?

[ ] Cost estimation [ ] Resource estimation [ ] Effort estimation

[ ] Other (please specify):_____

B. Risk Analysis

25. Risk Analysis should be performed to make a decision about requirement change. (If you Disagree, please give justification).

[ ] Agree

[ ] Disagree. Please justify:_____

_____

26. What are the risk types that you confront during requirement change in Agile Software development? (You can select more than one option if applicable)

[ ] Business risks [ ] Technical risks [ ] Schedule risks
[ ] Cost risks [ ] Quality risks [ ] People risks
[ ] Project risks [ ] Other (please specify):_____

27. Please indicate your perception on the importance of the Risk factors that you encounter during requirement change in Agile. (From 1: Not at all important to 7: Extremely important).

List of Risk	Not at all Important ⟷ Extremely Important						
Errors/bugs during development and after release.	1	2	3	4	5	6	7
Create defects that are difficult and expensive to fix	1	2	3	4	5	6	7
Unstable requirements and time-consuming changes	1	2	3	4	5	6	7
Market loses	1	2	3	4	5	6	7
Customer related risk	1	2	3	4	5	6	7
Limit the productivity of internal and external users	1	2	3	4	5	6	7
Business impact risk	1	2	3	4	5	6	7
Technical issues risk	1	2	3	4	5	6	7
Unclear product requirements.	1	2	3	4	5	6	7
Budget overrun	1	2	3	4	5	6	7
Schedule overrun	1	2	3	4	5	6	7
Not accurate to the stakeholders need.	1	2	3	4	5	6	7
Failure to meet performance criteria	1	2	3	4	5	6	7
Other Important **Risk Factor(s)** you encounter during requirement change (please identify, if any) :							

28. Please tick (√) the most appropriate sustainability characteristics that is linked with Risk. (You can select more than one sustainability characteristics for one risk)

List of Risk	Sustainability Characteristics							
	Performance Efficiency	Compatibility	Usability	Reliability	Security	Maintainability	Portability	Functional Suitability
Errors/bugs during development and after release.								
Create defects that are difficult and expensive to fix								
Unstable requirements and time-consuming changes								
Market loses								
Customer related risk								
Limit the productivity of internal and external users								
Business impact risk								
Technical issues risk								
Unclear product requirements.								
Budget overrun								
Schedule overrun								
Not accurate to the stakeholders need.								
Failure to meet performance criteria								

Note: Sustainability is a multi-dimensional structure. The **Environmental** dimension is long-time usage of systems and the evolution with changing surrounding conditions can be easily adapted to future change. The **Social** dimension refers the societal communities like organizations, groups of people and the factors effects of software systems on the society. The **Economic** dimension focuses on assets, added value, and capital that ensure stakeholders long term investments are safe from economic risks.

29. Considering your personal experiences, which sustainability characteristic is mapped with the sustainable dimension of **Environmental**, **Social and Economic**.

Sustainability Characteristics	**Sustainability Dimension**		
	Economic	**Social**	**Environmental**
Performance Efficiency			
Compatibility			
Usability			
Reliability			
Security			
Maintainability			
Portability			
Functional Suitability			

SECTION VI: REQUIREMENT PRIORITIZATION TECHNIQUE USED IN ASD.

---

30. Who are the decision-makers in the prioritization process?

[ ] Client [ ] Development team [ ] Product Owner
[ ] Project Manager [ ] Team Leader [ ] Consultant
[ ] Other (please specify):_____

31. Which of the prioritization technique you used to prioritize requirement after requirement change in Agile? (You can select more than one option if applicable)

[ ] Analytic hierarchy process (AHP) [ ] Binary Search Tree (BST)
[ ] Cost-value ranking [ ] Cumulative Voting
[ ] Kano Model [ ] MoSCoW
[ ] Planning Game [ ] Pair wise analysis
[ ] Wiegers' matrix approach [ ] Quality Functional Deployment (QFD)
[ ] Value-oriented prioritization (VOP)
[ ] Other (please specify):_____

32. Which aspect or evaluation criteria do you use to perform decisions during agile prioritization? (You can select more than one option if applicable)

[ ] Complexity/ Ease of Use [ ] Time consumed [ ] Consistency
[ ] Reliability of result [ ] Strategic Importance [ ] Business value
[ ] Customer preference [ ] Benefit [ ] Penalty
[ ] Cost [ ] Voice of the customer [ ] Technical risk
[ ] Judgments on participants experiences [ ] Other (please specify):_____

33. Considering your personal experiences, Please tick (√) the most appropriate Requirement prioritization criteria for the following prioritization technique. (You can select more than one criteria for each prioritization technique).

Requirement Prioritization Technique	**Requirement prioritization criteria**							
	Complexity/ Ease of Use	**Time Consumed**	**Consistency**	**Reliability of result**	**Customer preference**	**Business value**	**Benefit/Cost/ Penalty**	**Technical risk**
Analytic hierarchy process (AHP)								
Binary Search Tree (BST)								
Cost-value ranking								
Cumulative Voting								
Kano Model								
MoSCoW								
Planning Game								
Pair wise analysis								
Quality Functional Deployment (QFD)								
Value-oriented prioritization (VOP)								
Wiegers' matrix approach								

Note:

Could you indicate us one or more persons to answer this survey? Preferably someone who is/are involved in software development and enforces Agility and RCM practices throughout software development (Full name and email/address/telephone number)

----------END OF SURVEY----------

Thank You for participating in the survey

Appendix C

Expert Review Instrument

SCHOOL OF COMPUTING, COLLEGE OF ARTS AND SCIENCES
UNIVERSITI UTARA MALAYSIA

VERIFICATION BY EXPERT REVIEW ON AGILE REQUIREMENT CHANGE MANAGEMENT (ARCM) MODEL

Dear Respected Respondent,

We are conducting an expert reviews as a research activity to evaluate the proposed model. The aim is to verify and ensure that the proposed ARCM model is built correctly. Hence, we would appreciate if you can participate as one of the expert for this study. As an expert you are in ideal position to give us valuable information from your own perspective on the proposed model. You are provided with the comprehensive report of the proposed model.

You need to send us the answers within **two weeks** from receiving the reports.

You are advised to answer the questions based on your knowledge and experience. We would appreciate it very much if you could answer the questions carefully as the information you provide will influence the accuracy and the success of this research. It will take around 1 hour to complete the review. All answers will be treated as strictly confidential and will be used for the purpose of the study only.

There are three (3) sections in this expert review such as below:

Section 1: Overall verification of ARCM Model using PDCA phases, activities and ARCM practices.

Section 2: Categorization/classification of the of the requirement change.

Section 3: Sustainability characteristics and sub-characteristics as a non-Functional requirement.

Respondent Details

Respondent's Name:	
Organization:	
Telephone No:	Fax No:
e-Mail:	Date:
Company's Website: (Optional)	

Thank you for your cooperation and the time taken in answering this expert review. If you have any questions regarding this research, you may address them to us at the contact details below.

Researcher Contact Detail:

Najia Saher

Phone: Malaysia: +601112226491

Pakistan: +923336387864

Email: najiasaher@gmail.com

Research Supervisor Contact Detail:

Assoc. Prof. Dr. Fauziah Baharom

School of Computing, College of Arts and Sciences

Universiti Utara Malaysia,

(m): +60194741666 (o): +6049285093

Email: fauziah@uum.edu.my

Dr.Rohaida Romli

Email : aida@uum.edu.my

Telephone : +604 928 5063

Appendix D

The Overall Verification Form

Purpose of verification

The verification is performed through experts review method to ensure that the main components in the proposed ARCM Model, are built correctly. The verification is performed in order to verify whether the proposed model conforms its specification. In this study, the verification stage is to verify the:

Section 1: Overall verification of ARCM Model using PDCA phases, activities and ARCM practices.

Section 2: Categorization/classification of the of the requirement change.

Section 3: Sustainability characteristics and sub-characteristics as a non-Functional requirement.

The PDCA processes, activities and RCM Practices are verified on the basis of their accurateness, comprehensiveness, understandability, and well organized. These criteria are adapted from previous studies Al-Tarawneh (2014), Kunda (2003), and Shafinah (2015). The description of the verification criteria is as follows.

Accurate	The process, activities and practices of ARCM Model are adequately decomposed to achieve the process of change in Agile.
Comprehensive	All the process, activities and practices of manging requirement change are included.
Understandable	The process, activities and practices of ARCM Model are decomposed clearly and unambiguously.
Well Organized	The process, activities and components of ARCM Model are organized well.

Accurate	Comprehensive	Understandable	Well organized
The process, activities and practices of ARCM Model are adequately/correctly decomposed to achieve the process of change in Agile.	All the process, activities and practices of manging requirement change are included in the Model.	The process, activities and practices of ARCM Model are decomposed clearly and unambiguously	The process, activities and practices of ARCM Model are organized Well.
Agree ☐ Disagree ☐ Comments/ Suggestions: _____ _____ _____	Agree ☐ Disagree ☐ Comments/ Suggestions: _____ _____ _____	Agree ☐ Disagree ☐ Comments/ Suggestions: _____ _____ _____	Agree ☐ Disagree ☐ Comments/ Suggestions: _____ _____ _____

Overall Comments/ Suggestions:

_____

_____

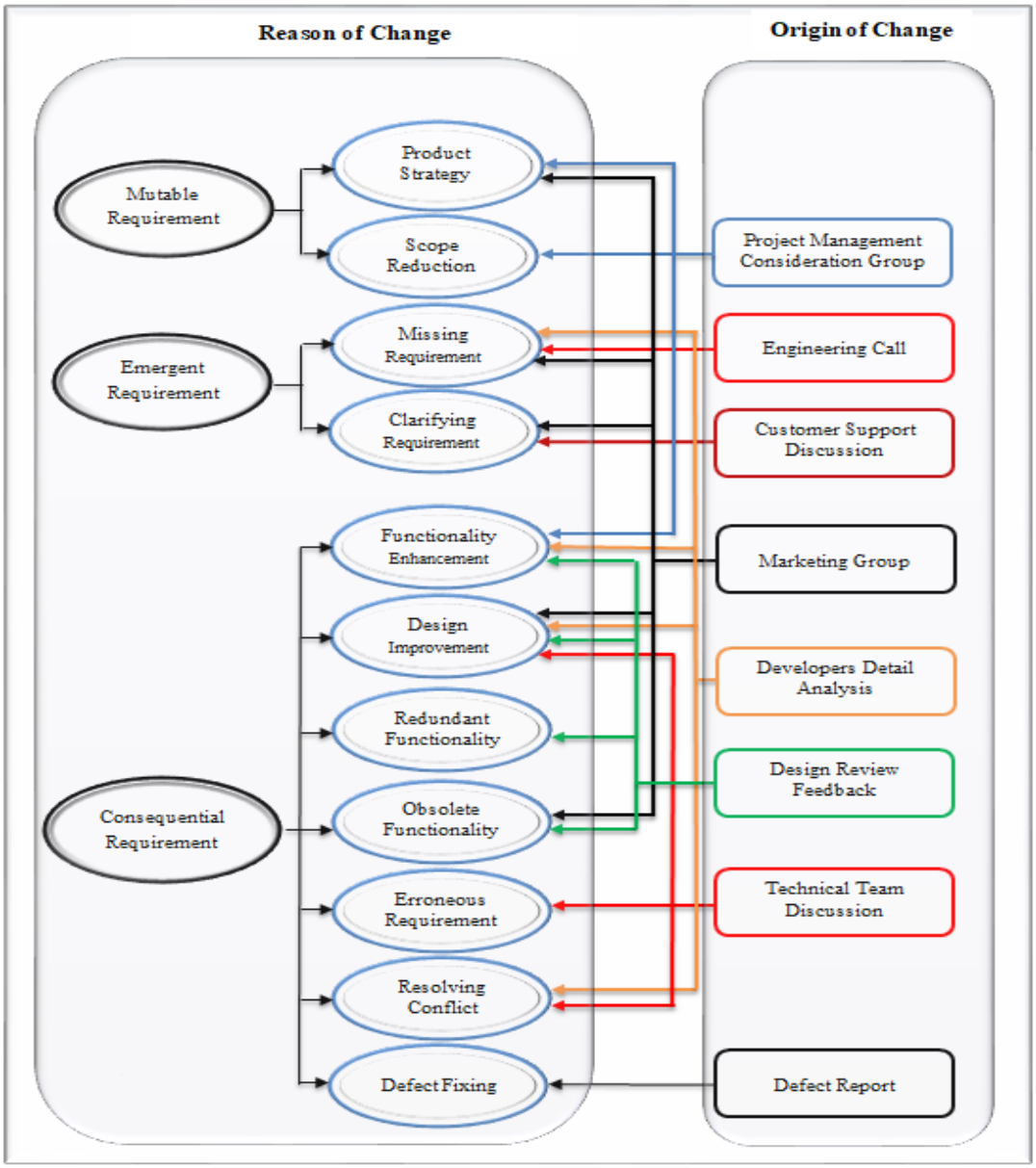
_____

_____

Appendix E

Requirement Change Categorization/Classification

The main objective of Requirement change identification is to create requirement change taxonomy and further categorize the requirement change element on the basis of "*reason"* and "*origin"* of change, for a better understanding of change request. Change categorization will be referred to propose a model of requirement change management in ASD.



Requirement Change Identification

The requirement change categorization/ classification on the basis of reason and origin of change has been arranged correctly.

Change Identification	**Comments/Suggestions**
To identify the request of change (RFC) referring to change categorization /classification on the basis of 'reason' and 'origin' of change has been arranged correctly.	Agree ☐ Disagree ☐ Comments/ Suggestions: _____ _____ _____ _____ _____ _____ _____

Overall Comments/ Suggestions:

_____

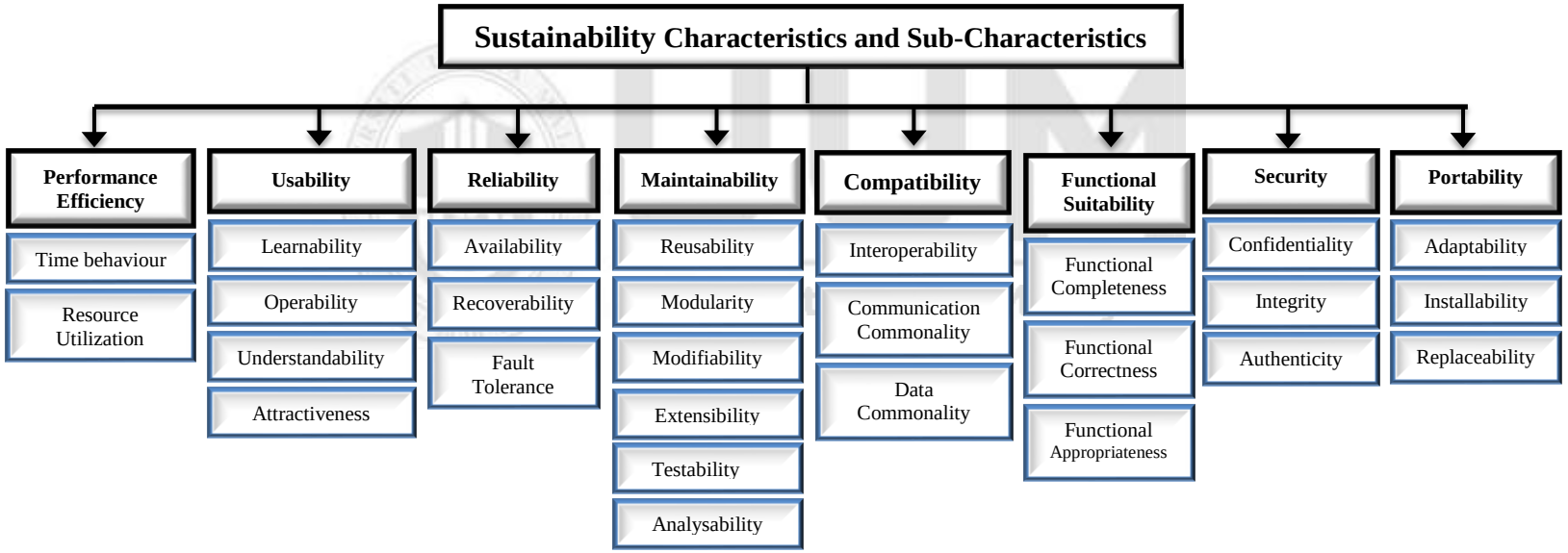
_____

_____

_____

Appendix F

Sustainability Characteristics and Sub-characteristics



Sustainability Sub-Characteristics Descriptions

Char	Sub-Characteristics	Definition	Resources
Performance Efficiency	Time behaviour	The capacity of a product while performing its functions to meet requirements regarding response time, throughput and processing time.	(ISO/IEC 25012, 2008)
	Resource utilization	When performing its functions, the amounts and types of resources used by a product or system to meet requirements.	(ISO/IEC 25012, 2008)
Usability	Learnability	To capacity to learn the use of a system in an efficient way with complete user satisfaction.	(ISO/IEC 25012, 2008)
	Operability	The capacity of a product that make it easy to operate and control.	(ISO/IEC-9126-1, 2001)
	Understandability	The ability to recognize whether the software is appropriate for the user needs.	(ISO/IEC-9126-1, 2001; ISO/IEC 25012, 2008)
	Attractiveness	User interface assists satisfying and pleasing interaction for the user.	(ISO/IEC 25012, 2008)
Reliability	Availability	The capacity of a product or component to be accessible and operational at the time of use.	(ISO/IEC 25012, 2008)
	Recoverability	The capacity of a product to recover the data and restore the state of the system during the event of an interruption or a failure.	(ISO/IEC 25012, 2008)
	Fault tolerance	The capacity of a product to be operated as required in spite of the existence of hardware or software faults.	(ISO/IEC-9126-1, 2001; ISO/IEC 25012, 2008)
Maintainability	Reusability	The capacity of a module or component that can be re-used in other software system.	(ISO/IEC 25012, 2008)
	Modularity	A discrete component that does not have an impact on other components during change.	(ISO/IEC 25012, 2008)
	Modifiability	Capacity to effectively and efficiently modified product or system without degrading the quality or introducing new defects.	(ISO/IEC 25012, 2008)
	Extensibility	The level of effort required to implement the extension.	(Grady & B., 1992)
	Testability	The capacity to construct the test criteria with effectiveness and perform tests to meet those criteria.	(ISO/IEC 25012, 2008)
	Analysability	Analyse the effect of intended change or identify any short comings or reasons of failures.	(ISO/IEC 25012, 2008)
Compatibility	Interoperability	Two or more systems, products or components can exchange and uses information that has been exchanged.	(ISO/IEC 25012, 2008)
	Communication commonality	The degree to which software is dependent on its associated hardware.	(McCall et al., 1977)
	Data commonality	The use of standard data representation.	(McCall et al., 1977)

Functional Suitability	Functional Completeness	The degree to which the set of functions covers all the specified tasks and user objectives.	(ISO/IEC 25012, 2008)
	Functional Correctness	The degree to which the functions provides the correct results with the needed degree of precision	(ISO/IEC 25012, 2008)
	Functional Appropriateness	The degree to which the functions facilitate the accomplishment of specified tasks and objectives.	(ISO/IEC 25012, 2008)
Security	Confidentiality	The degree to which the prototype ensures that data are accessible only to those authorized to have access.	(ISO/IEC 25012, 2008)
	Integrity	The degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data.	(ISO/IEC 25012, 2008)
	Authenticity	The degree to which the identity of a subject or resource can be proved to be the one claimed.	(ISO/IEC 25012, 2008)
Portability	Adaptability	The capacity of product or system to be adopted in different hardware, software or other operational environments.	(ISO/IEC 25012, 2008)
	Installability	The capacity of a product to be successfully installed or uninstalled in any environment.	(ISO/IEC 25012, 2008)
	Replaceability	The capacity of replacement of a product by another software product in the same environment.	(ISO/IEC 25012, 2008)

Sustainability characteristics and sub-characteristics

The sustainability characteristics and sub-characteristics as a Non-Functional Requirement are adequate to cover the requirement change in Agile.

Comments/Suggestions		
Agree		
Disagree		
Comments/ Suggestions: _____ _____ _____ _____ _____ _____ _____		

Overall Comments/ Suggestions:

_____

_____

_____

_____

Appendix G

Assessment of the Agile RCM Process

Metric Factor: Process- Completeness Metrics

Metrics(M)	Yes	No	Suggestion/Change (if needed)
M1: The request of change is identified by considering the source and reason of changes			
M2: Sustainability analysis was performed using Risk matrix			
M3: After change prioritization was performed using the guideline provided.			
M4: Change are implemented successfully			

Metric Factor: Process- Consistency Metrics

Metrics(M)	Yes	No	Suggestion/Change (if needed)
M5: Changes are identified by considering change categorization appropriately.			
M6: Requirement change analysis was performed by following the standards of requirement change analysis.			
M7: Requirement prioritization was done after every change.			
M8: Appropriate procedure was followed to implement the requirement change.			

Metric Factor: Process- Accuracy

Metrics(M)	Yes	No	Suggestion/Change (if needed)
M9: The requirement change identified by categorization/ classification on the basis of reason and origin of change has arranged correctly.			
M10: Appropriate procedure was used to analyse the requirement change.			
M11: Appropriate procedure was followed to select prioritization technique after requirement change.			
M12: Changes were implemented by following all the standard activities of requirement change.			

Assessment of the Critical Success Factor-CSF of Agile Project Management-APM

Ranking: The score ranged from 0 to 4 Likert scale

where: 0 = Never, 1 = Rarely, 2 = Sometimes, 3 = Often and 4 = Always.

Metric Factor: People

Metrics(M)	Score				
M13: Team capability Team members are capable to follow the process of Agile.	**0**	**1**	**2**	**3**	**4**
M14: Motivated team members: The current progress of iteration was revealed to everyone in the team during iteration.	**0**	**1**	**2**	**3**	**4**
M15: Agile process knowledge. All the Team members have the basic Agile process knowledge	**0**	**1**	**2**	**3**	**4**
M16: Self-organizing teamwork. Team members have the autonomous and make any decision related to change.	**0**	**1**	**2**	**3**	**4**
M17: Good customer relationship: Customer and end-user involvement were monitored in change activity	**0**	**1**	**2**	**3**	**4**
M18: Customer involvement: Customers or Project Owner (PO) was available on-site for face-to-face discussions during the requirement change	**0**	**1**	**2**	**3**	**4**

Metric Factor: Project

Metrics(M)	Score				
M19: The nature of project was Non-life-critical.	**0**	**1**	**2**	**3**	**4**
M20: Within the project scope varies with emergent requirement.	**0**	**1**	**2**	**3**	**4**
M21: Project was implemented with small team	**0**	**1**	**2**	**3**	**4**

Metric Factor: Organizational

Metrics(M)	Score				
M22: Organizational culture and mind set support frequent requirement change in Agile.	**0**	**1**	**2**	**3**	**4**
M23: Management of company support the process of change.	**0**	**1**	**2**	**3**	**4**

Metric Factor: Technology

Metrics(M)	Score				
M24: The Agile software techniques were implemented accurately.	**0**	**1**	**2**	**3**	**4**
M25: The software product are delivered accurately.	**0**	**1**	**2**	**3**	**4**

Can these metric be applied during the development process without consuming time?

_____

_____

_____

_____

_____

Appendix H

The Validation Form

Please validate and give comments on the below mentioned issues on the ARCM Model's implementation:

Perceived Ease of Use	Extremely Agree	Moderately Agree	Agree	Neutral	Extremely Disagree	Moderately Disagree	Disagree
Representation of ARCM model is simple	☐	☐	☐	☐	☐	☐	☐
It is simple to understand the process, activities and practices of Agile Requirement change Management (ARCM).	☐	☐	☐	☐	☐	☐	☐
It is simple to understand and apply sustainability Analysis during change management process.	☐	☐	☐	☐	☐	☐	☐
It is simple to implement ARCM model for change management process in Agile development.	☐	☐	☐	☐	☐	☐	☐
ARCM Model needs training to fully understand it	☐	☐	☐	☐	☐	☐	☐

Perceived Usefulness	Extremely Agree	Moderately Agree	Agree	Neutral	Extremely Disagree	Moderately Disagree	Disagree
Requirement change in Agile was difficult to perform without ARCM Model.	☐	☐	☐	☐	☐	☐	☐
Using ARCM Model gives the greater control over Requirement Change Management (RCM) process.	☐	☐	☐	☐	☐	☐	☐
ARCM Model enables to accomplish RCM process more quickly and save the time.	☐	☐	☐	☐	☐	☐	☐
Using ARCM Model enhances the effectiveness of requirement change process in Agile.	☐	☐	☐	☐	☐	☐	☐
Using ARCM Model improves the quality of the RCM process.	☐	☐	☐	☐	☐	☐	☐
Using ARCM Model makes it easier to do change process.	☐	☐	☐	☐	☐	☐	☐
Using ARCM Model increases the overall productivity.	☐	☐	☐	☐	☐	☐	☐
Structure of ARCM Model	**Extremely Agree**	**Moderately Agree**	**Agree**	**Neutral**	**Extremely Disagree**	**Moderately Disagree**	**Disagree**
The core components of ARCM are self-exploratory and no need of further explanation to be used effectively.	☐	☐	☐	☐	☐	☐	☐
The ARCM components are practical and could be used in software industry practicing Agile.	☐	☐	☐	☐	☐	☐	☐

The implementation of ARCM could assist an organization to identify issues related to requirement change.	☐	☐	☐	☐	☐	☐	☐
The components of ARCM model are enough to perform the complete process of change in Agile software development.	☐	☐	☐	☐	☐	☐	☐
Would you like to suggest further improvements or suggestions for ARCM Model?							
Are there any additional components for ARCM model, also give the reasons.							

Overall Comments/ Suggestions:

_____

_____

_____

Appendix I

Guideline Document for ARCM Model

Overview of the study

The purpose of this research is to improve the existing model for requirement change management that can provide a better support of requirement changes in Agile Software Development (ASD).

The study will explore the options to streamline the process of handling requirement changes in Agile due to the lack of formal process to manage requirement changes in ASD. Therefore, based on the review, this study found four main issues related to Agile Requirement Change Management (RCM): (1) the first issue is the need to employ the formal process to manage continuous requirement changes in Agile, (2) secondly, there is a need to construct a model that provides sustainable software development by considering sustainability characteristics as a non-functional requirement, (3) third issue is the need to improve the requirement change analysis step during the process of RCM, and (4) the fourth issue is related to the dynamic decision making using requirement re-prioritization after the process of requirement change. There are several shortcomings of the existing RCM models which need further investigation related to sustainability characteristics as a Non-Functional Requirement (NFR), sustainability analysis and dynamic decision making through prioritization. Realizing the gap in the extant literature, there is a need of RCM model for Agile Software Development. This study has overcome the above mentioned shortcomings by proposing Agile Requirement Change Management (ARCM) model. Therefore, this research will provide a solution for requirement change management of quality characteristics during Agile software development. Below is the proposed ARCM model with detailed description.

Proposed Model

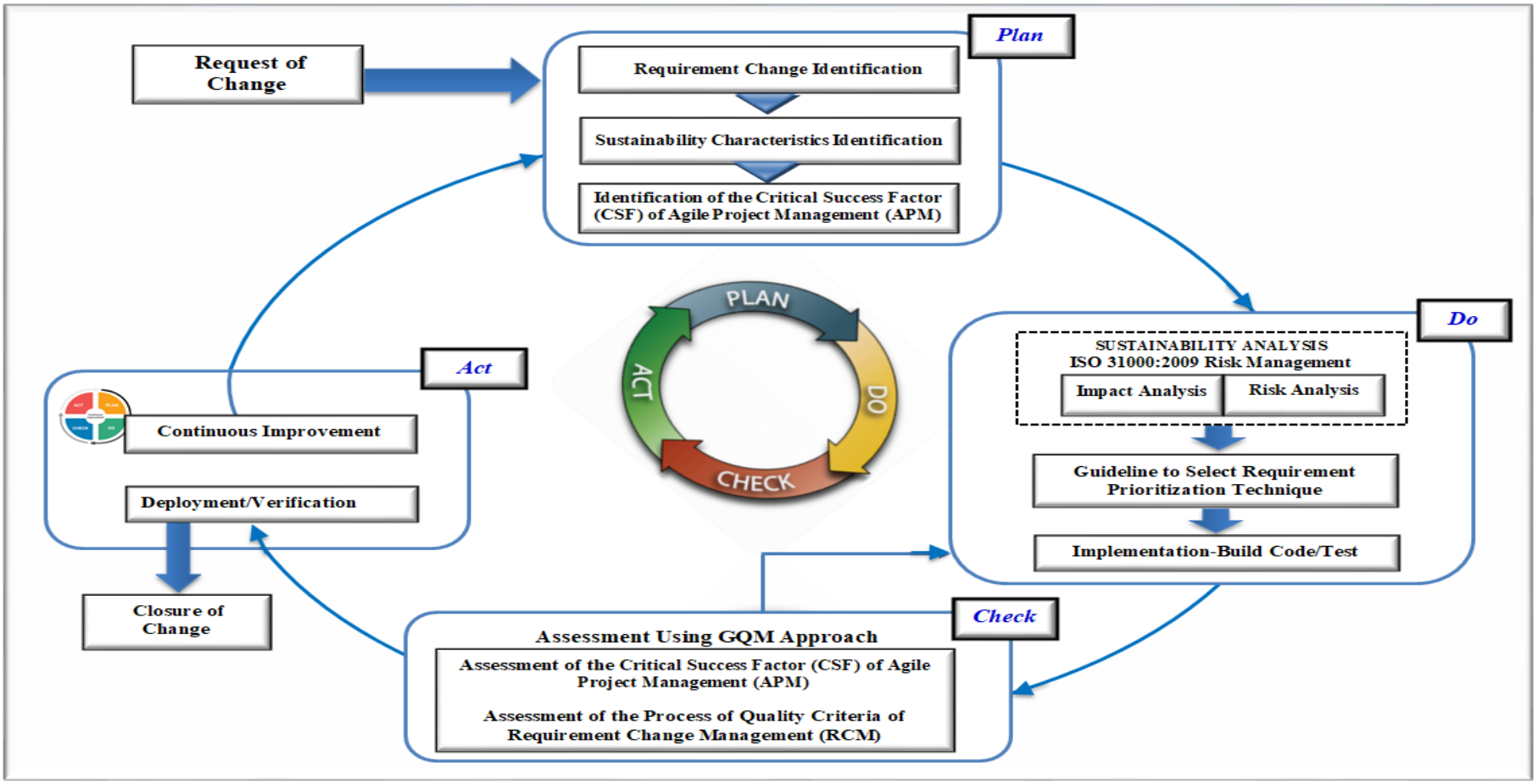


Figure 1: The proposed Agile Requirement Change Management (ARCM) Model

ARCM model is constructed by adapting the Plan-Do-Check-Act (PDCA) (Deming, 2000) method as a base theory besides the outcomes from the theoretical and exploratory studies. PDCA method consists of four main phases, which are plan, do, check, and act. Moreover, this study adapts other studies and theories among them are Goal Oriented Approach, ISO/IEC 25012-2008, and ISO 31000:2009 while constructing the proposed model. The description of the main processes of ARCM according to the PDCA is as follows.

1. **Plan phase**:

This aim of this phase is the requirement change identification. Requirement changes are identified by referring change taxonomy and categorization of change in the context of type, reason, and origin of change. The main activities of this phase are to identify the request of change referring to Change Taxonomy and Change Categorization and identified the request of change by considering sustainability characteristics as a non-functional requirement, in ASD rather than the functional requirement.

- **Requirement Change Categorization/Classification**

The main objective of Requirement change identification is to create requirement change taxonomy and further categorize the requirement change element on the basis of "*reason"* and "*origin"* of change, for a better understanding of change request. Change categorization will be referred to propose a model of requirement change management in ASD.

The proposed categorization is based on two studies of Harker et al. (1993) and Nurmuliani et al. (2004). Requirement elements are categorized according to the reason and origin of requirement change. This research categorized reason of change into three broad category of *mutable, emergent and consequential requirement*, which are further divided into 11 reason of change and these reason of changes are than

mapped with origin of changes found in Nurmuliani et al. (2004). Moreover, the origin of changes are divided into defect reports, engineering's call, project management consideration, marketing group, developer's detailed analysis, design review feedback, technical team discussion, and customer-support discussions. All the taxonomy elements were derived from the change request forms Nurmuliani et al. (2004).

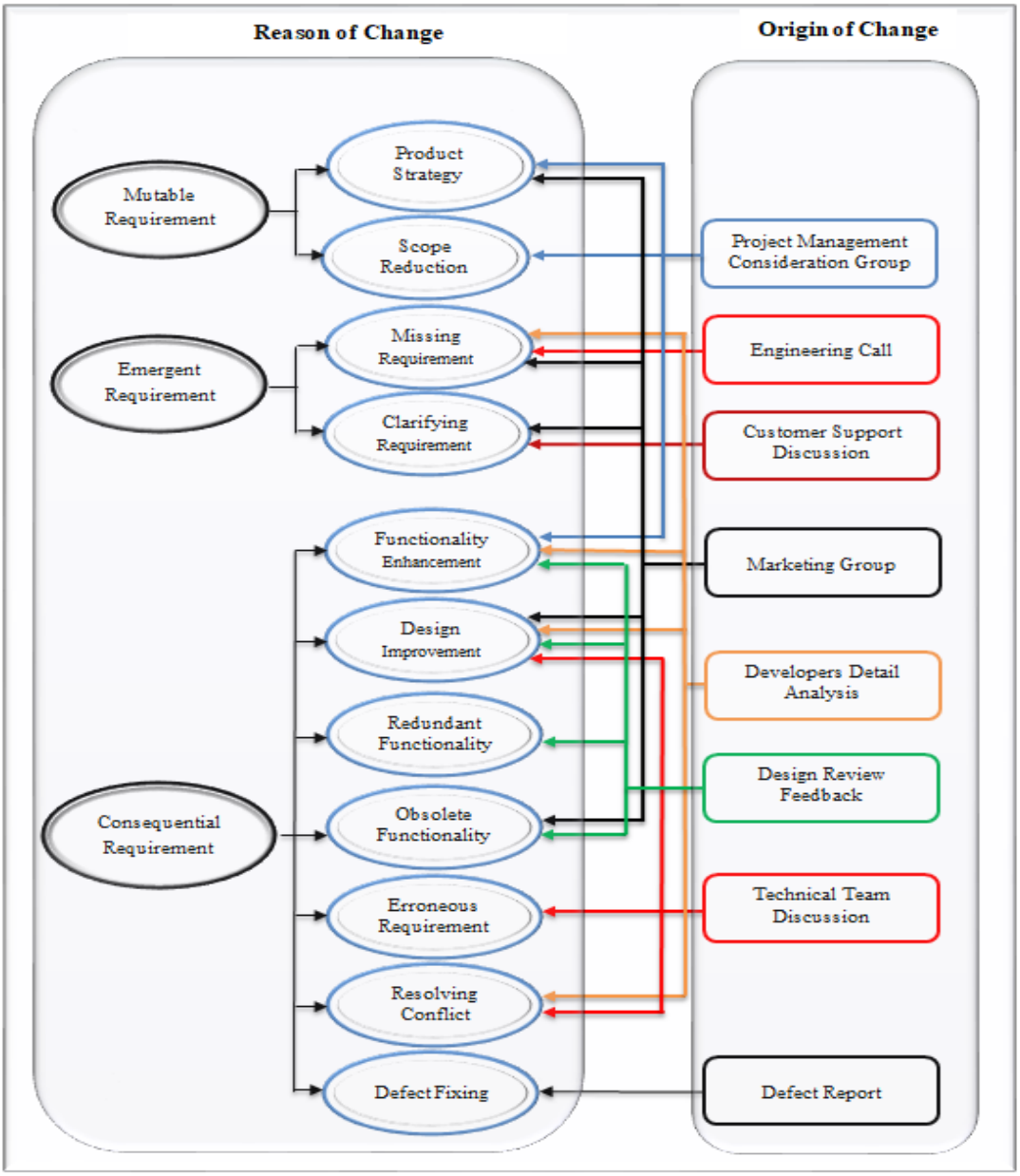


Figure 2: Requirement Change Categorization

- **Sustainability Characteristics and Sub-Characteristics.**

This study identified requirements changes as a sustainability characteristics related to non-functional requirement. A set of important sustainability characteristics from exploratory study as well as taken from the ISO 25010 which is the last standardized model for quality are proposed in this study. The complete 8 sustainability characteristics with their respective 27 sub-characteristics are provided in following Diagram

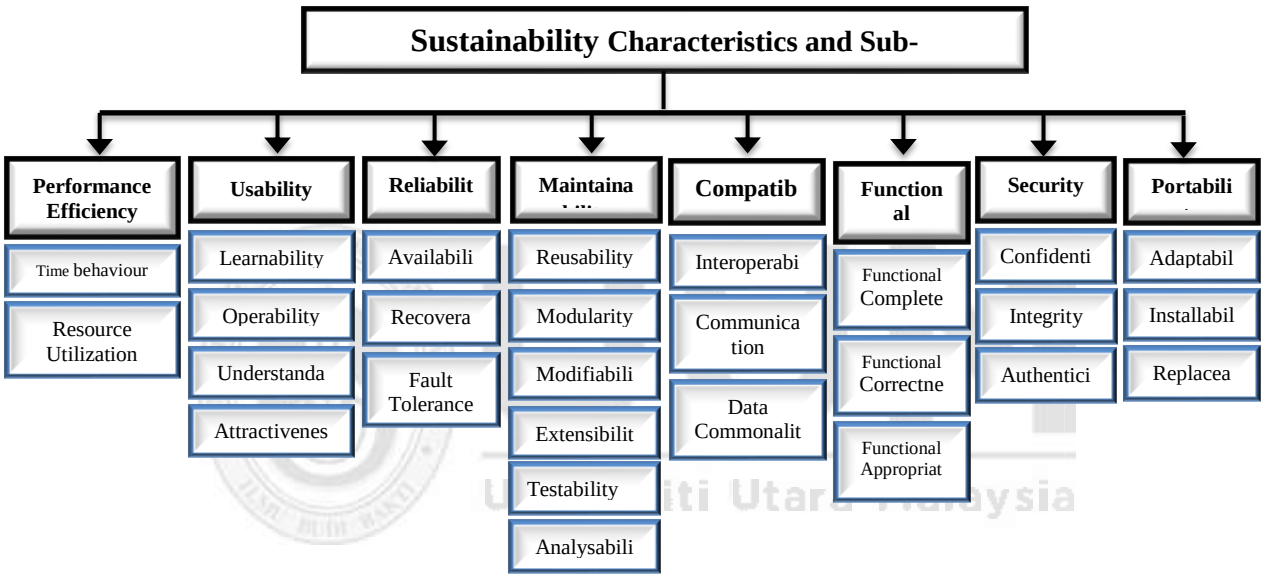


Figure 3: Sustainability Characteristics and Sub-Characteristics

2. Do Phase:

The aim of this phase is to analyse the request of change for the implementation of change. In this study the change analysis, is termed as “Sustainability Analysis”. The sustainability analysis has been performed by assessing and managing risks and its impact to support the decision making related to change. Sustainability analysis method will incorporate the sustainability philosophy during the requirement change process to access the sustainability of the system based on sustainability characteristics with three

dimension of sustainability such as environment, social and economy. Becker et al. (2015) describe the sustainability principles for Software Engineering (SE) as a multi-dimensional structure. The environmental dimension is concerned about the responsible use of natural resources. Further the social dimension refers the societal communities like organizations, groups of people and the factors that enhance trust in society and finally the economic dimension focuses on assets, added value, and capital.

Moreover, this study adapt risk management process of ISO 31000:2009, besides the impact analysis which is a brief and high-level set of principles and guidelines on how to implement risk management. Furthermore, the next process of do phase is the requirement prioritization. Agreed changes are then re-prioritizing by selecting the appropriate requirement prioritization technique. After prioritization, the development and implementation of changes are carried out.

5.3.2.1 Sustainability Analysis

The first step of Do phase is sustainability analysis which involves in understanding its impact and risk factors related to change on the software system so that informed decisions can be made. The sustainability analysis has been comprises of risk analysis and impact analysis to support the decision making related to change. The proposed sustainability analysis method has incorporated the sustainability philosophy during the requirement change process to access the sustainability of the system based on sustainability characteristics and sustainability dimension.

Additionally, the outcomes from the exploratory study also reveal that impact analysis provides an accurate perspective of the implications of a proposed change and handling of requirement change without planning and analysis can influence the quality of software (Refer to Section 4.4.4). Similarly, the survey result highlights that

quality, cost and project duration have the significant impact due to requirement change. Since, the impact of the change is used to predict the cost and effort estimation to manage the budget, schedule and quality of project. Hence, this study refers the change impact with respect to the quality, cost and project duration.

In order to implement risk, the risk management process of ISO 31000:2009, has adapted, which is a brief and high-level set of principles and guidelines on how to implement risk management. The four main steps of ISO 31000:2009 risk management processes adapted in this study are: identify, analyse, evaluate and treat the risk as shown in Figure 4.

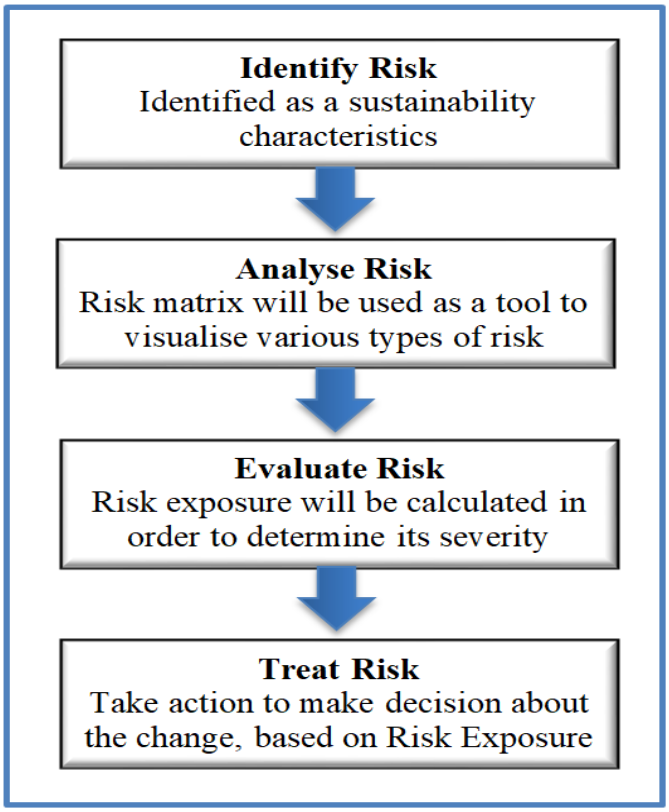


Figure 4. The Steps of Risk Management Process adapted by ISO 31000:2009

In risk management process the steps is: (1) to identify, the risk related to sustainability characteristic are identified by theoretical and exploratory study, (2) for risk analysis and assessment, the risk matrix are used as a tool to visualise various

types of risk, (3) to evaluate, the risk exposure is calculated in order to determine its severity, and (4) to treat the risk and take action is the final activity to make decision about the change and it is based on the value of risk exposure (Refer to, Section 2.3.8.2). For this study a risk management approach incorporates sustainability characteristics with sustainability dimension to identify emerging issues. The steps of risk management process are further discussed in detail.

5. Identify the risk

Table 1 represents the risk factors related to sustainability characteristic and then further mapped with the sustainability dimension in order of importance as extracted from the exploratory study (Refer to Chapter 4, Section 4.7.5.3 and 4.7.5.4).

Table 1

Risk Factors Related to Sustainability Characteristics and Dimension.

Sustainable Characteristics	Sustainable Dimension	Risk Factors
Performance Efficiency	4. Economics 5. Environment 6. Social	• Errors/bugs during development and after release • Create defects that are difficult and expensive to fix • Not accurate to the stakeholders need • Failure to meet performance criteria • Budget overrun • Schedule overrun • Unstable requirements and time-consuming changes • Customer related risk • Business impact risk • Unclear product requirements • Market loses • Technical issues risk
Usability	4. Social 5. Economics 6. Environment	• Create defects that are difficult and expensive to fix • Not accurate to the stakeholders need • Failure to meet performance criteria • Budget overrun • Schedule overrun • Unstable requirements and time-consuming changes • Customer related risk • Business impact risk • Unclear product requirements • Market loses • Technical issues risk • Limit the productivity of internal and external users

Reliability	4. Economics 5. Social 6. Environment	• Errors/bugs during development and after release • Create defects that are difficult and expensive to fix • Not accurate to the stakeholders need • Failure to meet performance criteria • Budget overrun • Schedule overrun • Unstable requirements and time-consuming changes • Customer related risk • Business impact risk • Unclear product requirements • Market loses • Technical issues risk • Limit the productivity of internal and external users
Maintainability	4. Economics 5. Social 6. Environment	• Errors/bugs during development and after release • Create defects that are difficult and expensive to fix • Not accurate to the stakeholders need • Failure to meet performance criteria • Budget overrun • Schedule overrun • Unstable requirements and time-consuming changes • Business impact risk • Unclear product requirements • Market loses • Technical issues risk • Limit the productivity of internal and external users
Compatibility	4. Economics 5. Social 6. Environment	• Errors/bugs during development and after release • Create defects that are difficult and expensive to fix • Not accurate to the stakeholders need • Failure to meet performance criteria • Budget overrun • Unstable requirements and time-consuming changes • Customer related risk • Business impact risk • Unclear product requirements • Market loses • Technical issues risk • Limit the productivity of internal and external users
Functional Suitability	4. Economics 5. Social 6. Environment	• Not accurate to the stakeholders need • Failure to meet performance criteria • Budget overrun • Schedule overrun • Unstable requirements and time-consuming changes • Business impact risk • Unclear product requirements • Market loses • Technical issues risk • Limit the productivity of internal and external users

6. Risk Analysis and Assessment

In the second step of risk analysis and assessment, a risk matrix is constructed by using probability and impact of a change in a form of graph. The input of this step is the value of impact and probability. A scale of five values is used for both impact and probability. Table 2 represents the impact of the risk that is categorized from scale 1 to 5 as 'negligible' to 'severe' respectively with the consequences of impact (PMBOK® Guide, 2013).

Table 2

Impact and Consequence

Score	Impact	Consequences
5	Severe	Catastrophic to the project survival (Greater than 30 % above budgeted).
4	Significance	Serious impact (20 to 29 % above budgeted).
3	Moderate	Cause significance damage (10 to 19 % above budgeted).
2	Minor	Effect in s minor way (5 to 9 % above budgeted).
1	Negligible	Insignificant impact on project (Within 5% of budgeted expenditure).

Moreover, the 'probability', chances of occurrences of risk from 'not likely' to 'expected' are ranked from 1 to 5 with the percentage of occurrence (Abdul Rahman et al., 2017).

Table 3

Probability of Occurrence

Score	Probability	Probability of Occurrence
5	Expected	More than 90% certainly to occur
4	High	64–89% highly likely to occur
3	Moderate	35–63% Possible chance to occur
2	Low	10–34% unlikely to occur
1	Not likely	Less than 10% unlikely to occur

Using this scale the value of impact related to the risk type on the basis of quality, cost and schedule risk has been taken. As, it has been evident from the empirical study that quality, cost and schedule are the main concern of the practitioners (Refer to Chapter 4, Section 4.3.5.1).

Here in this study, the value of impact is determined by the by the association between the two categorical data of impact of change on different type of risk as shown in Table 4.

Table 4

Association between Risk and the impact of change using Chi-Squares Tests

Type of Risks	**Asymp. Sig. (2-sided)**		
	Development Effort (Cost) p-value	**Project Duration p-value**	**Quality of Project p-value**
Business Risk	0.000	0.000	0.000
Technical Risk	0.001	0.000	0.000
Schedule Risk	0.017	0.000	0.003
Cost Risk	0.000	0.002	0.000
Quality Risk	0.000	0.031	0.000
People Risk	0.166	0.000	0.253
Project Risk	0.000	0.000	0.739

		Risk Type after Req change in Agile: Business Risk		
		No	Yes	Total
Impact: Development Effort (Cost)	Slightly importance	3	0	3
	Neutral	5	0	5
	Moderately important	5	12	17
	Very important	18	39	57
	Extremely important	34	21	55
Total		65	72	137

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	21.390[a]	4	.000
Likelihood Ratio	24.727	4	.000
Linear-by-Linear Association	.033	1	.855
N of Valid Cases	137		

a. 4 cells (40.0%) have expected count less than 5. The minimum expected count is 1.42.

Figure 5 The Chi Square Test on the Impact of Cost and Business Risk.

To extract the value of impact these association using Pearson Chi-Square value is categorized with impact to determine the value of impact. As represented in Figure 5 the p value shows the strong association while value is near to zero and represents no relationship whilst the value is near to one. It has been represented in Table 5.8 top row, "business risk" the Pearson Chi-Square value, $p < 0.001$, i.e. a very small probability that represents the strong association, since $p < 0.005$.

Here in this study, the value of impact and probability is used as a guideline by the practitioner. The value of impact is determined by dividing the score of impact with respect to p value ranges from 000 to 0.01 as depicted in Table 5. Moreover, the value of probability can be determined by the experience of practitioner's who is handling the change process as Agile approach is more people-oriented rather than process-oriented and people are an integral part of the Agile Software Development (ASD).

Table 5

The value of impact w.r.t Chi-Squares value

Score	Impact	p-value
5	Severe	000 to 0.002
4	Significance	> 0.002 to 0.004
3	Moderate	> 0.004 to 0.006
2	Minor	> 0.006 to 0.008
1	Negligible	> 0.008 to 0.01

Further, the risk matrix is constructed to visualise various types of risk. Further, the risk matrix is constructed to visualise various types of risk. The x-axis represents the impact with the categories, ranging from 'negligible' to 'severe' while the probability of occurring risk is represented on y-axis ranging from 'not likely' to 'expected'. The decision of the matric of 3 x 3 or 5 x 5 is an arbitrary choice by the creator of the matrix.

PROBABILITY						
	5- Expected	5	10	15	20	25
	4- High	4	8	12	16	20
	3- Moderate	3	6	9	12	15
	2- Low	2	4	6	8	10
	1-Not likely	1	2	3	4	5
		Negligible 1	Minor 2	Moderate 3	Significance 4	Severe 5
		IMPACT				

Figure 6. Risk Matrix with Probability and Impact

Legends	1-4 Verv Low	5-9 Low Risk	10-14 Moderate	15-19 High Risk	20-25 Extreme

7. Evaluate the risk

The last step is to determine the risk exposure for each risk factor related to sustainability dimension. The risk exposure can be calculated by multiplying risk probability and impact as shown in equation 5.1 below. Finally, the calculated risk exposure will be mapped to risk exposure scales as shown in Table 6 to determine the acceptable level of risk.

$$RE = P * I \quad (5.1)$$

Where:

RE = Exposure of risk

P = Probability of risk

I = Impact of risk

Finally, the ARCM Model evaluates the risk by calculating the value of risk exposure by adding the value of impact and probability in the risk matrix as depicted in Figure 7.

To calculate the risk exposure, the value of impact and probability to occur the risk related to the change is determined. It starts when the new Request for Change (RFC) related to the sustainability characteristic is arrived. Thus, the factor and its impact on cost, schedule and quality is determined as shown in Table 6. Afterward its value of impact has been determined by considering the above mentioned Table 4 and Table 5. Then the probability to occur this risk factor is determined through subjective approach. The software practitioners derive the value of probability by keeping the project nature in mind. Subsequently, the risk exposure is calculated by multiplying the impact with probability using the formula as shown in 5.1.

To visualize the risk exposure the risk matrix is used as shown in Figure 5 By considering the example given in Table 6 the outcome of the risk matrix is depicted in Figure 7.

PROBABILITY						
	5- Expected					
	4- High					RFC-3
	3- Moderate			RFC-1		
	2- Low					RFC-2
	1-Not likely					
		Negligible 1	Minor 2	Moderate 3	Significance 4	Severe 5
		IMPACT				

Figure 7. Example of Risk Matrix with Probability and Impact

8. Treat the risk

On the basis of the risk exposure as calculated in Table 6, the decision related to accept the change, reject the change or defer it to the next iteration is done on the basis of its impact on the sustainability dimension of economics, social and environment as mentioned in Table 6.

Table 6

Risk Exposure Scale adapted by Abdul Rahman et al. (2017)

Risk Exposure Level	Score	Risk Exposure Description
Extreme	20-25	**Economic**: Budget overrun. Change suspended. **Social**: Unsatisfied customer/stakeholder. Unable to satisfy customer/ stakeholder. **Environment**: Unable to comply with long term usage. **Overall**: Project failure if risk occurs in execution.
High	15-19	**Economic**: Budget overrun. Reschedule to later Iteration. **Social**: Incapable to satisfy customer/stakeholder. **Environment**: Able to comply with long term usage but incur additional charges. **Overall**: Significantly degrade project capabilities in term of required project standards. Not complete project on time, increases cost and degrade quality.
Medium	10-14	**Economic**: Limited budget estimated. Change is possible. **Social**: Capable to satisfy customer/stakeholder and mange changes. **Environment**: Able to comply with long term usage with limited additional cost. **Overall**: Degrade project capabilities in term of required output if risk occurs during the project.
Low	5-9	**Economic**: Reasonable budget estimated. Adequate resource to manage changes in the same iteration. **Social**: Capable to satisfy customer/stakeholder and mange changes. **Environment**: Able to comply with long term usage without additional cost. **Overall**: Expected loses have very little impact on project success.
Very Low	0-4	**Economic**: Moderate budget estimated. Adequate resource to manage changes in the same iteration. **Social**: Capable to satisfy customer/stakeholder and provides adequate services and manage all changes. **Environment**: Capable to comply with long term usage without cost. **Overall**: Expected loses have no impact on project success.

The exposure level of risk is ‘extreme’ ranges between 20-25, the change should be suspended as the value of risk exposure shows it is unlikely to change. The required change will affect the sustainability dimension of economic, social and environment and leads towards project failure in case of risk occur during execution. Next, in case the level of risk is high ranges between 15-19, the change should be rescheduled to later iteration. The required request of change can significantly degrade project capabilities in term of required project standards. Project will not complete on time, increases cost and degrade the quality. Moreover, in case, the level of risk is medium and low ranges between 10-14, and 5-9 respectively, the change is possible in current iteration. The required request of change can be possible due to the reasonable budget estimated and adequate resource to manage changes in the same iteration.

Provide a Guideline to Select Prioritization Technique

The second step of Do phase is to provide guidelines to prioritize requirements with considering all the relevant factors that have an effect on priorities with different stakeholder views together (Lehtola et al., 2004). By the insight of the prior literature of requirement prioritization techniques and comparison on different factor of prioritization, this research proposes a framework which will focuses on providing the guideline to select suitable RP technique, with other factors such as stakeholders, project constraints and requirement nature. After the phase of identification, analysis and prioritization, the next stage is the verification and assessment of the changes.

Table 7

Guideline to select prioritization technique after requirement change

RP Techniques	S Strength	W Weakness	O Opportunities	T Threats
Cost-value ranking	• Capability to combine the judgments of both cost and value of requirements for implementation. • Cost Value ranking determines top requirements by graph plots to visualize the requirements value against its implementation cost.	• Time consuming and un-scalable • Requirements computational complexity increases in managing interdependencies as the number of requirements increases	• Cost Value is the best technique while taking both dependency and benefit as a vital factor. • Need to increase the business value and market share.	• Reliability of the result. • Complexity
Value-oriented prioritization (VOP)	• Organization business value is taken into consideration in the prioritization process.	• Ignores requirement dependencies. • Not suitable for larger project.	• Stakeholder ratings by linking them to identified business values. • Customer preferences.	• Complexity • Reliability of the result. • Consistency • Technical risk
Cumulative Voting	• Simplicity of the approach.	• Not suitable for large number of requirements. • Does not permit evaluation of the relative priority difference among the requirements.	• Ease of use • Customers and stakeholders preferences. • Business Value. • Cost/Benefit/Penalty.	• Time consumed. • Consistency in result. • Reliability of result.
MoSCoW	• It is consistent, less difficult, less effort required and able to handle large number of alternative. • Easily scalable, as it is suitable for both small and large numbers of requirements.	• The problem comes with its lack of grading within categories. It is difficult to know which *SHOULD* or *COULD* requirements are more important than others. Better suited to product with less customers.	• Ease of use • Customers and stakeholders preferences.	• Time consumed. • Consistency in result. • Reliability of result.
Planning Game	• Planning game has a better modification of numerical computation. • Easy and fast to complete the prioritization process.	• Problematic with large number of requirements.	• Ease of use • Business Value. • Technical Risk	• Time consumed. • Reliability of result.
Analytic hierarchy process (AHP)	• Ability to resolve conflicting objectives. • Provide reliable result.	• Time consuming at higher number of requirements. • Not scalable so problematic for larger project.	• Reliability of the result. • Consistency.	• Complex in execution. • Time consumed.

Quality Functional Deployment (QFD)	• QFD is a structured methodology for customer needs in the form of "voice of the customer".	• Mainly applied in small systems. • Limitation in inconsistencies and scalability.	• Customers and stakeholders preferences. • Business Value. • Consistency.	• Complex in execution. • Reliability of result. • Time consumed.
Binary Search Tree (BST)	• BST could easily scale up to thousands of requirements, and still be a very fast candidate.	• BST does not assign any priority values rather only a simple ranking of requirements. • BST shows which requirement is more favourable but the extent to which the requirement is important cannot be known and therefore the comparison is just ordinal.	• Consistency.	• Complex in execution. • Reliability of result. • Time consumed.
Wiegers' matrix approach	• Matrix prioritization is easily scalable and based on several criteria (benefit, penalty, cost, and risk.) • Spreadsheet auto-calculates the priority values and very easy to conduct.	• It can be easily manipulated by stakeholders to accomplish their objectives.	• Customers and stakeholders preferences. • Technical Risk • Cost/Benefit/Penalty.	• Stakeholder biasness. • Complex in execution. • Time consumed.
Kano Model	• Kano is more concerned to the customer preferences for customer "Trustworthiness". • Kano method is the fastest way to prioritize requirements.	• It can only be used for analysing the effects. • It is not for suggesting new product features, something that is quite difficult to achieve.	• Ease of use • Customers and stakeholders preferences.	• Reliability of the result. • Consistency • Technical risk
Pair wise analysis	• Criteria for comparing options can remain informal, thereby basing judgments on participants' experiences.	• Tedious, complicated and provide unreliable results. • Ignores level of detail or sophistication of a multi-criteria analysis. • Limitation in scalability.	• Customers and stakeholders preferences.	• Consistency • Reliability of the result. • Time consumed

3. Check Phase:

The aim of the check phase is the assessment of the ARCM model. The model is assessed using GQM method (Basili et al., 1994; Solingen & Berghout, 1999). This phase assess the process of quality criteria of RCM and the Critical Success Factor (CSF) of Agility by focusing on three main factors i.e. completeness, consistency and accuracy (Baharom, Deraman, Hamdan, & Shafinah, 2012; Heck & Zaidman, 2017). In case the assessment is not satisfied than it will be again referred to the 'Do phase' for analysis.

In ARCM model GQM emphasizes on five main factors of CSF such as 'people', 'process', 'project', 'organization' and 'technical'. *As the focus of the requirement change management is in Agile, therefore the business goal in this exercise is to monitor the 'effectiveness' and 'Agile Project Management (APM)' of CSF to achieve the Agile characteristic, hence the process of change management remain Agile.* Critical success factor CSF is measured in terms of Agile Project Management Practices (APM) of involvement of good quality '*people'*, nature of '*project'*, supportability of working environment in '*organization'*, and the use of appropriate '*technology'* as mentioned in Table 5.4. Moreover, the effectiveness is measured in terms of the level of completeness, consistency and accuracy of the '*processes'* in managing requirement change management. The following is the definition of these three quality factors (Baharom, Deraman, Hamdan, & Shafinah, 2012; Heck & Zaidman, 2017).

- ***Completeness:*** The availability of all relevant data to satisfy the user requirement and all required elements of processes area should be present and as much formalized as possible.

- ***Consistency:*** All elements should be consistent with each other and with the other process areas and there is an absence of difference, when comparing two or more representations of a thing against a definition.
- ***Accuracy:*** Each element should be correct and describes the "real world" object or event being described.

4. Act Phase:

The aim of the Act phase is the deployment, verification and continuous improvement for the next iteration of change management process, and finally the closure of the request of change. For further detail of each phases, process, activities and practices are provided in the following Table.

The Flow of ARCM Model

The detailed flow of proposed ARCM Model is illustrated in following Figure 8.

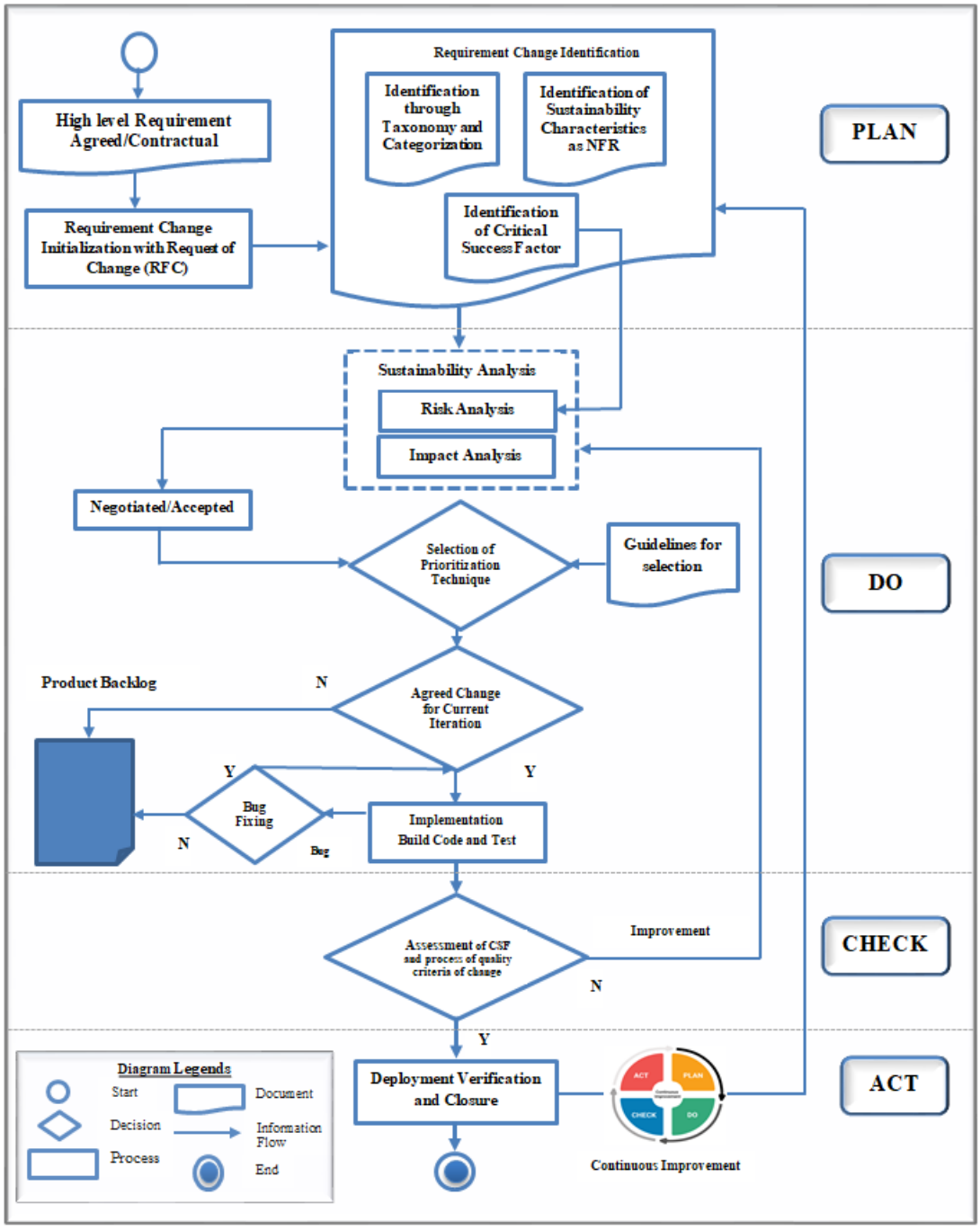


Figure 8: Flow of the ARCM Model

As shown in Figure, the first stage is the change initiation in the form of change request. The requirements change request needs identification in term of classification/categorization. For this study the type of change request consider the sustainability characteristics as a non-functional requirement. Furthermore, the next step is the “sustainability analysis” of the requirement change in the form of risk and impact analysis of change. Afterward, the next step is the dynamic decision making using requirement prioritization. Guideline for the selection of prioritization technique has been provided. Subsequently after the prioritization the next step is to implement the change which consists of the activities to build code and test. In case of failure of verification process, the request is sent to the product backlog to consider it in the next iteration.

Afterward, the quality of RCM process has been assessed using the Critical Success Factor (CSF) of Agile Project Management (APM). In case assessment does not follow Agile characteristics, it goes back to sustainability analysis for the continuous improvement, otherwise it continues towards deployment/verification and closure of change.

Appendix J

Validation Template for Case Study

PLAN PHASE- CHANGE IDENTIFICATION PROCESS

Change Identification		
RFC No:		
RFC Description:		
Reason of Change	**Selection**	**Comment**
Product Strategy	☐	
Scope Reduction	☐	
Missing Requirements	☐	
Clarifying Requirements	☐	
Functionality Enhancement	☐	
Design Improvement	☐	
Redundant Functionality	☐	
Obsolete Functionality	☐	
Erroneous Requirements	☐	
Resolving Conflicts	☐	
Defect Fixing	☐	
Other	☐	
Origin of change	**Selection**	**Comment**
Marketing Group.	☐	
Project Management Consideration.	☐	
Developer's Detailed Analysis.	☐	
Engineering's Call.	☐	
Customer-Support discussions	☐	
Design Review Feedback	☐	
Technical Team Discussion.	☐	
Defect Report	☐	
Other	☐	

Additional Comments:

Sustainability Characteristics Identification	Selection	Comment
Performance Efficiency: The characteristic represents the performance relative to the amount of resources used under stated conditions.		
Time behaviour (The capacity of a product while performing its functions to meet requirements regarding response time, throughput and processing time)	□	
Resource utilization (When performing its functions, the amounts and types of resources used by a product or system to meet requirements.	□	
Other Important sub-characteristics related to **Performance Efficiency** (please identify, if any):	□	
Compatibility: A product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.		
Interoperability (degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.)	□	
Communication commonality (The degree to which software is dependent on its associated hardware.)	□	
Data commonality (The use of standard data representation.)	□	
Other Important sub-characteristics related to **Compatibility** (please identify, if any):	□	
Usability: Feature that enable the system to be user friendly and to achieve identified goals with efficiency, satisfaction and effectiveness.		
Learnability (degree to which a product or system enables the user to learn how to use it with effectiveness, efficiency in emergency situations.)	□	
Understandability (The ability to recognize whether the software is appropriate for the user needs.)	□	
Operability (The capacity of a product that make it easy to operate and control.)	□	
Attractiveness (User interface assists satisfying and pleasing interaction for the user.)	□	
Other sub-characteristics related to **Usability** (please identify, if any):	□	
Reliability: The capacity of a product to perform specified functions for a specified period of time, under specified conditions.		
Availability (The capacity of a product or component to be accessible and operational at the time of use.)	□	
Fault tolerance (The capacity of a product to be operated as required in spite of the existence of hardware or software faults.)	□	
Recoverability (The capacity of a product to recover the data and restore the state of the system during the event of an interruption or a failure.)	□	
Other sub-characteristics related to **Reliability** (please identify, if any):	□	
Security: Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.		
Confidentiality (degree to which the prototype ensures that data are accessible only to those authorized to have access.)	□	
Integrity (degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data.)	□	
Authenticity (degree to which the identity of a subject or resource can be proved to be the one claimed.)	□	
Other sub-characteristics related to **Security** (please identify, if any):	□	
Maintainability: The characteristic represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in environment, and in requirements.)		

Modularity (A discrete component that does not have an impact on other components during change.)	□	
Analysability (A discrete component that does not have an impact on other components during change. Diagnose a product for deficiencies or causes of failures due to change.)	□	
Modifiability (Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.)	□	
Extensibility (The level of effort required to implement the extension.)	□	
Reusability (The capacity of a module or component that can be re-used in other software system.)	□	
Testability (The capacity to construct the test criteria with effectiveness and perform tests to meet those criteria.)	□	
Other sub-characteristics related to **Maintainability** (please identify, if any):	□	
Portability: Capability of the system to run under different computing environments		
Adaptability (The capacity of product or system to be adopted in different hardware, software or other operational environments.)	□	
Installability (The capacity of a product to be successfully installed or uninstalled in any environment.)	□	
Replaceability (The capacity of replacement of a product by another software product in the same environment.)	□	
Other sub-characteristics related to **Portability** (please identify, if any):	□	
Functional Suitability: The degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.		
Functional Completeness (degree to which the set of functions covers all the specified tasks and user objectives.)	□	
Functional Correctness (degree to which the functions provides the correct results with the needed degree of precision.)	□	
Functional Appropriateness (degree to which the functions facilitate the accomplishment of specified tasks and objectives.)	□	
Other sub-characteristics related to **Functional Suitability** (please identify, if any):	□	

Association between Risk and the impact of change using Chi-Squares Tests

Type of Risks	Asymp. Sig. (2-sided)		
	Development Effort (Cost) p-value	Project Duration p-value	Quality of Project p-value
Business Risk	0.000	0.000	0.000
Technical Risk	0.001	0.000	0.000
Schedule Risk	0.017	0.000	0.003
Cost Risk	0.000	0.002	0.000
Quality Risk	0.000	0.031	0.000
People Risk	0.166	0.000	0.253
Project Risk	0.000	0.000	0.739

The Value of Impact w.r.t Chi-Squares Value

Score	Impact	p-value
5	Severe	000 to 0.002
4	Significance	> 0.002 to 0.004
3	Moderate	> 0.004 to 0.006
2	Minor	> 0.006 to 0.008
1	Negligible	> 0.008 to 0.01

RFC-1:							
Risk Type	Sustainable Characteristics **Dimension**	**Impact Type(s)**	**Impact Value**	**Probability**	**Risk Exposure**	**Risk Exposure Description**	**Impact Description**
Business Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
Technical Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
Schedule Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
Cost Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
Quality Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
People Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
Project Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
		Total Probability		100 %			

Legends							
Score	**Impact**	**Score**	**Probability**	**Type**	**Description**	**Type**	**Description**
5	Severe	5	Expected	BS	Business Risk	PE	People Risk
4	Significance	4	High	TE	Technical Risk	PE	Project Risk
3	Moderate	3	Moderate	SC	Schedule Risk		
2	Minor	2	Low	CO	Cost Risk		
1	Negligible	1	Not likely	QU	Quality Risk		

Risk Exposure Level	Score	Risk Exposure Description
Extreme	20-25	**Economic**: Budget overrun. Change suspended. **Social**: Unsatisfied customer/stakeholder. Unable to satisfy customer/ stakeholder. **Environment**: Unable to comply with long term usage. **Overall**: Project failure if risk occurs in execution.
High	15-19	**Economic**: Budget overrun. Reschedule to later Iteration. **Social**: Incapable to satisfy customer/stakeholder. **Environment**: Able to comply with long term usage but incur additional charges. **Overall**: Significantly degrade project capabilities in term of required project standards. Not complete project on time, increases cost and degrade quality.
Medium	10-14	**Economic**: Limited budget estimated. Change is possible. **Social**: Capable to satisfy customer/stakeholder and mange changes. **Environment**: Able to comply with long term usage with limited additional cost. **Overall**: Degrade project capabilities in term of required output if risk occurs during the project.
Low	5-9	**Economic**: Reasonable budget estimated. Adequate resource to manage changes in the same iteration. **Social**: Capable to satisfy customer/stakeholder and mange changes. **Environment**: Able to comply with long term usage without additional cost. **Overall**: Expected loses have very little impact on project success.
Very Low	0-4	**Economic**: Moderate budget estimated. Adequate resource to manage changes in the same iteration. **Social**: Capable to satisfy customer/stakeholder and provides adequate services and manage all changes. **Environment**: Capable to comply with long term usage without cost. **Overall**: Expected loses have no impact on project success.

RP Technique selection after requirement change

RP Techniques	Choice of RP Technique	Reason/Comment
Cost-value ranking		
Value-oriented prioritization (VOP)		
Cumulative Voting		
MoSCoW		
Planning Game		
Analytic hierarchy process (AHP)		
Quality Functional Deployment (QFD)		
Binary Search Tree (BST)		
Wiegers' matrix approach		
Kano Model		
Pair wise analysis		

Assessment of the Agile RCM Process			
Metric Factor: Process- Completeness Metrics			
Metrics(M)	**Yes**	**No**	**Suggestion/Change (if needed)**
M1: The request of change is identified by considering the source and reason of changes	☐	☐	
M2: Sustainability analysis was performed using Risk matrix	☐	☐	
M3: After change prioritization was performed using the guideline provided.	☐	☐	
M4: Change are implemented successfully	☐	☐	
Metric Factor: Process- Consistency Metrics			
Metrics(M)	**Yes**	**No**	**Suggestion/Change (if needed)**
M5: Changes are identified by considering change categorization appropriately.	☐	☐	
M6: Requirement change analysis was performed by following the standards of requirement change analysis.	☐	☐	
M7: Requirement prioritization was done after every change.	☐	☐	
M8: Appropriate procedure was followed to implement the requirement change.	☐	☐	
Metric Factor: Process- Accuracy Metrics			
Metrics(M)	**Yes**	**No**	**Suggestion/Change (if needed)**
M9: The requirement change identified by categorization/ classification on the basis of reason and origin of change has arranged correctly.	☐	☐	
M10: Appropriate procedure was used to analyse the requirement change.	☐	☐	
M11: Appropriate procedure was followed to select prioritization technique after requirement change.	☐	☐	
M12: Changes were implemented by following all the standard activities of requirement change.	☐	☐	

Ranking: The score ranged from 0 to 4 Likert scale

where: 0 = Never, 1 = Rarely, 2 = Sometimes, 3 = Often and 4 = Always.

Assessment of the Critical Success Factor-People, Project, Organization, Technology		
Metrics	**Score**	**Comment**
Metric Factor: People		
M13: Team capability Team members are capable to follow the process of Agile.	**0 1 2 3 4**	
M14: Motivated team members: The current progress of iteration was revealed to everyone in the team during iteration.	**0 1 2 3 4**	
M15: Agile process knowledge. All the Team members have the basic Agile process knowledge	**0 1 2 3 4**	
M16: Self-organizing teamwork. Team members have the autonomous and make any decision related to change.	**0 1 2 3 4**	
M17: Good customer relationship: Customer and end-user involvement were monitored in change activity	**0 1 2 3 4**	
M18: Customer involvement: Customers or Project Owner (PO) was available on-site for face-to-face discussions during the requirement change	**0 1 2 3 4**	
Metric Factor: Project		
M19: The nature of project was Non-life-critical.	**0 1 2 3 4**	
M20: Within the project scope varies with emergent requirement.	**0 1 2 3 4**	
M21: Project was implemented with small team	**0 1 2 3 4**	
Metric Factor: Organizational		
M22: Organizational culture and mind set support frequent requirement change in Agile.	**0 1 2 3 4**	
M23: Management of company support the process of change.	**0 1 2 3 4**	
Metric Factor: Technology		
M22: Organizational culture and mind set support frequent requirement change in Agile.	**0 1 2 3 4**	
M23: Management of company support the process of change.	**0 1 2 3 4**	

ACT PHASE- DEPLOYMENT AND VERIFICATION

Following checklist will be used during Act phase:

Event	Agreed/Accepted
Change is agreed as mentioned in the RFC	
Timeline for deployment is acceptable to all stakeholders	
Authorization for deployment	

Appendix K

The Findings of the First Case Study (Company-A)

Target	To determine the acceptability of the ARCM Model in the software company that had experience and mature with the Agile methodologies.		
Project	The case study was performed on the projects produced by organization-A by assessing the Business Intelligence system for Automated Dashboard and Analytics Reporting of Shell company.		
Evaluation Team	**Roles**	**Skills**	**Experiences**
	Project Manager	Manager	19
	Team Lead	Manager	14
	Domain Expert	Agile Software Development	16
	Technical Lead	Front End Development	12

➢ PLAN PHASE- CHANGE IDENTIFICATION PROCESS

Change Identification		
RFC No: 01		
RFC Description: Data Corruption should be prevented by applying the possible backup procedures		
Reason of Change	**Selection**	**Comment**
Product Strategy	☐	
Scope Reduction	☐	
Missing Requirements	☐	
Clarifying Requirements	☐	
Functionality Enhancement	☐	
Design Improvement	☐	
Redundant Functionality	☐	
Obsolete Functionality	☐	
Erroneous Requirements	☐	
Resolving Conflicts	☐	
Defect Fixing	√	
Other	☐	

Origin of change	Selection	Comment
Marketing Group.	☐	
Project Management Consideration.	☐	
Developer's Detailed Analysis.	☐	
Engineering's Call.	☐	
Customer-Support discussions	√	
Design Review Feedback	☐	
Technical Team Discussion.	☐	
Defect Report	☐	
Other	☐	

Additional Comments:

Sustainability Characteristics Identification	Selection	Comment
Performance Efficiency: The characteristic represents the performance relative to the amount of resources used under stated conditions.		
Time behaviour (The capacity of a product while performing its functions to meet requirements regarding response time, throughput and processing time)	☐	
Resource utilization (When performing its functions, the amounts and types of resources used by a product or system to meet requirements.	☐	
Other Important sub-characteristics related to **Performance Efficiency** (please identify, if any):	☐	
Compatibility: A product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.		
Interoperability (degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.)	☐	
Communication commonality (The degree to which software is dependent on its associated hardware.)	☐	
Data commonality (The use of standard data representation.)	☐	
Other Important sub-characteristics related to **Compatibility** (please identify, if any):	☐	
Usability: Feature that enable the system to be user friendly and to achieve identified goals with efficiency, satisfaction and effectiveness.		
Learnability (degree to which a produc-t or system enables the user to learn how to use it with effectiveness, efficiency in	☐	

emergency situations.)		
Understandability (The ability to recognize whether the software is appropriate for the user needs.)	☐	
Operability (The capacity of a product that make it easy to operate and control.)	☐	
Attractiveness (User interface assists satisfying and pleasing interaction for the user.)	☐	
Other sub-characteristics related to **Usability** (please identify, if any):	☐	
Reliability: The capacity of a product to perform specified functions for a specified period of time, under specified conditions.		
Availability (The capacity of a product or component to be accessible and operational at the time of use.)	☐	
Fault tolerance (The capacity of a product to be operated as required in spite of the existence of hardware or software faults.)	☐	
Recoverability (The capacity of a product to recover the data and restore the state of the system during the event of an interruption or a failure.)	√	
Other sub-characteristics related to **Reliability** (please identify, if any):	☐	
Security: Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.		
Confidentiality (degree to which the prototype ensures that data are accessible only to those authorized to have access.)	☐	
Integrity (degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data.)	☐	
Authenticity (degree to which the identity of a subject or resource can be proved to be the one claimed.)	☐	
Other sub-characteristics related to **Security** (please identify, if any):	☐	
Maintainability: The characteristic represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in environment, and in requirements.)		
Modularity (A discrete component that does not have an impact on other components during change.)	☐	
Analysability (A discrete component that does not have an impact on other components during change. Diagnose a product for deficiencies or causes of failures due to change.)	☐	
Modifiability (Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.)	☐	
Extensibility (The level of effort required to implement the extension.)	☐	
Reusability (The capacity of a module or component that can be re-used in other software system.)	☐	
Testability (The capacity to construct the test criteria with effectiveness and perform tests to meet those criteria.)	☐	
Other sub-characteristics related to **Maintainability** (please identify, if any):	☐	
Portability: Capability of the system to run under different computing environments		
Adaptability (The capacity of product or system to be adopted in	☐	

different hardware, software or other operational environments.)		
Installability (The capacity of a product to be successfully installed or uninstalled in any environment.)	☐	
Replaceability (The capacity of replacement of a product by another software product in the same environment.)	☐	
Other sub-characteristics related to **Portability** (please identify, if any):	☐	
Functional Suitability: The degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.		
Functional Completeness (degree to which the set of functions covers all the specified tasks and user objectives.)	☐	
Functional Correctness (degree to which the functions provides the correct results with the needed degree of precision.)	☐	
Functional Appropriateness (degree to which the functions facilitate the accomplishment of specified tasks and objectives.)	☐	
Other sub-characteristics related to **Functional Suitability** (please identify, if any):	☐	

➢ DO PHASE- SUSTAINABILITY ANALYSIS

Risks Associated with Cost "CO" (Risks associated with cost of developing the project/product)

Sr. #	RISK TYPE	**Impact On**
1	Business Risk	**Development Effort (Cost)** √ Project Duration ☐ Quality of Project ☐
2	Technical Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐
3	Schedule Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐
4	Cost Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐
5	Quality Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐
6	People Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐
7	Project Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐

Association between Risk and the impact of change using Chi-Squares Tests

Type of Risks	Asymp. Sig. (2-sided)		
	Development Effort (Cost) p-value	Project Duration p-value	Quality of Project p-value
Business Risk	0.000	0.000	0.000
Technical Risk	0.001	0.000	0.000
Schedule Risk	0.017	0.000	0.003
Cost Risk	0.000	0.002	0.000
Quality Risk	0.000	0.031	0.000
People Risk	0.166	0.000	0.253
Project Risk	0.000	0.000	0.739

The Value of Impact w.r.t Chi-Squares Value

Score	Impact	p-value
5	Severe	000 to 0.002
4	Significance	> 0.002 to 0.004
3	Moderate	> 0.004 to 0.006
2	Minor	> 0.006 to 0.008
1	Negligible	> 0.008 to 0.01

Risks Associated with Cost "CO" (Risks associated with cost of developing the project/product)

Sr. #	DESCRIPTION	Applicable	Impact
1	Project will exceed initial cost estimates for the development or maintenance of the project	Yes	5
2	A revenue risk, or commercial risk (in revenue-based contracts), depending on the customer and criticality of application	N/A	
3	Costs associated with late delivery	N/A	
4	Costs associated with a defective product	N/A	

RFC-1: **System should be able to cater 10,000 users simultaneously.**							
Risk Type	Sustainable Characteristics **Dimension**	**Impact Type(s)**	**Impact Value**	**Probability**	**Risk Exposure**	**Risk Exposure Description** *Refer to Table 6* Risk Exposure *(*Appendix I - Guideline Document for ARCM Model*)*	**Impact Description By Practitioner**
Business Risk	Economics □ Environment □ Social □	**Development Effort (Cost)** √ Project duration (Schedule) □ Project Quality □	5	2	10	**Economic**: Limited budget estimated. Change is possible. **Social**: Capable to satisfy customer/stakeholder and mange changes. **Environment**: Able to comply with long term usage with limited additional cost. **Overall**: Degrade project capabilities in term of required output if risk occurs during the project.	Project will exceed initial cost estimates for the development or maintenance of the project
Technical Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
Schedule Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
Cost Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					

RFC-1: **System should be able to cater 10,000 users simultaneously.**							
Risk Type	Sustainable Characteristics **Dimension**	**Impact Type(s)**	**Impact Value**	**Probability**	**Risk Exposure**	**Risk Exposure Description** *Refer to Table 6* Risk Exposure *(*Appendix I - Guideline Document for ARCM Model*)*	**Impact Description By Practitioner**
Quality Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
People Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
Project Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
		Total Probability	40 %				

Legends							
Score	**Impact**	**Score**	**Probability**	**Type**	**Description**	**Type**	**Description**
5	Severe	5	Expected	BS	Business Risk	PE	People Risk
4	Significance	4	High	TE	Technical Risk	PE	Project Risk
3	Moderate	3	Moderate	SC	Schedule Risk		
2	Minor	2	Low	CO	Cost Risk		
1	Negligible	1	Not likely	QU	Quality Risk		

Selection of Requirement Prioritization Technique

*Refer to Table 7 Guideline to select prioritization technique after requirement change (*Appendix I Guideline Document for ARCM Model*)*

RP Techniques	**Choice of RP Technique**	**Reason/Comment**
Cost-value ranking		
Value-oriented prioritization (VOP)	√	Organization business value is the main concern while choosing the prioritization technique.
Cumulative Voting		
MoSCoW		
Planning Game		
Analytic hierarchy process (AHP)		
Quality Functional Deployment (QFD)		
Binary Search Tree (BST)		
Wiegers' matrix approach		
Kano Model		
Pair wise analysis		

➢ CHECK PHASE- ASSESSMENT OF THE ARCM MODEL

Assessment of the Agile RCM Process			
Metric Factor: Process- Completeness Metrics			
Metrics(M)	**Yes**	**No**	**Suggestion/Change (if needed)**
M1: The request of change is identified by considering the source and reason of changes	√	□	The participants indicated that all the processes and activities perform to manage change in the proposed model are sufficient for managing RCM in ASD.
M2: Sustainability analysis was performed using Risk matrix	√	□	
M3: After change prioritization was performed using the guideline provided.	√	□	
M4: Change are implemented successfully	√	□	
Metric Factor: Process- Consistency Metrics			
Metrics(M)	**Yes**	**No**	**Suggestion/Change (if needed)**
M5: Changes are identified by considering change categorization appropriately.	√	□	For the proposed ARCM model all the RCM processes are appropriate and following the standards of requirement change analysis and change implementation
M6: Requirement change analysis was performed by following the standards of requirement change analysis.	√	□	
M7: Requirement prioritization was done after every change.	√	□	
M8: Appropriate procedure was followed to implement the requirement change.	√	□	
Metric Factor: Process- Accuracy Metrics			
Metrics(M)	**Yes**	**No**	**Suggestion/Change (if needed)**
M9: The requirement change identified by categorization/ classification on the basis of reason and origin of change has arranged correctly.	√	□	The processes of ARCM model for requirement change identified by categorization/ classification on the basis of reason and origin of change has arranged correctly. Moreover an appropriate procedure was used to analyse the requirement change and the selection of prioritization technique after requirement change. Changes were implemented by following all the standard activities of requirement change.
M10: Appropriate procedure was used to analyse the requirement change.	√	□	
M11: Appropriate procedure was followed to select prioritization technique after requirement change.	√	□	
M12: Changes were implemented by following all the standard activities of requirement change.	√	□	

Ranking: The score ranged from 0 to 4 Likert scale

where: 0 = Never, 1 = Rarely, 2 = Sometimes, 3 = Often and 4 = Always.

Assessment of the Critical Success Factor-People, Project, Organization, Technology		
Metrics	**Score**	**Comment**
Metric Factor: People		
M13: Team capability Team members are capable to follow the process of Agile.	0 1 2 3 4	**4**
M14: Motivated team members: The current progress of iteration was revealed to everyone in the team during iteration.	0 1 2 3 4	**2**
M15: Agile process knowledge. All the Team members have the basic Agile process knowledge	0 1 2 3 4	**4**
M16: Self-organizing teamwork. Team members have the autonomous and make any decision related to change.	0 1 2 3 4	**2**
M17: Good customer relationship: Customer and end-user involvement were monitored in change activity	0 1 2 3 4	**3**
M18: Customer involvement: Customers or Project Owner (PO) was available on-site for face-to-face discussions during the requirement change	0 1 2 3 4	**2**
Mean is calculated for each factor practices to come up with a final score for each quality factor. Then the average is divided by (4) the highest value of the score. The result is then multiplied by 100%.		**((4+2+4+2+3+2) / 6) /4 = 0.708*100 =70.83 %**
Metric Factor: Project		
M19: The nature of project was Non-life-critical.	0 1 2 3 4	**3**
M20: Within the project scope varies with emergent requirement.	0 1 2 3 4	**3**
M21: Project was implemented with small team	0 1 2 3 4	**3**
Mean is calculated for each factor practices to come up with a final score for each quality factor. Then the average is divided by (4) the highest value of the score. The result is then multiplied by 100%.		**((3+3+3) / 3) /4 = 0.75 *100 =75 %**
Metric Factor: Organizational		
M22: Organizational culture and mind set support frequent requirement change in Agile.	0 1 2 3 4	**4**
M23: Management of company support the process of change.	0 1 2 3 4	**3**
Mean is calculated for each factor practices to come up with a final score for each quality factor. Then the average is divided by (4) the highest value of the score. The result is then multiplied by 100%.		**((4+3) /2) /4 = 0.875 *100 =87.5 %**

Metric Factor: Technology		
M22: Organizational culture and mind set support frequent requirement change in Agile.	0 1 2 3 4	**3**
M23: Management of company support the process of change.	0 1 2 3 4	**3**
Mean is calculated for each factor practices to come up with a final score for each quality factor. Then the average is divided by (4) the highest value of the score. The result is then multiplied by 100%.		**((3+3) /2) /4 = 0.75 *100 =75 %**

Based on this percentage, each factor is assessed based on the (NPLF) rating scale that adapted from ISO/IEC 15504, where "N = not achieved (0 – 15%), P = partially achieved (>15- 50%), L = largely achieved (> 50 -85%) and F = fully achieved (> 85-100%)", which demonstrate fulfilment of the process factors.

Metric Factor	**Percentage**	**Indicator**
People	70.83%	Largely Achieved
Project	75%	Largely Achieved
Organization	87.5%	Fully Achieved
Technology	75%	Largely Achieved

- ACT PHASE- DEPLOYMENT AND VERIFICATION

Following checklist will be used during Act phase:

Event	**Agreed/Accepted**
Change is agreed as mentioned in the RFC	√
Timeline for deployment is acceptable to all stakeholders	√
Authorization for deployment	√

The Validation Form

Please validate and give comments on the below mentioned issues on the ARCM Model's implementation

Perceived Ease of Use	Extremely Agree	Moderately Agree	Agree	Neutral	Extremely Disagree	Moderately Disagree	Disagree
Representation of ARCM model is simple	√	☐	☐	☐	☐	☐	☐
It is simple to understand the process, activities and practices of Agile Requirement change Management (ARCM).	☐	√	☐	☐	☐	☐	☐
It is simple to understand and apply sustainability Analysis during change management process.	☐	√	☐	☐	☐	☐	☐
It is simple to implement ARCM model for change management process in Agile development.	√	☐	☐	☐	☐	☐	☐
ARCM Model needs training to fully understand it	☐	☐	☐	√	☐	☐	☐
Perceived Usefulness	**Extremely Agree**	**Moderately Agree**	**Agree**	**Neutral**	**Extremely Disagree**	**Moderately Disagree**	**Disagree**
Requirement change in Agile was difficult to perform without ARCM Model.	☐	☐	√	☐	☐	☐	☐
Using ARCM Model gives the greater control over Requirement Change	√	☐	☐	☐	☐	☐	☐

Management (RCM) process.							
ARCM Model enables to accomplish RCM process more quickly and save the time.	√	☐	☐	☐	☐	☐	☐
Using ARCM Model enhances the effectiveness of requirement change process in Agile.	√	☐	☐	☐	☐	☐	☐
Using ARCM Model improves the quality of the RCM process.	√	☐	☐	☐	☐	☐	☐
Using ARCM Model makes it easier to do change process.	√	☐	☐	☐	☐	☐	☐
Using ARCM Model increases the overall productivity.	☐	√	☐	☐	☐	☐	☐
Structure of ARCM Model	**Extremely Agree**	**Moderately Agree**	**Agree**	**Neutral**	**Extremely Disagree**	**Moderately Disagree**	**Disagree**
The core components of ARCM are self-exploratory and no need of further explanation to be used effectively.	☐	√	☐	☐	☐	☐	☐
The ARCM components are practical and could be used in software industry practicing Agile.	☐	√	☐	☐	☐	☐	☐
The implementation of ARCM could assist an organization to identify issues related to requirement change.	☐	☐	√	☐	☐	☐	☐
The components of ARCM model are enough to perform the complete process of change in Agile software development.	☐	√	☐	☐	☐	☐	☐

Appendix L

The Findings of the Second Case Study (Company-B)

Target	To determine the acceptability of the ARCM Model in the software company that was the novice in adopting Agile methodology		
Project	The case study was performed on the projects produced by organization-B, was Human Resource, Payroll & Attendance System which is a desktop-based software solution.		
Evaluation Team	**Roles**	**Skills**	**Experiences**
	Project Manager	Manager	14 Years
	Team Lead	Manager	12 Years
	Domain Expert	Agile Software Development	15 Years
	2-Technical Lead	Front End Development	12, 8 Years

➢ PLAN PHASE- CHANGE IDENTIFICATION PROCESS

Change Identification		
RFC No: 01		
RFC Description: New user should be able to navigate links within 05 minutes.		
Reason of Change	**Selection**	**Comment**
Product Strategy	☐	
Scope Reduction	☐	
Missing Requirements	☐	
Clarifying Requirements	√	
Functionality Enhancement	☐	
Design Improvement	☐	
Redundant Functionality	☐	
Obsolete Functionality	☐	
Erroneous Requirements	☐	
Resolving Conflicts	☐	
Defect Fixing	☐	
Other	☐	

Origin of change	Selection	Comment
Marketing Group.	☐	
Project Management Consideration.	☐	
Developer's Detailed Analysis.	☐	
Engineering's Call.	☐	
Customer-Support discussions	☐	
Design Review Feedback	☐	
Technical Team Discussion.	√	
Defect Report	☐	
Other	☐	

Additional Comments:

Sustainability Characteristics Identification	Selection	Comment
Performance Efficiency: The characteristic represents the performance relative to the amount of resources used under stated conditions.		
Time behaviour (The capacity of a product while performing its functions to meet requirements regarding response time, throughput and processing time)	☐	
Resource utilization (When performing its functions, the amounts and types of resources used by a product or system to meet requirements.	☐	
Other Important sub-characteristics related to **Performance Efficiency** (please identify, if any):	☐	
Compatibility: A product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment.		
Interoperability (degree to which two or more systems, products or components can exchange information and use the information that has been exchanged.)	☐	
Communication commonality (The degree to which software is dependent on its associated hardware.)	☐	
Data commonality (The use of standard data representation.)	☐	
Other Important sub-characteristics related to **Compatibility** (please identify, if any):	☐	
Usability: Feature that enable the system to be user friendly and to achieve identified goals with efficiency, satisfaction and effectiveness.		
Learnability (degree to which a product or system enables the user to learn how to use it with effectiveness, efficiency in	☐	

emergency situations.)		
Understandability (The ability to recognize whether the software is appropriate for the user needs.)	☐	
Operability (The capacity of a product that make it easy to operate and control.)	☑	
Attractiveness (User interface assists satisfying and pleasing interaction for the user.)	☐	
Other sub-characteristics related to **Usability** (please identify, if any):	☐	
Reliability: The capacity of a product to perform specified functions for a specified period of time, under specified conditions.		
Availability (The capacity of a product or component to be accessible and operational at the time of use.)	☐	
Fault tolerance (The capacity of a product to be operated as required in spite of the existence of hardware or software faults.)	☐	
Recoverability (The capacity of a product to recover the data and restore the state of the system during the event of an interruption or a failure.)	☐	
Other sub-characteristics related to **Reliability** (please identify, if any):	☐	
Security: Degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.		
Confidentiality (degree to which the prototype ensures that data are accessible only to those authorized to have access.)	☐	
Integrity (degree to which a system, product or component prevents unauthorized access to, or modification of, computer programs or data.)	☐	
Authenticity (degree to which the identity of a subject or resource can be proved to be the one claimed.)	☐	
Other sub-characteristics related to **Security** (please identify, if any):	☐	
Maintainability: The characteristic represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in environment, and in requirements.)		
Modularity (A discrete component that does not have an impact on other components during change.)	☐	
Analysability (A discrete component that does not have an impact on other components during change. Diagnose a product for deficiencies or causes of failures due to change.)	☐	
Modifiability (Degree to which a product or system can be effectively and efficiently modified without introducing defects or degrading existing product quality.)	☐	
Extensibility (The level of effort required to implement the extension.)	☐	
Reusability (The capacity of a module or component that can be re-used in other software system.)	☐	
Testability (The capacity to construct the test criteria with effectiveness and perform tests to meet those criteria.)	☐	
Other sub-characteristics related to **Maintainability** (please identify, if any):	☐	
Portability: Capability of the system to run under different computing environments		

Adaptability (The capacity of product or system to be adopted in different hardware, software or other operational environments.)	☐	
Installability (The capacity of a product to be successfully installed or uninstalled in any environment.)	☐	
Replaceability (The capacity of replacement of a product by another software product in the same environment.)	☐	
Other sub-characteristics related to **Portability** (please identify, if any):	☐	
Functional Suitability: The degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.		
Functional Completeness (degree to which the set of functions covers all the specified tasks and user objectives.)	☐	
Functional Correctness (degree to which the functions provides the correct results with the needed degree of precision.)	☐	
Functional Appropriateness (degree to which the functions facilitate the accomplishment of specified tasks and objectives.)	☐	
Other sub-characteristics related to **Functional Suitability** (please identify, if any):	☐	

➢ DO PHASE- SUSTAINABILITY ANALYSIS

Risks Associated with Cost "CO" (Risks associated with cost of developing the project/product)

Sr. #	RISK TYPE	**Impact On**
1	Business Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐
2	Technical Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐
3	**Schedule Risk**	Development Effort (Cost) ☐ **Project Duration** √ Quality of Project ☐
4	Cost Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐
5	Quality Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐
6	People Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐
7	Project Risk	Development Effort (Cost) ☐ Project Duration ☐ Quality of Project ☐

Association between Risk and the impact of change using Chi-Squares Tests

Type of Risks	Asymp. Sig. (2-sided)		
	Development Effort (Cost) p-value	Project Duration p-value	Quality of Project p-value
Business Risk	0.000	0.000	0.000
Technical Risk	0.001	0.000	0.000
Schedule Risk	0.017	0.000	0.003
Cost Risk	0.000	0.002	0.000
Quality Risk	0.000	0.031	0.000
People Risk	0.166	0.000	0.253
Project Risk	0.000	0.000	0.739

The Value of Impact w.r.t Chi-Squares Value

Score	Impact	p-value
5	Severe	000 to 0.002
4	Significance	> 0.002 to 0.004
3	Moderate	> 0.004 to 0.006
2	Minor	> 0.006 to 0.008
1	Negligible	> 0.008 to 0.01

Risks Associated With Schedule "Sch" (Risks associated with schedule of developing the project/product)

Sr. #	DESCRIPTION	Applicable	Impact
1	New requirements/enhancements will effect the schedule	Yes	5
2	The milestones that are set will not be met	N/A	
3	Milestones are not flexible and realistic	N/A	
4	Costs associated with a defective product	N/A	

RFC-1: **System should be able to cater 10,000 users simultaneously.**							
Risk Type	Sustainable Characteristics **Dimension**	**Impact Type(s)**	**Impact Value**	**Probability**	**Risk Exposure**	**Risk Exposure Description** *Refer to Table 6* Risk Exposure *(*Appendix I - Guideline Document for ARCM Model*)*	**Impact Description By Practitioner**
Business Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
Technical Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
Schedule Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) √ Project Quality □	5	1	5	**Economic**: Reasonable budget estimated. Adequate resource to manage changes in the same iteration. **Social**: Capable to satisfy customer/stakeholder and mange changes. **Environment**: Able to comply with long term usage without additional cost. **Overall**: Expected loses have very little impact on project success.	New requirements can effect the schedule but it can be manageable within the same iteration
Cost Risk	Economics □ Environment □	Development Effort (Cost) □					

RFC-1: **System should be able to cater 10,000 users simultaneously.**							
Risk Type	Sustainable Characteristics **Dimension**	**Impact Type(s)**	**Impact Value**	**Probability**	**Risk Exposure**	**Risk Exposure Description** *Refer to Table 6* Risk Exposure *(*Appendix I - Guideline Document for ARCM Model*)*	**Impact Description By Practitioner**
	Social □	Project duration (Schedule) □ Project Quality □					
Quality Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
People Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
Project Risk	Economics □ Environment □ Social □	Development Effort (Cost) □ Project duration (Schedule) □ Project Quality □					
		Total Probability		20 %			

Legends							
Score	**Impact**	**Score**	**Probability**	**Type**	**Description**	**Type**	**Description**
5	Severe	5	Expected	BS	Business Risk	PE	People Risk
4	Significance	4	High	TE	Technical Risk	PE	Project Risk
3	Moderate	3	Moderate	SC	Schedule Risk		
2	Minor	2	Low	CO	Cost Risk		
1	Negligible	1	Not likely	QU	Quality Risk		

Selection of Requirement Prioritization Technique

Refer to Table 7 Guideline to select prioritization technique after requirement change

*(*Appendix I Guideline Document for ARCM Model*)*

RP Techniques	Choice of RP Technique	Reason/Comment
Cost-value ranking		
Value-oriented prioritization (VOP)		
Cumulative Voting		
MoSCoW		
Planning Game		
Analytic hierarchy process (AHP)		
Quality Functional Deployment (QFD)		
Binary Search Tree (BST)		
Wiegers' matrix approach		
Kano Model	√	Kano is more concerned to the customer preferences and it is the fastest way to prioritize requirements.
Pair wise analysis		

➢ **CHECK PHASE- ASSESSMENT OF THE ARCM MODEL**

Assessment of the Agile RCM Process			
Metric Factor: Process- Completeness Metrics			
Metrics(M)	**Yes**	**No**	**Suggestion/Change (if needed)**
M1: The request of change is identified by considering the source and reason of changes	√	□	The proposed ARCM model is found to be adequate and sufficient in managing RCM process in Agile in the real world environment.
M2: Sustainability analysis was performed using Risk matrix	√	□	
M3: After change prioritization was performed using the guideline provided.	√	□	
M4: Change are implemented successfully	√	□	
Metric Factor: Process- Consistency Metrics			
Metrics(M)	**Yes**	**No**	**Suggestion/Change (if needed)**
M5: Changes are identified by considering change categorization appropriately.	√	□	The participants found that the proposed model is internally consistent, and all the processes incorporate each other. In particular, it starts with identification of reason and origin of change that is followed by the change analysis on the basis of impact and risk analysis. The RCM process is then continued with proper implementation of change.
M6: Requirement change analysis was performed by following the standards of requirement change analysis.	√	□	
M7: Requirement prioritization was done after every change.	√	□	
M8: Appropriate procedure was followed to implement the requirement change.	√	□	
Metric Factor: Process- Accuracy Metrics			
Metrics(M)	**Yes**	**No**	**Suggestion/Change (if needed)**
M9: The requirement change identified by categorization/ classification on the basis of reason and origin of change has arranged correctly.	√	□	All the processes of ARCM model has arranged correctly and following all the standard activities of managing requirement change in ASD.
M10: Appropriate procedure was used to analyse the requirement change.	√	□	
M11: Appropriate procedure was followed to select prioritization technique after requirement change.	√	□	
M12: Changes were implemented by following all the standard activities of requirement change.	√	□	

Ranking: The score ranged from 0 to 4 Likert scale

where: 0 = Never, 1 = Rarely, 2 = Sometimes, 3 = Often and 4 = Always.

Assessment of the Critical Success Factor-People, Project, Organization, Technology		
Metrics	**Score**	**Comment**
Metric Factor: People		
M13: Team capability Team members are capable to follow the process of Agile.	0 1 2 3 4	**3**
M14: Motivated team members: The current progress of iteration was revealed to everyone in the team during iteration.	0 1 2 3 4	**1**
M15: Agile process knowledge. All the Team members have the basic Agile process knowledge	0 1 2 3 4	**3**
M16: Self-organizing teamwork. Team members have the autonomous and make any decision related to change.	0 1 2 3 4	**1**
M17: Good customer relationship: Customer and end-user involvement were monitored in change activity	0 1 2 3 4	**2**
M18: Customer involvement: Customers or Project Owner (PO) was available on-site for face-to-face discussions during the requirement change	0 1 2 3 4	**2**
Mean is calculated for each factor practices to come up with a final score for each quality factor. Then the average is divided by (4) the highest value of the score. The result is then multiplied by 100%.		**((3+1+3+1+2+2) / 6) /4 = 0.5*100 =50 %**
Metric Factor: Project		
M19: The nature of project was Non-life-critical.	0 1 2 3 4	**4**
M20: Within the project scope varies with emergent requirement.	0 1 2 3 4	**4**
M21: Project was implemented with small team	0 1 2 3 4	**3**
Mean is calculated for each factor practices to come up with a final score for each quality factor. Then the average is divided by (4) the highest value of the score. The result is then multiplied by 100%.		**((4+4+3) / 3) /4 = 0.91 *100 =91 %**
Metric Factor: Organizational		
M22: Organizational culture and mind set support frequent requirement change in Agile.	0 1 2 3 4	**2**
M23: Management of company support the process of change.	0 1 2 3 4	**3**
Mean is calculated for each factor practices to come up with a final score for each quality factor. Then the average is divided by (4) the highest value of the score. The result is then multiplied by 100%.		**((2+3) /2) /4 = 0.625 *100 =62.5 %**

Metric Factor: Technology		
M22: Organizational culture and mind set support frequent requirement change in Agile.	0 1 2 3 4	**2**
M23: Management of company support the process of change.	0 1 2 3 4	**2**
Mean is calculated for each factor practices to come up with a final score for each quality factor. Then the average is divided by (4) the highest value of the score. The result is then multiplied by 100%.		**((2+2) /2) /4 = 0.5 *100 =50 %**

Based on this percentage, each factor is assessed based on the (NPLF) rating scale that adapted from ISO/IEC 15504, where "N = not achieved (0 – 15%), P = partially achieved (>15- 50%), L = largely achieved (> 50 -85%) and F = fully achieved (> 85-100%)", which demonstrate fulfilment of the process factors.

Metric Factor	**Percentage**	**Indicator**
People	50%	Partially Achieved
Project	91%	Fully Achieved
Organization	62.5%	Largely Achieved
Technology	50%	Partially Achieved

- ACT PHASE- DEPLOYMENT AND VERIFICATION

Following checklist will be used during Act phase:

Event	**Agreed/Accepted**
Change is agreed as mentioned in the RFC	√
Timeline for deployment is acceptable to all stakeholders	√
Authorization for deployment	√

Appendix M

Cross Case Analysis of Company A and B

The overall perception on the ARCM Model's implementation

Perceived Ease of Use	Extremely Agree	Moderately Agree	Agree	Neutral	Extremely Disagree	Moderately Disagree	Disagree
Representation of ARCM model is simple	☐	√	☐	☐	☐	☐	☐
It is simple to understand the process, activities and practices of Agile Requirement change Management (ARCM).	☐	☐	√	☐	☐	☐	☐
It is simple to understand and apply sustainability Analysis during change management process.	☐	☐	√	☐	☐	☐	☐
It is simple to implement ARCM model for change management process in Agile development.	√	☐	☐	☐	☐	☐	☐
ARCM Model needs training to fully understand it	☐	☐	☐	☐	☐	☐	√

Perceived Usefulness	Extremely Agree	Moderately Agree	Agree	Neutral	Extremely Disagree	Moderately Disagree	Disagree
Requirement change in Agile was difficult to perform without ARCM Model.	☐	☐	☐	√	☐	☐	☐
Using ARCM Model gives the greater control over Requirement Change Management (RCM) process.	☐	√	☐	☐	☐	☐	☐
ARCM Model enables to accomplish RCM process more quickly and save the time.	☐	√	☐	☐	☐	☐	☐
Using ARCM Model enhances the effectiveness of requirement change process in Agile.	√	☐	☐	☐	☐	☐	☐
Using ARCM Model improves the quality of the RCM process.	☐	☐	√	☐	☐	☐	☐
Using ARCM Model makes it easier to do change process.	☐	√	☐	☐	☐	☐	☐
Using ARCM Model increases the overall productivity.	☐	☐	√	☐	☐	☐	☐
Structure of ARCM Model	**Extremely Agree**	**Moderately Agree**	**Agree**	**Neutral**	**Extremely Disagree**	**Moderately Disagree**	**Disagree**
The core components of ARCM are self-exploratory and no need of further explanation to be used effectively.	☐	☐	☐	√	☐	☐	☐
The ARCM components are practical and could be used in software industry practicing Agile.	☐	☐	√	☐	☐	☐	☐

The implementation of ARCM could assist an organization to identify issues related to requirement change.	☐	☐	☐	√	☐	☐	☐
The components of ARCM model are enough to perform the complete process of change in Agile software development.	☐	☐	√	☐	☐	☐	☐

Cross Case Analysis of Company A and B - Perceived Ease of Use-PEOU

Perceived Ease of Use (PEOU)	No of Participants (n=2)									Comment
	Positive (+ Ve)				Negative (- Ve)					
	Extremely Agree (EA)	Moderately Agree (MA)	Agree (A)	EA+MA+A	Extremely Disagree (ED)	Moderately Disagree (MD)	Disagree (D)	ED+MD+D	Neutral	
Representation of ARCM Model is simple.	1	1	0	**2**	0	0	0	**0**	0	The result reveals that the model was perceived as easy to use due to its well defined process activities and techniques. The result represents that the participants were positively agreed with the questions regarding the perceived ease of use (PEOU) of ARCM Model. Moreover one participant was disagree concerning the need of conducting training sessions to completely understand the use of model and one was neutral
It is simple to understand the process, activities and practices of Agile Requirement change Management (ARCM).	0	1	1	**2**	0	0	0	**0**	0	
It is simple to understand and apply sustainability Analysis during change management process.	0	1	1	**2**	0	0	0	**0**	0	
It is simple to implement ARCM Model for change management process in Agile development.	2	0	0	**2**	0	0	0	**0**	0	
ARCM Model needs the training session for the team members to fully understand it	0	0	0	**0**	0	0	1	**1**	1	

Cross Case Analysis of Company A and B - Perceived Usefulness (PU)

Perceived Usefulness (PU)	No of Participants (n=2)									Comment
	Positive (+ Ve)				Negative (- Ve)				Neutral	
	Extremely Agree (EA)	Moderately Agree (MA)	Agree (A)	EA+MA+A	Extremely Disagree (ED)	Moderately Disagree (MD)	Disagree (D)	ED+MD+D		
Requirement change in Agile was difficult to perform without ARCM Model.	0	0	1	**1**	0	0	0	**0**	1	All the participants have positively responded and highlighted that it would be useful to their company as this model enables to accomplish RCM process more quickly and save the time and make easier to manage changes in Agile development. Moreover, both companies have selected the positive category of the questionnaire. It shows that the organization were confident with the assessment results of ARCM.
Using ARCM Model gives the greater control over Requirement Change Management (RCM) process.	1	1	0	**2**	0	0	0	**0**	0	
ARCM Model enables to accomplish RCM process more quickly and save the time.	1	1	0	**2**	0	0	0	**0**	0	
Using ARCM Model enhances the effectiveness of requirement change process in Agile.	2	0	0	**2**	0	0	0	**0**	0	
Using ARCM Model improves the quality of the RCM process.	1	0	1	**2**	0	0	0	**0**	0	
Using ARCM Model makes it easier to do change process.	1	1	0	**2**	0	0	0	**0**	0	
Using ARCM Model increases the overall productivity.	0	1	1	**2**	0	0	0	**0**	0	

Cross Case Analysis of Company A and B - Structure of ARCM

Structure of ARCM	No of Participants (n=2)									Comment
	Positive (+ Ve)				Negative (- Ve)				Neutral	
	Extremely Agree (EA)	Moderately Agree (MA)	Agree (A)	EA+MA+A	Extremely Disagree (ED)	Moderately Disagree (MD)	Disagree (D)	ED+MD+D		
The core components of ARCM are self-exploratory and no need of further explanation to be used effectively.	0	1	0	**1**	0	0	0	**0**	1	The participants agreed that the proposed ARCM Model is well structured and it contains all the components which were enough to perform the complete process of change in Agile. Moreover, the model is practical and could be used in software industry practicing Agile..
The ARCM components are practical and could be used in software industry.	0	1	1	**2**	0	0	0	**0**	0	
The implementation of ARCM could assist an organization to identify issues related to requirement change.	0	0	1	**1**	0	0	0	**0**	1	
The components of ARCM Model are enough to perform the complete process of change in Agile software development.	0	1	1	**2**	0	0	0	**0**	0	