AN EXPLORATORY STUDY ON CURRENT SOFTWARE DEVELOPMENT PRACTICES IN MALAYSIA FOCUSING ON AGILE BASED SOFTWARE DEVELOPMENT

[1]Shafinah Farvin Packeer Mohamed, [2]Fauziah Baharom
(Universiti Utara Malaysia, Sintok, Malaysia
[1]shafinah, [2]fauziah {@uum.edu.my})
Aziz Deraman
(Universiti Malaysia Terengganu, Kuala Terengganu, Malaysia
a.d@umt.edu.my)

ABSTRACT: *Agile based software development has gained much interest among practitioners and researchers due its ability to produce high quality software in a shorter time of period. Even though its importance has been revealed, nevertheless, only few studies were conducted regarding its current practices in Malaysian software industry. Thus, a survey was conducted to study the practices and perception on the agile based software development in real-world projects. This paper discusses on the findings from the study. Structured questionnaire was used for data collection purpose. For data analysis, simple statistical methods were used which are frequency and cross tabulation. Outcome from this study reveals that agile based software development practices are important to produce high quality software. Thus, it prolonged to the needs of incorporating agile based software development practices as the reference standard in the process based software certification model which will be proposed.*

Key words: Agile based software development practices, software quality

1. INTRODUCTION

Quality has become vital for survival and competitiveness in all industries, including software industry. However, the success rate of software projects is reported to be decreasing, from 35% in year 2006 to 32% in year 2009. The failure rate is 24% whereby they are considered failed and had to be cancelled prior to completion and were never used. The rest of 44% were reported as challenged, which means they are either delivered late, over budget and/or delivered with less functionality than what was initially agreed upon [1]. Malaysia also encounters the same situation, whereby the software practitioners are facing problems in delivering good quality software, on time and within budget. This is because there is still lack in good software development practices among them [2]. Therefore, the software development practices need to be given proper attention in order to produce high quality software, since software development practices have influence on the quality of produced software [3].

In today's business environment, software developers need to fulfill few characterisitics of producing high quality software which are: 1) faster time to market, 2) lower development cost and 3) ability to move and change quickly. Consequently, nowadays software developers need to incorporate agility during software development process to fulfill these needs [4,5]. Despite of the importance of incorporating agility during software development, only few studies related to current industrial practice of agile based software development practices have been conducted in Southeast Asia region, particularly Malaysia. Most of the other studies were conducted in Western countries [6].

Therefore, an exploratory study has been conducted to investigate the Malaysian software practitioners' practices and perception on the software development practices concerning on agile based software development. Next section provides overview of agile based software development, continued with related works. Then the methodology used in this study is presented followed by discussions on the outcomes of the study. This paper is ended with the conclusion.

2. Agile based Software Development

Agile based software development process is *a lightweight software development approach which emphasizes on iterative, incremental, self-organizing and emergent practices* [7]. Agile is introduced recently as a consequence from the problems faced in conventional methodologies which are not flexible in accepting unstable and volatile requirements. It is aimed to produce higher quality software in a shorter time. Currently there are many agile methods such as Extreme Programming (XP), Scrum and Agile Modeling [8]. These methods have similar values and practices, whereby they follow 12 principles, for instance: "Welcome changing requirements, even late in development" and "Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale" [9].

3. Existing Studies on Agile based Software Development

Only few studies related to agile based software development practices in the industry have been conducted in Southeast Asia region, particularly Malaysia, while most of the other studies were conducted in Western countries [6]. Version One started surveying the state of agile adoption since 2006 [13]. It is aimed for getting insight on the status of agile development adoption and practices. In 2011, the survey was participated by 6042 respondents all over the world. The outcome from the survey indicated that more than 80% of the respondents are practicing agile and the most used agile methods were Scrum and Scrum/XP hybrid. Forrester Research conducted a similar survey participated by 1298 respondents [14]. Among the most used agile practices are short iterations (79%), constant feedback (77%) and daily scrum meeting (71%), while the least used are test driven development (42%) and metaphor (15%).

Salo & Abrahamsson investigated the usefulness of Extreme Programming and Scrum in European embedded software development organization [15]. They found out that open office workspace (66%), coding standards (60%) and 40-hour week (59%) to be the most used agile practices. Furthermore, TDD and pair programming were found to be the least practiced.

Santos, Bermejo, Oliveira and Tonelli studied the perception of software practitioners on the relationship of agile practices with the quality of software [16]. The finding from this study shows that bigger involvement of the staff, agile management of the requirements proposed and code developed can lead to high quality of software.

Sison & Yang conducted two case studies among software practitioners to explore the implementation of agile practices in Philippines [17]. One organization implemented only five XP practices while the rest practices were neglected. Another organization implemented Scrum and they seemed to appreciate the Scrum's sprint, abd the daily Scrum meetings. They highlighted that Scrum has improved their productivity.

Ani Liza, Gravell and Wills investigated the issues and problems faced by the early adopters when implementing agile methods [18]. They investigated the factors that are important in adapting agile. The researchers concluded that social and human factors are important and technical factor is less important when using agile methods.

Omar, Abdullah and Yasin studied the impact of agile approach among the software engineering teams in a computer center in Malaysia [19]. This study concluded that effective methodology and organizational culture are important factors that must be considered to produce innovative teams and quality software.

The existing studies only focused on the practices of a particular agile method, such as Scrum or XP. Thus, this study takes into consideration the practices of XP, Scrum and AM together in order to cover the overall process of software development, including the management and documentation.

4. Research Approach

The study was conducted through survey by using structured questionnaire due its cost effectiveness, ease of analysis, wide area coverage and integrity assurance [20]. There were four main activities as depicted in Figure 1.
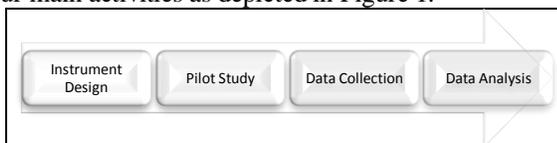
Fig 1: Main activities in the study

4.1 Instrument Design

The instrument was constructed by referring to the previous works such as [13,16,21,22]. It consisted of 29 questions with sub questions, organized in two main sections: demographic and agile based software development practices. In general, 5-Level Likert Scale was used for most of the questions. Additionally, multiple responses questions and yes/no questions were also included. Example of questions is included in the Appendix.

4.2 Pilot Study

Prior to the real survey, a pilot study has been conducted to confirm the validity and readability of the questionnaire. Pilot test is very important in investigating the wellness and feasibility of the questionnaire [23]. Ten respondents were chosen to answer and give feedback about the instrument. They were system analysts and programmers who have at least 3 years' experience. The respondents gave some suggestions to improve the instrument, including simplifying the questions to be more readable and understandable, reducing the number of questions and reorganizing the presentation of questions. Consequently, the instrument was refined based on their feedback.

4.3 Data Collection

The researcher contacted all of the potential respondents through telephone and asked their willingness to participate in the survey. However, only 32 of software practitioners responded. Accordingly, appointments were made with them to conduct face-to-face data collection session. This approach was used in order to ensure that the respondents clearly understand each question and answer them properly. Furthermore, they can get immediate clarification if they do not understand the questions. The respondents were from Kuala Lumpur, Penang Kedah and Selangor, as these are the places where software development companies most located in Malaysia (Ani Liza et al., 2012). Moreover, these states have the big software technology parks and International software development organizations.

4.4 Data Analysis

Data obtained from the survey was analyzed using simple statistical analysis, which are frequency and cross tabulation. The SPSS software was used for this purpose.

5.   THE RESULTS AND DISCUSSIONS

This section discusses the results obtained from the study. They are presented based on the items in the questionnaire and the objectives of the study.

5.1 Demographic Data

Most of the respondents were programmers (47%) and system analyst (47%), while only 3% was a project manager and 3% was a quality assurance/tester/auditor. They worked either in service and public administration (6%), education/training (25%), software development (53%) or health and social work (16%) sector. Figure 2 depicts the percentages. From these various organization sectors, majority of the respondents were from private organizations (59%).
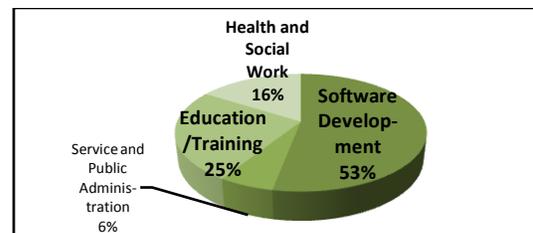

Fig 2: Respondents' organization sector

In term of experience, out of the 32 respondents, only 13% have experience more than 10 years. Most of respondents

have 1 to 5 years experience (59%) and among them, 38% were programmers. Table 1 shows the analysis result.

Table 1 Respondents' Experience

Position	<1 year	1-5 years	6-10 years	11-20 years	Total
System Analyst	1 (3%)	6 (19%)	6 (19%)	2 (6%)	15 (47%)
Program-mer	1 (3%)	12 (38%)	1 (3.1%)	1 (3.1%)	15 (47%)
Project Manager	0 (0%)	0 (0%)	0 (0%)	1 (3%)	1 (3%)
Quality Assuranc e/ Tester	0 (0%)	1 (3%)	0 (0%)	0 (0%)	1 (3%)
Total	2 (6%)	19 (59%)	7 (22%)	4 (13%)	32 (100%)

5.2 Software Practitioners' Experience and Perceptions on Agile based Software Development

This section presents the results based on the stated objectives.

Objective 1: To study Malaysian software practitioners' perception on the importance of including agile based software development practices in producing high quality software.

Based on the literature, agile based software development approach is important and should be included in current software process as it ensures that high quality software could be marketed faster in most cost-effective [24]. The respondents were asked about this issue. All of the respondents who has experience in agile agreed that it has influence on the quality of produced software (100%). They also agreed that incorporating agility during software development can enhance the ability to manage changing requirements (50%), enhance software quality (47%), accelerates time-to-market (47%) and increases productivity (41%). This highlights that agile based software development is vital for today's business environment and must be included in the proposed process based software certification model.

Objective 2: To study Malaysian software practitioners' experience with agile based software development

Referring to previous study by [18], agile based software development in Malaysia is still in early stage and some of the respondents do not even have heard of it. However, this study found out that more than half of the respondents have experience in agile (62%), either being a member of agile team previously (22%), currently being a member of agile team (34%), currently being agile coach (3%) or leading an agile team (3%). This shows that this approach is gradually being practiced nowadays in Malaysia. However, this study also indicates the same outcome of [18], whereby there still

exist among the respondents who have never heard about agile based software development (13%). This shows that this important practice of software development is still being neglected by the Malaysian software practitioners, even though their importance has been clearly realized.

The rest of the questions were then answered only by the respondents who have experience in agile based software development. This is to ensure the validity of the collected data. Most of the respondents were familiar with Extreme Programming (XP) (53%), followed by Scrum (28%), as depicted in Figure 3. This result is aligned with the findings from the study conducted by [13,15], whereby XP and Scrum were the most used agile methods.
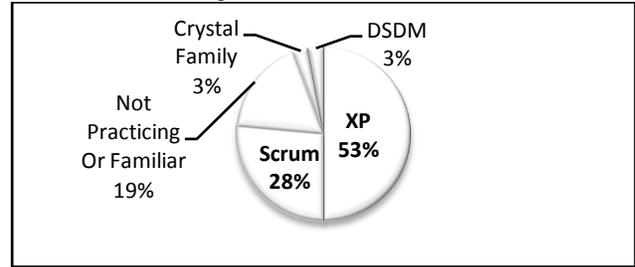

Fig 3: Agile methods

Objective 3: To investigate the Malaysian software practitioners' practices and perception on the agile based software development practices

Furthermore, the respondents were asked about the software development practices needed in order to produce high quality software, concerning on agility. The practices are categorized into requirement engineering, design, coding, testing, project management and change management. The practices were derived from agile methods such as Extreme Programming, Scrum and Agile Modeling, as well as concerning on the agile principles and values [9]. These methods were chosen since XP and Scrum complements each other, whereby Scrum focuses on project management, while XP focuses on project development [25]. Additionally, Agile Modeling also referred, as it provides a methodology for effective modeling and documentation for agile [11].

Outcome from the study shows that mostly these practices obtained high consideration among the respondents. This shows that they are important practices in producing high quality software. Mode value for each practice is used in the analysis. The scale used is the 5-point Likert Scale whereby 1 represents *Unimportant*, 2 represents *Of Little Importance*, 3 represents *Moderately Important*, 4 represents *Important* and 5 represents *Very Important*.

● **Requirement engineering**

Unlike the conventional software development approach which emphasizes on completed and well-defined requirements up-front, requirement engineering in agile is performed iteratively and incrementally [26,27,28,29]. Additionally, it emphasizes on face-to-face communication during requirement elicitation, enables continuous requirement prioritization, with minimal requirements documentation. Moreover, frequent review meetings are important in order to verify the requirements and showing

the progress to customers by using prototyping. The results from this study regarding the agile requirement engineering practices are aligned with previous studies by [26,28,30]. In order to ensure the consistency and traceability of requirements, the use of product and iteration backlog has been agreed by the respondents as important. This finding is same with previous study by [15].

- **Design**

Agile based software development approach emphasizes on simple initial design which continuously evolve by time. This practice is one of the agile based software development's success factors concluded by [31] in their study. Furthermore, refactoring is another important agile design practice. It is a valuable tool that can be used to improve the design of software [33]. This is agreed by the respondents in this study. Moreover, metaphor is used as architecture of the system by the respondents in [34]. However, this practice gained low percentage in this study, same as in [32].

- **Coding**

Similar to requirement engineering and designing, coding in agile is implemented iteratively and incrementally. Moreover, it emphasizes on collective code ownership. This practice gained high consideration among the respondents in this study. Also, previous studies by Salo and Abrahamsson, Williams et al. and Rumpe and Schroder [15,30,32] reported the same. In addition, before starting the coding, all programmers will agree upon a set of coding standard that everybody will follow when writing the source code. Studies by [13,15,17,30,31,32,34] indicated that coding standard is a highly adopted practice among their respondents. This is same with this study's outcome.

Pair programming is one of the most accepted and succeeded in the industry and academic. One of the most significant results is the design and code quality improvement [35,36,37]. Studies by [30,32,36] found that pair programming is among the top agile practices. On the contrary, in [13,15], this practice was rated as the least practiced.

Test driven development (TDD) is a critical practice in order to produce high quality software [35]. The developers create the unit tests before writing the production code. Many studies has proven its ability to produce high quality software, such as [38]. Meanwhile, [41] stated that the complexity of code and design were reduced with this practice. Study by [30] indicates that TDD is an important practice, while studies by [15,34] reported in contradict. However, this practice gained high consideration in this study.

By practicing refactoring on code and database, the software will be easier to understand, helps in finding bugs, and helps program faster. It is concerned with restructuring the internal code across existing classes, without changing its external behavior [33]. Moser pointed out that refactoring increases the quality of software as well as improves the productivity [42]. On the other hand, [43] found out that refactoring does

not influence the quality of produced software. However, in this study, this practice gained high consideration.

Continuous integration of source code and database to the system baseline has been found as an important practice in [13,14,30,32,34,15], same as the outcome of this study. By performing this practice, compatibility problems can be detected or avoided earlier [29]. Next, is the practice of delivering software frequently with increments of features. By doing so, the software can be demonstrated earlier to customers and enables them to review the software, identify defects and adjust for future [39]. These practice has been considered important by the respondents in this study, as well as in studies by [17,32].

Delivering features with high priority first is one of the agile principles. It can ensure that the most business value can be delivered first. It gained high consideration among respondents in this study. Moreover, study by [12] concluded that it is one of the factors that influence the success of agile implementation. Furthermore, defining code integration strategy and revising it is vital [27]. This practice also gained high consideration among the respondents. In addition, having customer on-site facilitates in providing continuous and immediate feedback [39]. [14] found that it is an important practice in their study, in the same way gained by this study.

- **Testing**

In contrast to the conventional software development approach which conducts testing after the implementation stage, testing in agile based software development is done continuously throughout the development, whereby it involves the unit tests, system integration tests, user interface tests and database regression tests, as well as user acceptance tests. However, the user acceptance tests are written by the customers to assure that the systems fulfill their needs. In some cases when customers lack of technical knowledge, then the developers will help the customers in writing the acceptance tests. The acceptance tests acts as a mechanism to validate and verify user's requirements. In addition, agile also emphasizes on automating these tests. These practices gained high consideration by the respondents in this study, as well as studies by [13,26].

- **Project management**

Project management in agile is different than the conventional software development approach. It consists of three levels of planning, which are release plan, iteration plan and daily plan. These planning are done iteratively and collaboratively, rather than planning the whole project up-front [11]. Additionally, the planning is done according to feature. A study conducted by [4] indicates that conducting these planning leads to better estimation.

Furthermore, the agile project management emphasizes on the sprint review and sprint retrospectives which are held at the end of a sprint to look back what worked well and what need to be improved [2]. These practices gained high consideration in this study with the studies by [13,26,32] However, in [15], the collaborative planning is rated as low consideration. Besides, the progress of the team should be

revealed in an open space so that everyone is aware of the current progress of the project. This practice gained high consideration in this study, as well as [13,14] Also, the working hours should not exceed 40 hours in a week to ensure productivity. They gained high consideration among the respondents in this study, as well as studies by [15,17,32].

• **Change management**

Agile involves with a lot of frequent changes. Thus change management and traceability is imperative [8]. Furthermore, a particular individual who will be responsible in managing the changes must be identified [27]. To enable the change management activities to be more efficient, the change management activities are automated. In order to avoid scope crepe, the changes are controlled by monitoring the product backlog and by restricting changes once the iteration starts. These practices were rated as important by the respondents of this study. The analysis results are presented subsequently in Table 2. On the whole, the agile based software development practices obtained high consideration from the respondents, thus they will be included in the proposed model as the reference standard.

Table 2 Agile based Software Development Practices

	Practices	Mode	Freq/ Percent
Requirement Engineering	. Gather requirements iteratively and incrementally	5	17 (53%)
	. Emphasize on face-to-face communication	4	15 (47%)
	. Only document important information	4	21 (6%)
	. Produce product and iteration backlogs	4	13 (41%)
	. Use prototype for validating requirements	4	14 (44%)
	. Conduct frequent review meetings	4	13 (41%)
	. Enable customers to prioritize and reprioritize requirements	4	13 (41%)
	. Enable development team to reestimate the execution time and velocity	4	14 (44%)
	. Identify the scope of project from beginning	5	20 (63%)
	0. Emphasize on single source information	4	17 (53%)
	.Produce executable specification	4	17 (53%)
Design	. Start design with simple initial design and integrate it continuously	4	18 (56%)
	. Refactor (reorganize) the design	4	16 (50%)
	. Use metaphor as architecture of the system	3	17 (53%)

	Practices	Mode	Freq/ Percent
Coding	. Implement model storming	4	17 (53%)
	. Create an initial model at the beginning of iteration	4	19 (59%)
	. Collective code ownership	4	16 (50%)
	. Follow coding standards	5	19 (59%)
	. Implement pair programming	3	12 (38%)
	. Implement test driven development	4	15 (47%)
	. Implement code refactoring	4	14 (44%)
	. Implement database refactoring	4	16 (50%)
	. Integrate database continuously	4	19 (59%)
	. Deliver the software frequently	4	16 (50%)
	0. Determine code integration strategy and revise it	4	23 (72%)
	. Get customers' continuous and immediate feedback	5	14 (44%)
	. Produce deliverable documentation late	4	16 (50%)
	. Deliver high prioritized features first	5	19 (59%)
Testing	. Implement automated tests	3	13 (41%)
	. Implement tests continuously throughout the development	4	15 (47%)
	. Implement frequent integration testing	4	13 (41%)
	. Acceptance tests are written or at least modeled by customers	5	12 (38%)
	. Use acceptance tests to validate and verify user's requirements	4	14 (44%)
	. Implement user interface test	5	22 (69%)
	. Implement database regression test	5	17 (53%)
Project Management	. Perform project planning continuously and collaboratively with team	5	15 (47%)
	. Carry out release meeting	4	16 (50%)
	. Carry out iteration meeting	4	15 (47%)
	. Carry out daily stand-up meetings	4	13 (41%)

	Practices	Mode	Freq/ Percent
	.Estimate the cost and schedule based on features/stories	5	13 (41%)
	.Conduct review meeting at end of each iteration	5	13 (41%)
	.Conduct retrospective at end of each iteration	4	14 (44%)
	.Monitor customer and end-user involvement	4	15 (47%)
	.Ensure that the weekly working hours do not exceed 40 hours	4	16 (50%)
	0. Reveal the current progress of iteration	4	17 (53%)
Change Management	1. Control changes using product backlog	4	13 (41%)
	2. Not allowing changes once an iteration has begin, until the iteration ends	4	12 (38%)
	3. Identify the responsible individual for change management activities	4	14 (44%)
	4. Automate the change management activities	4	16 (50%)

6. CONCLUSION

As a whole, this study has achieved all of the stated objectives. Findings from the study can be concluded as:

- All of the respondents agreed that agility should be considered during software development in order to produce high quality software. They highlighted that incorporating agility during software development can enhance the ability to manage changing requirements, enhance software quality, accelerates time-to-market and increases productivity. Thus, it prolonged to our future research to include agile based software development as reference standard in the proposed process based certification model.
- Software practitioners in Malaysia are gradually implementing agile based software development. However, there still exist among them who have never heard about it. The most implemented agile methods are XP and Scrum.
- The respondents also gave high consideration to the agile based software development practices as included in the questionnaire. This indicates that they are important practices in producing high quality software. Basically these practices are aligned with the agile principles and values in Agile Manifesto [9]. They emphasize on individuals and interactions, working software rather than documentation, encourage customer collaboration and adapting to changes.

For our next step, the Quality Functional Deployment [40] approach will be adapted in constructing the reference standard. The assessment criteria will be determined (the WHATs) and each of them will be mapped with the HOWs of achieving it. Mainly the HOWs will be attained from the practices which gained high consideration from software practitioners in this study. Interested readers may refer to [10] which discusses the background of our research.

ACKNOWLEDGEMENT

REFERENCES

[1] Chaos Report 2009, available at: http://www1.standishgroup.com/newsroom/chaos_2009.php.
[2] Blankenship, J., Bussa, M. and Millett, S. *Managing agile projects with Scrum*, *Pro Agile. NET Development with Scrum,* 2011.
[3] Deming, W. Out of Crisis, Cambridge, MA, MIT Center for Advanced Engineering Study, 1982.
[4] Tessem, B. *Experiences in learning XP practices: A qualitative study*, Extreme Programming and Agile Processes in Software Engineering, 2003.
[5] Boehm, B. Making a difference in the software century, *Computer*, **41**, 32-38(2008).
[6] Sison, R., Jarzabek, S., Hock, O. S., Rivepiboon, W. and Hai, N. N. *Software practices in five ASEAN countries: an exploratory study*, *International Conference on Software Engineering*, 2006.
[7] Boehm, B. and Turner, R. Management Challenges to Implement Agile Processes, *Traditional Development Organizations,* (2005).
[8] Jyothi, V.E. and K.N. Rao. Effective implementation of agile practices, *International Journal of Advanced Computer Science and Applications*, 41-48(2011).
[9] Principles behind Agile Manifesto, available at: http://agilemanifesto.org/principles.html.
[10] Baharom, F., Yahya, J., Deraman, A.and Hamdan, A.R. SPQF: software process quality factor for software process assessment and certification, *International Conference on Electrical Engineering and Informatics*, 2011.
[11] Agile project planning tips, available at: http://www.ambysoft.com/.
[12] França, A.C.C., da Silva, F.Q.B. and de Sousa Mariz, L.M.R. *An empirical study on the relationship between the use of agile practices and the success of Scrum projects*, *Proceedings* of ESEM, 2010.
[13] State of agile survey, available at: http://www.versionone.com/pdf/,_State_of_Agile_Development_Survey_Results.pdf 2011.
[14] West, D. and Grant, T. *Agile development: mainstream adoption has changed agility,* Forrester Research, 2010.
[15] Salo, O. and Abrahamsson, P. *Agile methods in European embedded software development organizations: a survey on the actual use and usefulness of Extreme Programming and Scrum*, in Software, 2008.,
[16] Santos, M. D. A., Bermejo, P. H. D., Oliveira and M. S. D. Tonelli, A.O. Agile practices: an assessment of perception of value of professionals on the quality criteria in performance of projects, *Journal of Software Engineering and Applications,* 700-799(2011),.

[17] Sison, R. and Yang, T. *Use of Agile Methods and Practices in the Philippines*, Software Engineering Conference, 2007.

[18] Asnawi, A. L., Gravell, A. M. and Wills, G. B. Empirical investigation on agile methods usage: issues identified from early adopters in Malaysia, *Agile Processes in Software Engineering and Extreme Programming*, 2011.

[19] Omar, M., Abdullah, S. L. and Yasin, A. The impact of agile approach on software engineering teams, *American Journal of Economics and Business Administration*, 12-17(2011).

[20] Questionnaires in usability engineering, available at: www.ucc.ie/hfrg/resources/qfaq1.html.

[21] Corbucci, H., Goldman, A., Katayama, E., Kon, F., Melo, C. and Santos, V. Genesis and Evolution of the Agile Movement in Brazil:Perspective from Academia and Industry, *Brazilian Symposium on Software Engineering*, 2011.

[22] Misra, S., Kumar, V. and Kumar, U. Identifying Some Important Success Factors In Adopting Agile Software, *The Journal of Systems and Software*, 1869-1890(2009).

[23] Teijlingen, E. and Hundley, V. *The importance of pilot studies*, Social research update, 2001.

[24] Pressman, R. S. Software engineering a practitioner's approach, 7th Ed. England, McGraw Hill, 2010.

[25] Fernandes J. M. and Almeida, M. Classification and comparison of agile methods, *Seventh International Conference on the Quality of Information and Communications Technology*, 2010.

[26] Liu, J., Wang, Q., Gao, L. Application of agile requirement engineering in modest-sized information systems development, *2nd WRI World Congress on Software Engineering*, 2010.

[27] CMMI Product Team, CMMI for Development V1.3 Technical Report, SEI: Pittsburg, USA, 2010.

[28] Lan, C. and Ramesh, B. *Agile requirements engineering practices: an empirical study*, *IEEE Software*, 2008.

[29] Extreme Programming, available at: http://www.extremeprogramming.org.

[30] Williams, L. Rubin, K. and Cohn, M. *Driving Process Improvement via Comparative Agility Assessmen*t, Agile 2010, 2010..

[31] Chow, T. and Cao, D. A Survey Study Of Critical Success Factors In Agile Software Projects, *The Journal of Systems and Software*, 961-971(2008),.

[32] Rumpe, B. and Schröder, A.Quantitative survey on extreme programming projects, *Third International Conference on Extreme Programming and Flexible Processes in Software Engineering*, 2002.

[33] Fowler, M. Refactoring Improving the Design of Existing Code, Addison-Wesley, 1999.

[34] Begel, A. and Nagappan, N. *Usage and perceptions of agile software development in an industrial context: An exploratory study*, ESEM '07, 2007.

[35] Sfetsos, P., Stamelos, I., Angelis, L. & Deligiannis, I. An experimental investigation of personality types impact on pair effectiveness in pair programming, *Empirical Software Engineering*, 187—226(2009).

[36] Begel, A. and Nagappan, N. Pair programming: what's in it for me? *Proceedings of the Second ACM-IEEE international symposium* on Empirical software engineering and measurement, 2008.

[37] Canfora, G., Cimitile, A., Garcia, F., Piattini, M., Visaggio, C. A. Evaluating performances of pair designing in industry, *Journal of Systems and Software*, **80**, (2007), 1317-1327.

[38] Huang, L., Holcombe, M. Empirical investigation towards the effectiveness of test first programming, *Information and Software Technology*, **51**, 182--194(2009),.

[39] Abrantes, J. F., Travassos, G. H. Common agile practices in software processes, *International Symposium on Empirical Software Engineering and Measurement*, 355-358(2011),

[40] Zultner, R. E. Quality Function Deployment (QFD) for Software. American Programmer, 1992.

[41] Sanchez, J. C., Williams, L., Maximilien, E. M. *On the sustained use of a test-driven development practice at IBM*, The AGILE 2007, 2007.

[42] Moser, R., Abrahamsson, P., Pedrycz, W., Sillitti, A., Succi, G. A case study on the impact of refactoring on quality and productivity in an agile team, *Balancing Agility and Formalism in Software Engineering*, 252-266(2008).

[43] Alshayeb, M. Empirical investigation of refactoring effect on software quality, *Information and software technology*, **51**, 1319-1326(2009).

APPENDIX

Please indicate the importance of these practices in producing high quality software.

1	2	3	4	5
Unimportant	Of Little Importance	Moderately Important	Important	Very Important

Requirement Engineering Practices	Importance Level
Gathering requirements iteratively	
Emphasizing on face-to-face communication	
Only documenting important information (core issues and functions)	
Discussing the detailed requirements at each development cycle's start	
Ensuring the consistency and traceability of requirements through mechanisms such as story cards, product backlog (prioritized user stories) or mockups and also review meetings and retrospectives	