THE INVESTIGATION ON THE BEST PRACTICES OF EXTREME PROGRAMMING (XP) QUALITY IMPLEMENTATION AT UUM IT

RANA ALAULDEEN ABDULRAHMAN

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)
UNIVERSITI UTARA MALAYSIA
2015

Permission to Use

In presenting this dissertation in partial 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 dissertation 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 dissertation 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 dissertation.

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

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

Abstrak

Kejuruteraan perisian (SE) memainkan peranan yang penting dalam meningkatkan kesejahteraan masyarakat melalui penggunaan perisian yang berkualiti tinggi. Kebanyakan projek perisian gagal disebabkan organisasi perisian tidak mempraktis amalan pembangunan perisian yang sewajarnya. Sehubungan itu, organisasi perisian perlu mempunyai metodologi pembangunan perisian yang baik bagi memenuhi keperluan pihak pemegang taruh. Salah satu metodologi pembangunan perisian dalam SE yang semakin berkembang penggunaannya adalah metodologi *Extreme Programming* (XP). Metodologi ini merupakan pendekatan baru dalam SE yang mampu meningkatkan kualiti perisian dan berupaya mengurangkan masa pembangunan perisian dan kos. Walau bagaimanapun, tahap penggunaan metodologi ini di kalangan pembangun perisian di Pusat UUM IT masih tidak jelas. Oleh yang demikian, kajian ini bertujuan untuk mengkaji penggunaan amalan XP di pusat ini. UUM IT telah dipilih sebagai kajian kes kerana peranan organisasi ini telah berubah bagi memenuhi permintaan yang tinggi di kalangan masyarakat kampus. Oleh itu, penyelidikan yang memfokuskan kepada kepada 12 amalan XP di UUM IT amat diperlukan. Kajian ini dijalankan dengan menemubual secara separa berstruktur dengan lima (5) pakar dari UUM IT bagi mengenal pasti kejayaan pelaksanaan amalan XP. Hasil kajian telah menunjukkan bahawa sebahagian besar daripada amalan XP digunakan oleh pembangun perisian di UUM IT tetapi perlu dipertingkatkan Sebaliknya, beberapa amalan seperti *pair programming* dan *test first programming* tidak digunakan oleh pembangun perisian di UUM IT. Ini disebabkan jenis dan sifat projek perisian yang terlibat, dan juga disebabkan oleh personaliti, pengalaman dan tahap pendidikan yang berbeza di kalangan pembangun perisian. Kajian ini menyediakan bukti kualitatif yang dapat membantu pengurus projek perisian dalam membimbing mereka meningkatkan amalan pembangunan perisian bagi menghasilkan perisian yang berkualiti tinggi.

Abstract

Software engineering (SE) plays an important role for improving society's well-being through the use of high quality software. There is noted that most of the software projects are failed, due to missing or poor software development practices in software organizations. Due to this reason, having a good and sound software development methodology is crucial for software organization to satisfy stakeholder's requirements. One of the prevalent software development methodologies in SE is Extreme programming (XP) methodology. This methodology is an emerging SE approach, which is able to increase software quality and hence reducing software development time and cost. However, the level of application of this methodology among software developers in UUM IT centre is still unclear. Therefore, this study aims to investigate the application of XP practices in this centre. UUM IT was chosen as a case study because the role of this organization has changed to meet high demand among campus communities. Thus, research that focuses on the 12 XP practices of UUM IT is highly needed. This study was conducted using a semi–structured interview with five (5) experts from the UUM IT, to identify the successful implementation of the XP practices. The findings have shown that, most of the practices are used by UUM IT developers but need to improve. In contrast, some of the practices such as pair programming and test first programming are not used by the UUM IT developers. This is due to the nature and type of software projects involved, also because of the personality, experiences and the education level differences among developers. This study provides qualitative evident that can assist software project managers to guide them in improving software development practices for producing high quality software.

Acknowledgement

In the name of Allah Gracious and most Merciful. In the first place, I wish to express my deepest gratitude to Allah for providing me with the substance, time, health, strength and patience to participate in this journey to acquire knowledge.
In accomplishing this research, I would like to express my gratitude to Dr. Mazni Omar for supervision, advice, and guidance of this research as well as giving me from her experiences.

My deepest thanks to my father Alauldeen Abdulrahman who put the fundamental of my learning character since I was a child and my mother Luma Alauldeen who sincerely raised me with her caring and gently love. Thank you for your love, your support, your prayers, for everything you did for me during my study.

Table of Contents

List of Tables

List of Figures

List of Appendices

CHAPTER ONE
INTRODUCTION

1.1 Overview

This initial chapter introduces the background on the phenomenon under study, problem statement, research questions, and research objectives. The research scope and significance of this research are also discussed. The chapter ends with the outline of the thesis structure and summary of the current chapter.

1.2 Background of Study

Software engineering (SE) is a domain that deals with engineering discipline in software construction. It has been kept formal and has practical methodologies as guidance in software development. It has been manifested by software life cycle that is composed of requirement elicitation and analysis, design specification, implementation, verification and validation, deployment and maintenance (Wu, 2011). Software development processes are an important part of software engineering, which influence the product outcome (Senapathi & Srinivasan, 2012; Päivärinta & Smolander, 2015). Several studies noted that software projects are considered a failure for many reasons. Tan (2011) refers that the research conducted by Gartner where data was collected from 845 project sample has shown that 42.5% did not deliver all the benefits, 44% were delivered over budget and 42% were not delivered on time. Furthermore, Gulla (2011) mentions that missing methodology is one of the reasons for software failure. The reason of software failure has also been discussed by Haughey (2011), who claims that poor or missing methodologies and tools are among the reasons.

REFERENCES

Abdullah, M. S., al-Tarawnehb, M. Y., & Alia, A. B. M. (2012). Software process improvement in small software development firms. *Computer Science, 1*, 782-787.

Abrahamsson, P. (2003). *Extreme programming: first results from a controlled case study*. Paper presented at the Euromicro Conference, 2003. Proceedings. 29th.

Abrahamsson, P., Conboy, K., & Wang, X. (2009). "Lots done, more to do": the current state of agile systems development research.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. Agile Software Development Methods: Review and Analysis. 2002. VTT Publications: Finland.

Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New directions on agile methods: a comparative analysis. Paper presented at the Software Engineering, 2003. Proceedings. 25th International Conference on.

Abrantes, J. F., & Travassos, G. H. (2011). *Common agile practices in software processes.* Paper presented at the Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on.

Ackroyd, S., & Hughes, J. A. (1992). *Data collection in context*: Longman London.

Agarwal, N., & Deep, P. (2014). *Obtaining better software product by using test first programming technique.* Paper presented at the, 2014 5th International Conference Confluence The Next Generation Information Technology Summit(Confluence).

Aguanno, K. (2004). *101 Ways to Reward Team Members for $20 (or Less!)*: Multi-Media Publications Inc.

Ahlemann, F., El Arbi, F., Kaiser, M. G., & Heck, A. (2013). A process framework for theoretically grounded prescriptive research in the project management field. *International Journal of Project Management, 31*(1), 43-56.

Alite, B., & Spasibenko, N. (2008). Project Suitability for Agile methodologies. *Umeå School of Business*.

Alshehri, S. A. J. (2014). *AHP-Based Methodology for a Complex Decision Support in Extreme Programming.* Faculty of Graduate Studies and Research, University of Regina.

Al-Tarawneh, M. Y. (2013). *Harmonizing CMMI-DEV 1.2 and XP Method to Improve The Software Development Processes in Small Software Development Firms.* Universiti Utara Malaysia.

Ambu, W., & Gianneschi, F. (2003). Extreme programming at work *Extreme Programming and Agile Processes in Software Engineering* (pp. 347-350): Springer.

Asnawi, A. L., Gravell, A. M., & Wills, G. B. (2012). *Emergence of agile methods: perceptions from software practitioners in Malaysia.* Paper presented at the AGILE India (AGILE INDIA), 2012.

Asnawi, A. L., Gravell, A. M., & Wills, G. B. (2014). *Significant aspects in relation to Agile usage: Malaysian perspective.* Paper presented at the Information and Communication Technology (ICoICT), 2014 2nd International Conference on.

Aveling, B. (2004). XP lite considered harmful? Extreme Programming and Agile Processes in Software Engineering (pp. 94-103): Springer.

Avison, D., Cole, M., & Fitzgerald, G. (2006). Reflections on teaching information systems analysis and design: from then to now! *Journal of Information Systems Education, 17*(3), 253.

Baker, S. E., & Edwards, R. (2012). How many qualitative interviews is enough.
Beck, K. (1999). Embracing change with extreme programming. Computer, 32(10), 70-77.

Beck, K. (2000). *Extreme programming explained: embrace change*: Addison-Wesley Professional.

Beck, K. (2002). The metaphor metaphor. *Keynote speech-ACM OOPSLA, 2.*

Becker, C. H. (2010). Using eXtreme Programming in a Student Environment.

Begel, A., & Nagappan, N. (2008). *Pair programming: what's in it for me?* Paper presented at the Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement.

Bird, M. (2007). *Comprehensive Examination Written Responses Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy.* Capella University.

Blokdijk, A. (2014). *Planning and design of information systems*: Academic Press.

Boehm, B. (2006). *A view of 20th and 21st century software engineering.* Paper presented at the Proceedings of the 28th international conference on Software engineering.

Bowers, J., May, J., Melander, E., Baarman, M., & Ayoob, A. (2002). Tailoring XP for large system mission critical software development *Extreme Programming and Agile Methods—XP/Agile Universe 2002* (pp. 100-111): Springer.

Burman, E. (2015). Agile in action: Hybrid methodologies in practice.

Bustard, D., Wilkie, G., & Greer, D. (2013). *The maturation of agile software development* presented at the Engineering of Computer Based Systems (ECBS), 2013 20th IEEE International Conference and Workshops on the. *Principles and practice: observations on successive industrial studies in 2010 and 2012.* Paper

Cagle West, M. (2010). Effective Software Engineering Leadership for Development Programs. *ProQuest LLC.*

Cano, S. P., González, C. S., Collazos, C. A., Arteaga, J. M., & Zapata, S. (2015). Agile Software Development Process Applied to the Serious Games Development for Children from 7 to 10 Years Old. *International Journal of Information Technologies and Systems Approach (IJITSA), 8*(2), 64-79.

Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2004). *How extreme does extreme programming have to be? Adapting XP practices to large-scale projects.* Paper presented at the System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on.

Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems, 18*(4), 332-343.

Chandra Misra, S., Kumar, V., & Kumar, U. (2010). Identifying some critical changes required in adopting agile practices in traditional software development projects. *International Journal of Quality & Reliability Management, 27*(4), 451-474.

Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2012). *Non-functional requirements in software engineering* (Vol. 5): Springer Science & Business Media.

Cockburn, A., & Highsmith, J. (2001). Agile software development: The people factor. *Computer*(11), 131-133.

Cockburn, A., 2007. Agile Software Development: A Cooperative Game. 2nd Edn., Addison Wesley, ISBN: 0-321-48275-1, pp: 504.

Cohn, M. (2005). *Agile estimating and planning*: Pearson Education.

Conboy, K., & Fitzgerald, B. (2010). Method and developer characteristics for effective agile method tailoring: A study of XP expert opinion. *ACM Transactions on Software Engineering and Methodology (TOSEM), 20*(1), 2.

Creswell, J. (2009). *Research design: Qualitative, quantitative, and mixed methods approaches*: SAGE Publications, Incorporated.

Creswell, J. W. (2005). *Educational Research: Planning, Conducting and Evaluating Qualitative and Quantitative Research.* New Jersey. Pearson.

Creswell, J. W. (2006). *Educational Research: Planning, Conducting and Evaluating Qualitative and Quantitative Research*. New Jersey. Pearson.

Creswell, J. W. (2012). Qualitative inquiry and research design: Choosing among five approaches: Sage.

Creswell, J. W. (2013). Research design: Qualitative, quantitative, and mixed methods approaches: Sage publications.

Creswell, J. W., & Clark, V. L. P. (2007). Designing and conducting mixed methods research.

Cyganek, B., & Siebert, J. P. (2011). *An introduction to 3D computer vision techniques and algorithms*: John Wiley & Sons.

Da Silva Estácio, B. J., & Prikladnicki, R. (2014). *A Set of Practices for Distributed Pair Programming.* Paper presented at the ICEIS (2).

Da Silva Estácio, B. J., & Prikladnicki, R. (2015). Distributed Pair Programming: A Systematic Literature Review. *Information and Software Technology, 63,* 1-10.

Darwish, N. R. (2011). Improving the Quality of Applying eXtreme Programming (XP) Approach. *The International Journal of Computer Science and Information Security, 9*(11), 16.

Darwish, N. R. (2013). Towards an Approach for Evaluating the Implementation of eXtreme Programming Practices. *International Journal of Intelligent Computing and Information Sciences (IJICIS), Ain Shams University, 13*(3).

Dey, I. (1993). *Qualitative Data Analysis: A User Friendly Guide for Social Scientists*, London, Routledge.

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, 85*(6), 1213-1221.

Douglas, I. (2006). Issues in software engineering of relevance to instructional design. *TechTrends, 50*(5), 28-35.

Dubinsky, Y., & Hazzan, O. (2002). *Improvement of software quality: Introducing extreme programming into a project-based course.* Paper presented at the 14th International Conference of the Israel Society for Quality.

Dudziak, T. (1999). eXtreme Programming An Overview. *Methoden und Werkzeuge der Softwareproduktion WS, 2000*.

Eckstein, J. (2013). *Agile software development in the large: Diving into the deep*: Pearson Education.

Elssamadisy, A. (2008). *Agile adoption patterns: a roadmap to organizational success*: Addison-Wesley Professional.

Flick, U. (2015). Introducing research methodology: A beginner's guide to doing a research project: Sage.

Fontana, A., & Frey, J. H. (2005). The interview: From neutral stance to political involvement. *The Sage handbook of qualitative research, 3,* 695-728.

Fowler, M., & Foemmel, M. (2006). Continuous integration. *Thought-Works) http://www. thoughtworks. com/Continuous Integration. pdf*.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development, 9*(8), 28-35.

Fruhling, A., & Vreede, G.-J. D. (2006). Field experiences with eXtreme programming: developing an emergency response system. *Journal of Management Information Systems, 22*(4), 39-68.

Gable, G. G. (1994). Integrating case study and survey research methods: an example in information systems. European Journal of Information Systems, 3(2), 112-126.

Ghani, I., Izzaty, N., & Firdaus, A. (2013). Role-based Extreme Programming (XP) for secure software development. *Science International (Lahore), 25*(4 (Spe), 1071-1074

Gittins, R., Hope, S., & Williams, I. (2001). *Qualitative studies of xp in a medium sized business.* Paper presented at the Proceedings of the 2nd Conference on eXtreme programming and flexible processes in software engineering, Cagliari, Italy.

Goldhor, H. (1972). Introduction to scientific research in librarianship: University of Illinois, Graduate School of Library Science.

Guha, P., Shah, K., Shukla, S. S. P., & Singh, S. (2011). Incorporating Agile with MDA Case Study: Online Polling System. *arXiv preprint arXiv:1110.6879*.

Gulla, J. (2011). *Seven reasons why information technology projects fail.* Paper presented at the SHARE Conference.

Haider, M. T., & Ali, I. (2011). Evaluation of the Effects of Pair Programming on Performance and Social Practices in Distributed Software Development.

Haider, M. T., & Ali, I. (2011). Evaluation of the Effects of Pair Programming on Performance and Social Practices in Distributed Software Development.

Harrison, N. B. (2003). A study of extreme programming in a large company. *Avaya Labs*.

Hass, K. B. (2007). The blending of traditional and agile project management. *PM world today, 9*(5), 1-8.

Haughey, D. (2011). The Four Levels of Project Success–The Project Management Maturity Matrix. *URL: http://www. projectsmart. co. uk/four-levels-of-project-success. html, retrieval on, 20*(11).

Hernon, P. (1991). The elusive nature of research in LIS. *Library and information science research: Perspectives and strategies for improvement*, 3-14.
Highsmith, J. (2000). Extreme programming.

Highsmith, J. (2000). Retiring Lifecycle Dinosaurs A look at Adaptive Software Development, an alternative to traditional, process-centric software management methods. *Software testing and quality engineering, 2,* 22-30.

Highsmith, J. (2013). *Adaptive software development: a collaborative approach to managing complex systems*: Addison-Wesley.

Hneif, M., & Hock Ow, S. (2009). Review of Agile Methodologies in Software Development. *International Journal of Research and Reviews in Applied Sciences, 1*(1). 1-8.

Hockey, L. (1984) The nature and purpose of research. In Cormack, D.F.S.(ed) *The Research Process in Nursing, (1st edn).* London: Blackwell Science, 1-10.

Hummel, M. (2014). *State-of-the-Art: A Systematic Literature Review on Agile Information Systems Development.* Paper presented at the System Sciences (HICSS), 2014 47th Hawaii International Conference on.

Hussain, Z., Lechner, M., Milchrahm, H., Shahzad, S., Slany, W., Umgeher, M., & Wolkerstorfer, P. (2008). *Integrating Extreme Programming and User-Centered Design.* Paper presented at the PPIG'08: Proceedings of the 20th annual meeting of the Psychology of Programming Interest Group, Lancaster, UK

Ishak, I. S., & Alias, R. A. (2005). Designing a strategic information systems planning methodology for malaysian institutes of higher learning (isp-ipta).

Jeffries, R. (2003). Extreme Programming and Agile Software Development Methodologies: CRC Press LLC.

Jeffries, R., Anderson, A., & Hendrickson, C. (2001). *Extreme programming installed*: Addison-Wesley Professional.

Jun, L., Qiuzhen, W., & Lin, G. (2010). *Application of agile requirement engineering in modest-sized information systems development.* Paper presented at the Software Engineering (WCSE), 2010 Second World Congress on.

Kalermo, J., & Rissanen, J. (2002). Agile software development in theory and practice. *University of Jyväskylä*.

Kaplan, B., & Duchon, D. (1988). Combining qualitative and quantitative methods in information systems research: a case study. MIS quarterly, 571-586.

Kircher, M. (2001). *eXtreme programming in open-source and distributed environments.* Paper presented at the JAOO (Java And Object-Orientation) conference, Aarhus, Dinamarca.

Kircher, M., & Levine, D. L. (2000). The XP of TAO: extreme programming of large, open-source frameworks.

Kircher, M., Jain, P., Corsaro, A., & Levine, D. (2001). Distributed extreme programming. *Extreme Programming and Flexible Processes in Software Engineering, Italy*, 66-71.

Kongyai, B., & Edi, E. (2011). Adaptation of Agile Practices: A Systematic Review and Survey.

Koskela, J., & Abrahamsson, P. (2004). On-site customer in an XP project: empirical results from a case study *Software Process Improvement* (pp. 1-11): Springer.

Kruchten, P. (2013). Contextualizing agile software development. *Journal of Software: Evolution and Process, 25*(4), 351-361.

Kumar Srivastava, D., Singh Chauhan, D., & Singh, R. (2011). Square Model A Proposed Software Process Model for BPO based Software Applications. *International Journal of Computer Applications, 13*(7), 33-36.

Kuppuswami, S., Vivekanandan, K., Ramaswamy, P., & Rodrigues, P. (2003). The effects of individual XP practices on software development effort. *ACM SIGSOFT Software Engineering Notes, 28*(6), 6-6.

Lankshear, C., & Knobel, M. (2004). *A handbook for teacher research*: McGraw-Hill Education (UK).

Larman, C. (2004). *Agile and iterative development: a manager's guide*: Addison-Wesley Professional.

Layman, L., Williams, L., & Cunningham, L. (2004). *Exploring extreme programming in context: an industrial case study.* Paper presented at the Agile Development Conference, 2004.

Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012). *Software development life cycle AGILE vs traditional approaches.* Paper presented at the International Conference on Information and Network Technology.

Lee, N. G. F. R. M. (1991). Using computers in qualitative research: Sage.

Leffingwell, D. (2010). *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*: Addison-Wesley Professional.

Lemos, O. A. L., Ferrari, F. C., Silveira, F. F., & Garcia, A. (2012). *Development of auxiliary functions: Should you be agile? an empirical assessment of pair programming and test-first programming.* Paper presented at the Proceedings of the 34th International Conference on Software Engineering.

Lewins, A., Taylor, C., & Gibbs, G. (2005). What is qualitative data analysis (QDA). *Online QDA. Online: onlineqda. hud. ac. uk/Intro_QDA/what_ is_qda. php.{Accessed 19 July 2008}.*

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry* (Vol. 75): Sage.

Lindstrom, L., & Jeffries, R. (2004). Extreme programming and agile software development methodologies. *Information systems management, 21*(3), 41-52.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., . . . Kähkönen, T. (2004). Agile software development in large organizations. *Computer, 37*(12), 26-34.

Lippert, M., Becker-Pechau, P., Breitling, H., Roock, S., Schmolitzky, A., Wolf, H., & Heinz, Z. (2003). Developing complex projects using XP with extensions. *Computer*(6), 67-73

Macholz, C. W. (2007). *XP Project Management.* Master of Science, The University of Montana, United States.

Mannaro, K., Melis, M., & Marchesi, M. (2004). Empirical analysis on the satisfaction of it employees comparing xp practices with other software development methodologies *Extreme Programming and Agile Processes in Software Engineering* (pp. 166-174): Springer.

Marchesi, M. (2005). *Extreme programming and Agile processes in software engineering*: Springer.

Marrington, A., Hogan, J. M., & Thomas, R. (2005). *Quality assurance in a student-based agile software engineering process.* Paper presented at the Software Engineering Conference, 2005. Proceedings. 2005 Australian.

Marshall, C., & Rossman, G. B. (1999). Designing qualitative research Thousand Oakes: CA: Sage Publications.

Martin, R. C. (2003). *Agile software development: principles, patterns, and practices*: Prentice Hall PTR.

Maurer, F., & Martel, S. (2002). Extreme programming: Rapid development for Web-based applications. *IEEE Internet computing*(1), 86-90.

Maxwell, J. A. (2013). *Qualitative research design: An interactive approach* (Vol. 41): Sage.

McBurney, D., & White, T. Research methods/-Belmont (Calif.): Thomson/Wadsworth, 2007.–441 p: ISBN 0-495-09208-8.

McConnell, S. (2004). *Code complete*: Pearson Education.

McMillan, J. H., & Wergin, J. F. (1998). *Understanding and Evaluating Educational Research*: ERIC.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*: Sage.

Miller, J. H., & Page, S. E. (2009). *Complex adaptive systems: an introduction to computational models of social life: an introduction to computational models of social life*: Princeton university press.

Mills, A. J. (2010). *Encyclopedia of case study research* (Vol. 1): Sage.

Mingers, J. (2001). Combining IS research methods: towards a pluralist methodology. Information systems research, 12(3), 240-259.

Mohamed, P., Farvin, S., Baharom, F., & Deraman, A. (2014). An Exploratory Study on Agile based Software Development Practices. *International Journal of Security & Its Applications, 8*(5).

Mohammed, H., & Rauf, A. (2015). Agile Project Management: Brief Review. *Lecture Notes on Software Engineering, 3*(3), 225.

Munassar, N. M. A., & Govardhan, A. (2010). A comparison between five models of software engineering. *IJCSI, 5*, 95-101.

Mushtaq, Z., & Qureshi, M. R. J. (2012). Novel Hybrid Model: Integrating Scrum and XP. *International Journal of Information Technology and Computer Science (IJITCS), 4*(6), 39.

Nawrocki, J., Jasiński, M., Walter, B., & Wojciechowski, A. (2002). *Extreme programming modified: embrace requirements engineering practices.* Paper presented at the Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on.

Omar, M., & Abdullah, S. L. S. (2015). The Impact of Agile Methodology on Software Team's Work-Related Well-Being. *International Journal of Software Engineering & Its Applications, 9*(3).

Omar, M., Abdullah, S., & Lailee, S. (2013). Agile practices: A cognitive learning perspective.

Omar, M., Syed-Abdullah, S.-L., & Yasin, A. (2010). Adopting Agile Approach: A Case in Malaysia.

Omar, M., Syed-Abdullah, S.-L., & Yasin, A. (2011). The impact of agile approach on software engineering teams. *American Journal of Economics and Business Administration, 3*(1), 12.

Päivärinta, T., & Smolander, K. (2015). Theorizing about software development practices. *Science of Computer Programming, 101*, 124-135.

Paulk, M. (2001). Extreme Programming from a CMM Perspective. *IEEE Software, 18*(6), 19-26.

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.

Pickard, A. (2012). Research methods in information: Facet publishing.

Pickering, C. (2001). Building an Effective E-project Team. *E-Project Management Advisory Service, Cutter Consortium, 2*(1).

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

Powell, R. R. (1997). Basic research methods for librarians: Greenwood Publishing Group.

Pressman, R. (2009). *Software Engineering: A Practitioner's Approach*. (7th ed.). New York, USA: McGraw-Hill Education.

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

Pressman, R. S., & David Brian, L. (2009). *Web engineering:: a practitioner's approach.*

Puvenesvary, M., Rahim, R. A., Naidu, R. S., Badzis, M., Nayan, N. F. M., & Aziz, N. H. A. (2008). *Qualitative Research: Data Collection & Data Analysis*: UUM press.

Qureshi, M. (2011). Empirical Evaluation of the Proposed eXSCRUM Model: Results of a Case Study. *International Journal of Computer Science Issues (IJCSI), 8*(3). 150-157.

Ragin, C. C. (1987). *The comparative method: Moving beyond qualitative and quantitative strategies*: Univ of California Pr.

Rejab, M. M., Omar, M., Mohd, M., & Ahmed, K. B. (2011). *Pair programming in inducing Knowledge sharing.* Paper presented at the Proceedings of the 3rd international conference on computing and informatics.

Rittenbruch, M., McEwan, G., Ward, N., Mansfield, T., & Bartenstein, D. (2002). *Extreme participation-moving extreme programming towards participatory design.* Paper presented at the PDC2002 Proceedings.

Roulston, K. (2010). *Reflective interviewing: A guide to theory and practice*: Sage.

Rumpe, B., & Schröder, A. (2014). Quantitative survey on extreme programming projects. *arXiv preprint arXiv:1409.6599*.

Saldaña, J. (2012). *The coding manual for qualitative researchers*: Sage.

Salo, O., & Abrahamsson, P. (2008). Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. *Software, IET, 2*(1), 58-64.

Santos, R. P. d. (2014). *ReuseSEEM: an approach to support the definition, modeling, and analysis of software ecosystems.* Paper presented at the Companion Proceedings of the 36th International Conference on Software Engineering.

Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2005). *Object-oriented Analysis and Design: With the Unified Process*: Thomson Course Technology.

Saunders, M., & Lewis, P. (2012). *Doing research in business and management: An essential guide to planning your project*: Financial Times Prentice Hall.

Schwaber, K., & Beedle, M. (2002). gilè Software Development with Scrum.

Sekaran, U., & Bougie, R. (2009). Research methods of business: A skill-building approach (ed.). New York: John Willey & Sons: Inc.

Sekaran, U., & Bougie, R. (2010). Research methods for business: A skill building approach. Wiley: London.

Senapathi, M., & Srinivasan, A. (2012). Understanding post-adoptive agile usage: An exploratory cross-case analysis. *Journal of Systems and Software, 85*(6), 1255-1268.

Sfetsos, P., & Stamelos, I. (2007). Improving Quality by Exploiting Human Dynamics in Agile Methods. *Agile Software Development Quality Assurance,* 154.

Sfetsos, P., Angelis, L., & Stamelos, I. (2006). Investigating the extreme programming system–An empirical study. *Empirical Software Engineering, 11*(2), 269-301.

Shore, J., & Warden, S. (2008). The Art of Agile Development O'Reilly Media Inc: Shroff Publishers and Distributors Pvt. Ltd.

Siebra, C., Mozart Filho, S., Silva, F. Q., & Santos, A. L. (2008). *Deciphering extreme programming practices for innovation process management.* Paper presented at the Management of Innovation and Technology, 2008. ICMIT 2008. 4th IEEE International Conference on.

Sillitti, A., Succi, G., & Vlasenko, J. (2012). *Understanding the impact of pair programming on developers attention: a case study on a large industrial experimentation.* Paper presented at the Proceedings of the 34th International Conference on Software Engineering.

Singhal, A., & Banati, H. (2014). FISA-XP: an agile-based integration of security activities with extreme programming. *ACM SIGSOFT Software Engineering Notes, 39*(3), 1-14.

Sinkovics, R. R., & Alfoldi, E. A. (2012). Progressive focusing and trustworthiness in qualitative research. *Management International Review, 52*(6), 817-845.

Sison, R., & Yang, T. (2007). *Use of Agile Methods and Practices in the Philippines.* Paper presented at the Software Engineering Conference, 2007. APSEC 2007. 14th Asia-Pacific.

Sison, R., Jarzabek, S., Hock, O. S., Rivepiboon, W., & Hai, N. N. (2006). *Software practices in five ASEAN countries: an exploratory study.* Paper presented at the Proceedings of the 28th international conference on Software engineering.

Skinner, M., & CIS, F. C. I. (2001). Enhancing an Open Source UML Editor by Context-Based Constraints for Components: University of Berlin, Thesis.

Sliger, M., & Broderick, S. (2008). *The software project manager's bridge to agility*: Addison-Wesley Professional.

Solinski, A., & Petersen, K. (2014). Prioritizing agile benefits and limitations in relation to practice usage. *Software Quality Journal*, 1-36.

Soundararajan, S., Arthur, J. D., & Balci, O. (2012). *A methodology for assessing agile software development methods*. Paper presented at the Agile Conference (AGILE), 2012.

Stamelos, I. G. (2007). *Agile software development quality assurance*: Igi Global.

Stellman, A., & Greene, J. (2014). *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*: " O'Reilly Media, Inc.".

Stober, T., & Hansmann, U. (2010). *Best Practices for Large Software Development Projects*: Springer.

Syed-Abdullah, S. L., Omar, M., Hamid, M. N. A., bt Ismail, C. L., & Jusoff, K. (2009). Positive affects inducer on software quality. *Computer and Information Science, 2*(3), p64.

Syed-Abdullah, S., Holcombe, M., & Gheorge, M. (2006). The impact of an agile methodology on the well being of development teams. *Empirical Software Engineering, 11*(1), 143-167.

Tan, S. (2011). How to increase your IT project success rate: Gartner.

Tessem, B. (2003). Experiences in learning xp practices: A qualitative study *Extreme Programming and Agile Processes in Software Engineering* (pp. 131-137): Springer.

Tian, Y. (2009). Adapting Extreme Programming For Global Software Development Project.

Tsvara, P. (2013). *The relationship between the management strategies of school principals and the job satisfaction levels of educators.* Doctoral dissertation, University Of South Africa, Pretoria.

Turk, D., France, R., & Rumpe, B. (2014). Assumptions underlying agile software development processes. *arXiv preprint arXiv:1409.6610*.

Turk, D., France, R., & Rumpe, B. (2014). Limitations of agile software processes. *arXiv preprint arXiv:1409.6600*.

Turk, D., France, R., Rumpe, B. (2002). Limitations of Agile Software Processes. *In Proceeding of the Third International Conference on extreme Programming and Agile Processes in Software Engineering held on 26-30 May 2002 at Alghero, Sardinia, Italy* (pp. 43-46). New York: ACM.

Unterkalmsteiner, M., Gorschek, T., Cheng, C. K., Permadi, R. B., & Feldt, R. (2012). Evaluation and measurement of software process improvement—a systematic literature review. *Software Engineering, IEEE Transactions on, 38*(2), 398-424.

Valacich, J., George, J., & Hoffer, J. (2009). *Essentials of system analysis and design*: Prentice Hall Press.

Vanhanen, J., & Korpi, H. (2007). *Experiences of using pair programming in an agile project.* Paper presented at the System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on.

Wells, D. (2009). Agile process. extreme programming: a gentle introduction.

Williams, L., Krebs, W., Layman, L., Antón, A., & Abrahamsson, P. (2004). Toward a framework for evaluating extreme programming. *Empirical Assessment in Software Eng.(EASE)*, 11-20.

Williams, L., Layman, L., & Krebs, W. (2004). Extreme programming evaluation framework for object-oriented languages. *Computer Science TR-2004-18*.

Willig, C., & Stainton-Rogers, W. (2007). *The SAGE handbook of qualitative research in psychology*: Sage.

Willig, C., & Stainton-Rogers, W. (2008). Qualitative research in psychology: Los Angeles & London: SAGE Publications.

Wood, S., Michaelides, G., & Thomson, C. (2013). Successful extreme programming: Fidelity to the methodology or good teamworking? *Information and Software Technology, 55*(4), 660-672.

Wood, W., & Kleb, W. L. (2003). Exploring XP for scientific research. *Software, IEEE, 20*(3), 30-36.

Wu, B. H. (2011). *On software engineering and software methodologies a software developer's perspective.* Paper presented at the Information Science and Technology (ICIST), 2011 International Conference on.

Xu, B. (2009). *Towards high quality software development with extreme programming methodology: practices from real software projects.* Paper presented at the Management and Service Science, 2009. MASS'09. International Conference on.

Xu, Y., Lin, Z., & Foster, W. (2003). Agile Methodology in CMM Framework: an Approach to Success for Software Companies in China. Proceedings of the GITM.

Yin, R. K. (2011). *Applications of case study research*: Sage.

Yousef Al-tarawneh, M., Syazwan Abdullah, M., & Bashah Mat Ali, A. (2012). Comparison of Extreme Programming (XP) method and key process areas of CMMI-DEV1. 2. *Global Journal on Technology, 1.*

Zikmund, W. (2003). *Business Research Methods the Dryden Press*: Harcourt College Publishers: Fort Worth.

Zuiderveld, N. R. (2003). *eXtreme Programming and SCRUM: A Comparative Analysis of Agile Methods.* Paper presented at the Published in the Proceedings of the International Conference on Software Engineering, Portland.