Improvement opportunity in Agile Methodology and a survey on the adoption rate of the Improved Methodology

Mohammad Naderuzzaman (Corresponding author)
Department of Computer Science and Engineering Dhaka University of Engineering and Technology (DUET)
Gazipur, Dhaka, Bangladesh
nader_u@yahoo.com
Dr. Mohammod Abul Kashem
Faculty of Computer Science and Engineering Dhaka University of Engineering and Technology (DUET)
Gazipur, Dhaka, Bangladesh
Drkashemllduet.ac.bd

Abstract

Agile method promotes an iterative process on software development. It is a lightweight process that employs short iterative cycles, actively involve users and developers to establish, prioritize, and verify requirements and rely on a team's tacit knowledge as opposed to documentation. In this paper we describe the existing agile methodologies and improve its different parameters so that software development industries can adopt it more easily. It describes the improvement of overall understanding of the constituent parts of agile systems development methodologies and some improvement of different parameters. In our proposed method we design a Tool of adoption matrix which will help software development industries for adoption decision solution of the Improved Agile Methodology. We have described the result of different software workshop where the adoption assessment will be used to assess the existing agile system and the improved agile system. The result from this tool will help software industries to apply the improved agile methodologies.

Keywords: Agile Methodology, Adoption decision, Adoption assess tool, Improvement of Agile.

I. Introduction

Agile method promotes well disciplined process encouraging frequent inspection and adaptation, a leadership philosophy encouraging teamwork, self-organizing team and self-accountability. Agile method is a set of engineering best practices allowing clients/customers for rapid delivery of high quality software. Keeping this entire thing in mind, it is still a troublesome work to adopt agile methods for the team/organization. However, in the study of agile process in particular, it is desirable that the metrics collection program is lightweight and inconspicuous to the team's daily activities [12]. It is also widely believed that systems development methodologies (SDMs) can help for improvement of the software development process. The latest batch of SDMs is most appropriate in dealing with volatile business requirements [11]. Agile software development is widely recognized as a mainstream software development methodology and agile-driven software projects are managing the disciplined manner and attain the impressive result. In the agile system testing gets top priority when the customer is an ongoing interaction with the development and procedures and time estimation are set, gathered, analyzed and also acted upon [13]. There has a rich and interesting issue between the relationship of the deployment of agile systems development and organizational culture as this are the richness of the concept of "organizational culture". Cultures is such a organizations are always contested, changing and emergent and meanings are always created, recreated, negotiated and struggled in organizations [14]. In this paper, it has been designed an adoption tool (depending on some agile critical factors) by which software industries can assess how confident it's team to adopt agile method. Here, in this paper it also improved the some core agile parameters and use the same tool on the same team. After that in the discussion point, it has been shown a list of changed parameters for the agile software development methods.

II. Background

The word "agile" by itself means that something which is flexible and responsive, so agile methods implies its "[ability] to survive in an atmosphere of constant change and emerge with success" (Anderson, 2004, p. xxviii). This "maneuverability" in software business is a characteristic that is more important than ever these days since "deploying software to the Web has intensified software competition further than before" and "staying in business involves not only getting software out and reducing defects but tracking continually moving user and marketplace demands" (Cockburn, 2002, p. xxii). The definition of agile software development has been contained in a form of "manifesto" in Feb/2001 by a software development methodologies group. [1]

Most Agile software development literature cites the use of application development projects, mostly implemented in object-oriented languages, it also applied successfully on embedded projects as both share common themes. Particularly, agile methodologies target toward problems involving change and uncertainty, and are adaptive rather than predictive [1]. Agile methodologies highlights collaboration and team interactions, it values people over process. As per it's manifesto, it can be apply to any team endeavor. It also commonly advocate a barely sufficient process [2], which suits well with hardware engineers [3].

Are agile approaches effective to introduce and sustain meaningful change in any software development organization? How agile approaches are leverages beyond the projects development work? What are the process involved in approaching agile into an organization and how a leader becomes more agile in adopting agile approaches? DTE Energy's Information Technology Services (ITS) organization continues to grip and extend the agile mindset with their culture [4]. In this paper we assert that agile principals and techniques linked with software development projects can be eagerly applied in other types of organizational work and in creating and sustaining an effective leadership culture.

Agile methods are a response to the inability of traditional methods to embrace change in a turbulent business environment requires the software to meet your needs quickly (Highsmith & Cockburn 2001, Kruchen 2001). The basic underlying principles of agile methodologies are:

- Individuals are more important than processes and tools;
- Working software is more important than comprehensive documentation;
- Customer collaboration is more important than contract negotiation;
- Responding to change is more important than following a plan (Abrahamsson et al. 2002).

These principles are referred to as the 'Agile Manifesto'. There is no single agile development methodology. Some examples of agile approaches and methodologies that share many of these core values include: Extreme Programming (XP); Crystal Methods; AGILE; Dynamic Systems Development Method (DSDM); Feature Driven Development (FDD); and Adaptive Software Development (ASD) (Highsmith 2001, Sutherland 2001).

Glass (2001) describes the debate between proponents of traditional development approaches and proponents of the newer agile approaches. However, neither approach is correct in all circumstances, and the 'best-fit' needs to be determined for a given circumstance. "There are no silver bullets in software development, and there probably never will be" (Jeffries 2001). Of all the agile methods, DSDM (DSDM 1998) comes closest to recommending an approach for determining whether a project is suitable for using an agile development method. DSDM recommends that the organization must have the right culture for using agile approaches, but it is not specific about how one should go about measuring or evaluating this culture. Furthermore, the steps in the feasibility study that follow this recommendation involve educating a key stakeholder, and producing a strategy and plan. Based on these recommendations, it appears that the choice has already been made to adopt DSDM, so it is not a decision whether to adopt an agile approach or not. The debate concerning agile methodologies has predominantly been based around whether it was a better choice than tradition development methods (De Marco & Boehm 2002), rather than a debate on the appropriateness of an agile approach for a given company, team, or project.

-Critical Factors for Adopting agile

Critical Factors	Description	Reference
Project Duration	The project Timeframe should be short for an agile project	Highsmith (2001 Kruchen (2001)
Customer Involvements	Customer involvement is vital for the success of a project	Beck and Fowler (2001) Young (2003)
Acceptance of change (to requirements)	Agile methods are specifically aimed at projects subject to continual change	Highsmith (2001)
Team Size	Agile methodologies emphasize the importance of teams, recommending smaller team sizes	Boehm (2002) Rising and Janoff (2000)
Skill level of team	Highly skilled developers are required	Reifer (2002) Levine et al. (2002) Lindvall et al. (2002)
Documentation	While agile methodologies do not prohibit documentation, it should be kept to a minimum	Greening (2001) Olson and Stimmel (2002)
Workshop view	Open planned offices, with shared areas, are required to promote communication and team work	Poole and Huisman (2001) Kalita (2003)
Tasks	Tasks are identified and each is estimated (1-16 hours)	Mike Cohn(2002)
Meeting	Daily AGILE meeting includes • Daily • 15-minutes • Stand-up	Mike Cohn(2002)
Sprint	Product is designed, coded, and tested during the sprint	Mike Cohn(2002)

Table 1: Critical Adoption Factors for an Agile Methodology

One assumption can be made that, by further research this critical factors can be change dynamically. These critical factors have been chosen from the point of team's satisfaction. As the team is the most important part for software development, TEAM should always come first.

Problem Statement: Many industries profoundly negative experiences with agile. Many other developers and contractors think that AGILE is a scam.

Here's some issues with AGILE:

1. Allowing customers to "change mind" means that the software development teams are expected to incur the cost of unlimited wants of consumers. If customers cannot express their needs, it is probably cannot distinguish between a "need" and a "want", but expect to pay the bill in the same way. Combined unlimited needs with limited resources are the classic problem of economy.
2. Some developers want to work in the area surrounded by cubes and glass, some developers want to listen to songs at work, some developers can take some rest in their work. But agile supports open space which is may turn into developer's dissatisfaction.
3. Daily meeting (SCRUM: one of AGILE methodology) is a vital point. But maybe it is not always good for daily scrum. Daily scrum is a kind of micro-management. No always everybody willing to attend on daily scrum. Sometimes daily scrum become boring because not always everybody listen to everybody, as example somebody may work on UI design work and other is working on DB operations. So these 2 developers may not be interested to listen on other scrum. Rather sometimes developers feel that SCRUM meeting is a kind of cross-examination.

Methodology: This paper will show some improvements of core critical factors. This may bring lots of arguments but further research can improve the existing improvements on the critical factors. This paper is proposing the following changed critical factors below in Table 2:

Critical Factors	**Points**
Acceptance of change (to requirements)	Only the minor tweaks or cosmetic changes can be acceptable, Requirement should not change the core functionalities and Business Logic.
Workshop view	Workstation should be cubical or personalized and private.
Tasks	Every task must be only 3 hrs or 6 hrs of length
Meeting	Daily meeting may not mandatory always. Daily meeting can be team Lead or the whole team's choice.

Table 2: Improvement on the Critical Adoption Factors of the Agile Methodology

Now this paper will show a Tool to assess the adoption rate of the Improved Agile (IA) and the Current Agile (CA). To develop this tool this paper considered the improved critical factors. In Table 3, this paper shown the Matrix tool to adoption assess between IA and CA.

After creating this Tool, this paper will show the data collecting from a software Industry in UK. After collecting the data, a simple algorithm will be used to find out the adoption rate of Improved Agile (IA) and Current Agile (CA).

Critical Factors	**Points**	**Weight**	**Factor ID**
Acceptance of change (to requirements)	1 = Only minor tweaks are accepted 2 = Continuous changes are accepted	1	1 = A1 2 = A2
Workshop view	1 = Private Workstation 2 = Open, shared and public view	1	1 = W1 2 = W2
Tasks	1 = Tasks must be 3hrs or 6hrs 2 = Tasks can be 1 hr to 16 hrs of length	1	1 = T1 2 = T2
Meeting	1 = Not mandatory, depends on team leads decision 2 = Daily meeting is mandatory	1	1 = M1 2 = M2

Table 3: Matrix Tool to adoption assess between IA and CA.

These critical factors will be represented as very simple TRUE/FALSE questions to different software professionals. Collecting data will be applied on the algorithm below to find out if the improved agile (IA) is easier to adopt or the current agile (CA).

Result: Software professional's satisfaction is important to secure high quality software development. Keeping this thing in mind we contact software and web development company iBACS (http://www.ibacs.co.uk/) in UK which has an offshore development center in Bangladesh. We make a survey among all the developers, designers and QA team members. We describe our result below:

Profession	No of Professionals	Number of Questions	IA	CA
Manager	3	12	8	4
Architect	2	8	5	3
Senior Developers	20	80	70	10
Developers	30	120	115	5
Jr. Developers	40	160	153	7
QA programmer	20	80	72	8
Artist/Designer	20	80	78	2

Table 4: Survey report on Improved Agile and Current Agile

From the result, only Managers and Architects have higher rate of supporting non modified agile method, although the number of answer more supports on the modified agile. On the other hand, all the other professions highly support the modified agile (IA). Since developer's satisfaction is highly required it is better to modify any methodology for the company specific professionals.

Conclusion: This paper has described some problem found from the existing agile methodology and some improvement factor of this method. Based on the developer's satisfaction we improved some of the critical factors of existing agile method, we surveyed on a company and published our result. Further study and survey can improve other agile factors.

References

1. Tsun Chow, Dac-Buu Cao, "A survey study of critical success factors in agile software projects", The Journal of Systems and Software 81 (2008) 961–971
2. chen jianbin, shi tong, fang deying, "A HeavyWeight IT Project Management Framework based on Agile Theory" IEEE conference on Management and Service Science, 2009. MASS '09, on pp. 1 – 5
3. Bill Greene, "Agile Methods Applied to Embedded Firmware Development" IEEE conference on Agile Development Conference, 2004, on pp. 71 – 77
4. Steven W. Baker, Joseph C. Thomas, "Agile Principles as a Leadership Value System: How Agile Memes Survive and Thrive in a Corporate IT Culture" IEEE, AGILE 2007, on pp. 415 – 420
5. Lee, Keeo, "An agile method for web applications in dynamic requirements" IEEE conference on Networked Computing and Advanced Information Management, 2008. NCM '08, pp. 178 – 182
6. A. Qumer, B. Henderson-Sellers, "An evaluation of the degree of agility in six agile methods and its applicability for method engineering" Information and Software Technology 50 (2008) 280–295
7. Woi Hin, Kee, "Future Implementation and Integration of Agile Methods in SoftwareDevelopment and Testing" IEEE conferences on Innovations in Information Technology, 2006,pp. 1 – 5
8. Michael Coram and Shawn Bohner, "The Impact of Agile Methods on Software Project Management" IEEE conferences on Innovations in Information Technology, 2006, pp. 1 – 5
9. Dr. David F. Rico, Dr. Hasan H. Sayani, "Use of Agile Methods in Software Engineering Education" IEEE conference on Agile Conference, 2009. AGILE '09, pp. 174 – 179
10. Jayakanth Srinivasan, Kristina Lundqvist, "Using Agile Methods in Software Product Development: A Case Study1" IEEE conference on nformation Technology: New Generations, 2009. ITNG '09.pp. 1415 – 1420
11. Frank K.Y. Chan, James Y.L. Thong, "Acceptance of agile methodologies: A critical review and conceptual framework" , Decision Support Systems 46 (2009) 803–814
12. Lucas Layman a,, Laurie Williams a,1, Lynn Cunningham, "Motivations and measurements in an agile case study" , Journal of Systems Architecture 52 (2006) 654–667
13. Orit Hazzan , Uri Leron, "Disciplined and free-spirited: 'Time-out behaviour' at the Agile conference" The Journal of Systems and Software 83 (2010) 2363–2365
14. Juhani Iivari , Netta Iivari, "The relationship between organizational culture and the deployment of agile methods" , information and Software Technology. j.infsof.2010.10.008
15. Subhas Chandra Misra a,*, Vinod Kumar b, Uma Kumar, "Identifying some important success factors in adopting agile software development practices" , The Journal of Systems and Software 82 (2009) 1869–1890
16. Petri Kettunen, "Adopting key lessons from agile manufacturing to agile software product development—A comparative study", Technovation 29 (2009) 408–422
17. Michael Pearson , RonMasson, AnthonySwain, "Process control in an agile supply chain network" , Int. J. Production Economics 128 (2010) 22–30
18. Kieran Conboy, Lorraine Morgan, "Beyond the customer: Opening the agile systems development process" , Information and Software Technology . j.infsof.2010.10.007
19. Tsun Chow, Dac-Buu Cao, "A survey study of critical success factors in agile software projects", The Journal of Systems and Software 81 (2008) 961–971
20. Eric Germain, Pierre N. Robillard, "Engineering-based processes and agile methodologies for software development: a comparative case study" , The Journal of Systems and Software 75 (2005) 17–27
21. Kai Petersen , Claes Wohlin, "A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case" , The Journal of Systems and Software 82 (2009) 1479–1490

22. Eduardo Miranda a, Pierre Bourque, "Agile monitoring using the line of balance", The Journal of Systems and Software 83 (2010) 1205–1215
23. Ivana Turnu, Marco Melis, Alessandra Cau, Alessio Setzu, Giulio Concas *, Katiuscia Mannaro, "Modeling and simulation of open source development using an agile practice", Journal of Systems Architecture 52 (2006) 610–618
24. Helen Sharp , Hugh Robinson, Marian Petre, "The role of physical artefacts in agile software development: Two complementary perspectives" , Interacting with Computers 21 (2009) 108–116
25. Witold Pedrycz, "Quantitative logic-based framework for agile methodologies" Journal of Systems Architecture 52 (2006) 700–707
26. A. Qumer, B. Henderson-Sellers, "A framework to support the evaluation, adoption and improvement of agile methods in practice" The Journal of Systems and Software 81 (2008) 1899–1919