Agile Methodology for Different Project Size

Dr.Osamah Rabab'ah , Maeed Said Alammary
Department Information Technology Of King Abdul Aziz University, Jeddah, Saudi Arabia

Abstract:- Extreme programming (XP) is one of the most widely used agile methodologies for software development. It intends to improve software quality and responsiveness to changing customer requirements. Despite the facts that the use of XP offers a number of benefits and it has been a widely used agile methodology, XP does not offer the same benefits when it comes to medium and large software projects. Some of the reasons for this are weak documentation, lack of strong architecture and ignorance to risk awareness during the software development.

1. INTRODUCTION

Extreme Programming is one of several agile software development methods that have emerged in the past few years. XP was first introduced in [9]. The method focuses on delivering immediate business value to the customer. The XP process can be characterized by short development cycles, incremental planning, evolutionary design, and its ability to response to changing business needs. The method itself is built around what appears to be easy-to-understand set of practices, which have been fairly well documented in the literature (see references for details). These practices are planning game, small releases, metaphor, simple design, testing (test-driven development), refactoring, pair programming, collective ownership, continuous integration, 40-hour work week (also known as sustainable pace) and on-site customer, just rules and open workspace. In addition, spikes [10] is also often associated to the XP method's practices. The XP method is designed to meet the needs of a skilled small, i.e. less than 10 developers, team that is working in a co-located office together with the customer developing software that is not safety-critical on an object-oriented technology [11]. This type of situation is what can be called an ideal surrounding for the XP method or what Boehm [12] calls an agile home ground.

2. LITERATURE REVIEW

In [1] shows that most of the companies that employ XP as a development methodology for medium and large projects face this problem, which echoes the importance of this problem. To address this problem, in this study XP model is extended in such a way that it equally offers its benefits for medium- and large-scale projects. As an evaluation of the extended XP, three independent industrial case studies are conducted. The case studies are described and results are presented in the study. The results provide evidence that the extended XP can be beneficial for medium and large software development projects. The extended XP model is validated using three case studies. The results are concluded based on the postmortem analysis and number of defects reported of the three case studies. The time (in hours) and effort (in %), spent per release to conduct postmortem analysis of the three case studies.

In [2] an agile deployment framework is proposed. It is compatible with the ideology of continuous improvement of organizational practices (QIP), while it also integrates it with the opportunities provided by short iterations of agile process model. The suggested framework includes the procedures and methods needed for selecting suitable new agile practices in an organization. It also embodies the means for iteratively tailoring and validating the deployed practices within agile projects and gaining feedback rapidly from projects to the organization. The paper presents the empirical experiences of a case study where the F-Secure Corporation deployed a new agile software development process (Mobile-D) in a pilot project in order to utilize its experiences in developing an organization specific agile process model alongside their traditional F-Secure product realization process. In this paper, the mpirical results from a case study are presented in order to illustrate how an agile development method (Mobile-D) was deployed in a pilot project in F-Secure Corporation. The organizational goal was to utilize the experiences from the pilot project in establishing organizational agile process. The pilot project applied a post-iteration workshop method [6] (i.e. PIWs) for iterative adaptation and improvement of agile practices. Some more traditional mechanisms were also used for collecting the experience based feedback from the project for the needs of the organization (i.e., project postmortems).

In [3] Previously available studies have mostly focused on comparing XP(extreme Programming) with some other Agile methodologies, rather than comparing it with traditional plan-driven software development methodologies. In this Paper, we identify the XP phases and practices, how they ensure product quality, and

map XP phases against the Spiral model phases to prove that XP has built-in QA (Quality Assurance) practices in its life cycle, in addition to its focus on productivity. A case study is also included to empirically investigate quality of the product developed using XP with comparison to the product developed using Spiral Model.

As a summary we can say the primary purpose of this paper is two-fold. First, it identifies and describes the built-in QA practices in both XP and Spiral Model. Secondly, it provides a case study to investigate the relation of quality in both processes. Quality is chosen because this is one of the major issues for which XP is supposed to answer properly.

As a result of this research, there are two main contributions of this research: it performs a detailed study to show that QA practices are built into an XP process like they are addressed by Spiral model. It provides empirical results related to Quality Assurance activities in XP. It can be concluded that an agile process like XP addresses quality issues repeatedly and continuously as compared to a traditional plan-driven software development method.

In this paper [4] They discuss their experiences in introducing agile practices in student software development projects and reflect on both the benefits and drawbacks of agile processes in this setting. the result shows that one of main conclusions were that the 12 inter-related practices of extreme Programming have limited value for educating about large-scale system development, but selected practices of XP may be helpful for educating about small scale development. For this work, they have, therefore, broadened their perspective and observed the application of a number of agile practices in the context of student software development projects in order to give recommendations on aspects to consider when introducing these practices in an educational setting.

In this paper [5] they present the results of an empirical study, in which we examined the continued use of an agile methodology in a later assimilation stage. Particularly, we investigated factors that foster or impede the acceptance of agile methodologies in the long-term. Using the Diffusion of Innovations Theory as a lens for analysis, we address the following research questions: which factors of agile methodologies are perceived as benefits or drawbacks by developers? How does this influence the commitment of developers to agile development?

The result of this research that is a little research has examined the use of agile methodologies beyond the adoption stage. This is especially true for studies which examine the perceptions of developers in Scrum projects. To close this literature gap, we presented results of a study in which we investigated how developers perceive Scrum in the long term. Taking the Diffusion of Innovations Theory as a lens for analysis, we identified several acceptance factors of Scrum and hypothesized about their perception by the developers. Using qualitative field data, we evaluated our research hypotheses and provided explanations for the perceptions that we observed in practice.

In this paper [6] clarify the effectiveness of agile methods, they reviewed the agile development literature and conducted a systematic study of what we know empirically about its benefits and limitations. their review clearly shows the need for more and better research to determine the situations in which practitioner advice on agile development can be suitably applied. We urge companies to participate in research projects that target goals relevant for the software industry.

In this paper [7] They've identified some barriers to integrating agile and traditional methods as perceptual, not technical. However, they have also seen that many technical barriers do exist. they don't believe that these are insurmountable and are confident that organizations can overcome them with diligence, patience, and work. Clearly, agile concepts will continue to migrate into traditional organizations (and vice versa) through planned or clandestine vectors. Research is needed in several areas to provide new approaches and to harmonize the methods. For example, the Systems and Software Consortium is finalizing a document called Disciplined Agility, which describes an approach to implementing agile methods in high-maturity and process-compliant environments Most importantly, though, is the need to capture metrics data and lessons learned from your experiences. Data is critical to validate the integration activity's value and the return on investment. Lessons learned can support more rapid integration, eliminate the repetition of ineffective approaches and practices, and disseminate experience across the organization and throughout the community.

In this paper [13] Waterfall and Extreme Programming are two software project methods used for project management. Although there are a number of opinions comparing the two methods regarding how they should be applied, none have used project data to clearly conclude which one is better .Authors present the results of a controlled empirical study conducted at Carnegie Mellon University in Silicon Valley to learn about the effective transition from traditional development to agile development. They conducted a comparison research against these two approaches. Multiple teams were assigned a project; some used Waterfall development, others used Extreme Programming. The purpose of this research is to look at advantages and disadvantages based upon the outcomes, generated artifacts, and metrics produced by the teams.

we observed and presented the data from five years of 50 teams developing the same project each year and the affects of transitioning from Waterfall to Extreme Programming. The characteristics between these two

methods were evaluated and compared. Waterfall teams spent more time creating high ceremony documents where as Extreme Programming teams spent more time writing code and documenting their design in their code. Surprisingly, the amount of code and features completed were roughly the same for both methods suggesting that on a three month project with three to four developers it doesn't matter the method used.

Table Of Limitations

Title of paper	Limitations
[1] Agile software development methodology for medium and large projects	The XP module used for only small project
[2] Deploying Agile Practices in Organizations: A Case Study	The methods needed for selecting suitable new agile .
[3] Software Quality Assurance in XP and Spiral - A Comparative Study	it is not comparing Xp with traditional plan-driven software
[4] Agile Practices in Software Development – Experiences from Student Projects	selected practices of XP may be helpful for educating about small scale development
[5] Investigating the Long-Term Acceptance of Agile Methodologies: An Empirical Study of Developer Perceptions in Scrum Projects	it is a little research has examined the use of agile methodologies beyond the adoption stage
[6] What Do We Know about Agile Software Development	the need for more and better research to determine the situations in which practitioner advice on agile development can be suitably applied

1. Problem definition:

There is a lot of question about agile methodology as tool to develop projects, especially when this project is described as large project. So there is need to valid agile software development methodology (for example extended extreme programming) is good method not only for small projects but also it can be good for medium and large projects.

2. Proposal Solution:
1) Using extend XP agile methodology under agile methodology.
2) Implement on medium and large project using XP practice .

3. Validation of the Proposed Solution:

Validation of proposed solution is conducting a survey through the questioner. Our main conclusion is that XP is used for small project and is not suitable for medium and large projects while there is other methods by agile methodology can be used for this purpose as extended XP.

4. Finding:

From this study we find that most of people who answer this questionnaire are consider XP as beneficial for develop projects. And most of them they know it is used to small projects. But the disadvantage is they say extended XP is not clear.

a. **Likert Scale**

b. **Validation Using Survey**
    Work a brief online questionnaire obtained through the opinions of a group of people, and that we have identified targets in advance .The result was based on the objectives as :
A. Cumulative Statistical Analysis of Goal1.
Using extend XP agile methodology under agile methodology.

5	Strongly Agreed
4	Agreed
3	Neither Agreed Nor Disagree
2	Disagree
1	Strongly Disagree
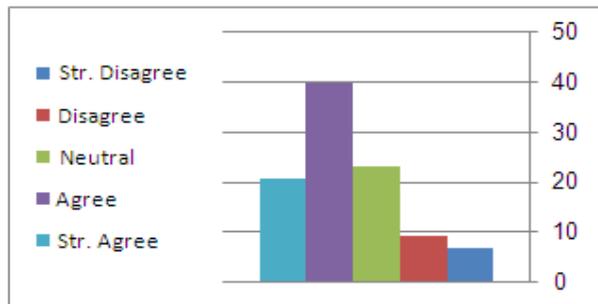
Q.No	Str. Disagree	Disagree	Str. Disagree	Neutral	Agree	Str. Agree
1	1	5	1	8	11	5
2	-	4	-	7	14	5
3	2	2	2	5	14	7
4	4	2	4	7	11	6
5	3	1	3	8	10	8
Total	10	14	10	35	60	31
Avg.	6.667	9.33	6.667	23.33	40	20.667

B. Cumulative Statistical Analysis of Goal 2
Implement on medium and large project using XP practice .

Q.No	Str. Disagree	Disagree	Neutral	Agree	Str. Agree
1	3	6	10	7	4
2	5	4	9	8	4
3	8	7	5	6	4
4	1	5	12	7	5
Total	17	22	36	28	17
Avg.	14.1667	18.333	30	23.333	14.1667
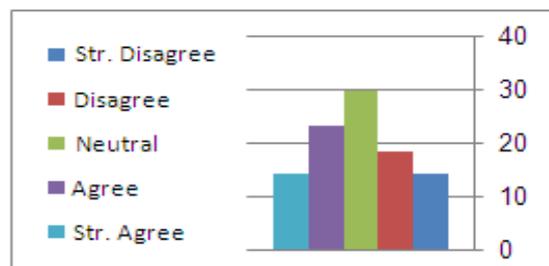
CONCLUSION

The main conclusion is that XP is used for small project and is not suitable for medium and large projects while there is other methods by agile methodology can be used for this large and medium projects as extended XP which is appear by questionnaire is not clear enough. Also from questionnaire we can conclude the importance of quality for developing of projects.
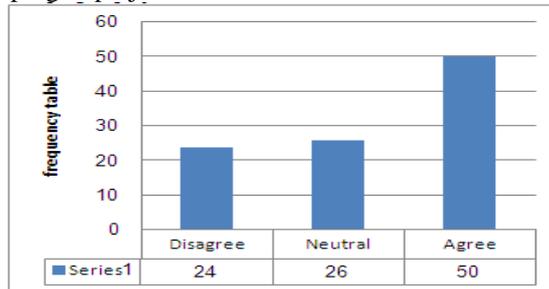
REFERENCE

[1]. Agile software development methodology for medium and large projects.
[2]. Deploying Agile Practices in Organizations: A Case Study
[3]. Software Quality Assurance in XP and Spiral - A Comparative Study
[4]. Agile Practices in Software Development – Experiences from Student Projects
[5]. Investigating the Long-Term Acceptance of Agile Methodologies: An Empirical Study of Developer Perceptions in Scrum Projects
[6]. What Do We Know about Agile Software Development?
[7]. Management challenges to implementing agile processes in traditional development organizations
[8]. Usage and Perceptions of Agile Software Development in an Industrial Context
[9]. K. Beck, "Embracing change with extreme programming," IEEE Computer, pp. 70-77, 1999.
[10]. R. Jeffries, A. Anderson, and C. Hendrickson, Extreme Programming Installed. Upper Saddle River, NJ: AddisonWesley, 2001.
[11]. K. Beck, Extreme programming explained: Embrace change. Reading, MA.: Addison Wesley Longman, Inc., 2000.
[12]. B. Boehm, "Get Ready For The Agile Methods, With Care," Computer, vol. 35, pp. 64-69, 2002.
[13]. Comparing Extreme Programming and Waterfall Project Results