Agile Software Methodologies: Strength and Weakness

Dr. Adel Hamdan Mohammad,
Computer Science Department, The world Islamic Sciences and Education University, Amman-Jordan,
Faculty of Information Technology,
Adel.hamdan@wise.edu.jo
Adel_hamdan@yahoo.com

Dr. Tariq Alwada'n,
Computer Science Department, The world Islamic Sciences and Education University, Amman-Jordan,
Faculty of Information Technology,
Tariq.alwadan@wise.edu.jo

Dr. Jafar "M.Ali" Ababneh
Network and Computer Information System Department, The world Islamic Sciences and Education University, Amman-Jordan,
Faculty of Information Technology,
jafar.ababneh@wise.edu.jo

Abstract

Agile methodologies are great software development methodologies. No doubt that these methodologies have widespread reputation. The core of agile methodologies is people. Customer and each team member in agile development teams are the key success or failure factor in agile process. In this paper authors demonstrate strength and weakness points in agile methodologies. Also authors demonstrate how strength and weakness factors can affect the overall results of agile development process.

Keywords: Agile methodologies; Agile Strength; Agile Weakness.

1- Introduction

Software development is a complicated process to deliver products in more rapidly, enhanced and cheaper ways. There have been several studies and ideas on improving and enhancing the overall development process. To defeat the rapid change in organization and business need in traditional methods a new mythology emerges. This methodology is called agile method or agile methodology. Agile methods focus on people, customer's satisfaction and rapid response to change [1, 2, 3].

First, on February, 2001, the creators of agile methodologies are met to talk and prepared the manifesto of Agile Software Development (ASD). In the meeting seventeen people discussed and talked about the future of software development methodologies. They notice that they share some common characteristics in general. These characteristics are mainly focused on people. The result of the meeting was the creation of a new intelligent methodology named "Agile Alliance". [2, 3, 4, 5]

Old methodologies or traditional software methodologies such as System Development Life Cycle (SDLC) is a structure imposed on the development of a software product. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. Traditional software development methodologies has been quite famous and been used by many software development teams and enterprises across the world. SDLC includes several stages from preliminary development analysis to post development software testing and evaluation. It also consists of the models and methodologies that development teams use to create the software. These heavy aspects became to be known as heavyweight. Thus some researchers tend to use more flexible method than traditional methodologies. Theses methodologies and techniques are based on iterative development. As a result the era of agile methodologies start to emerge [2, 3, 4, 5].

Agile software process or agile software methodologies are a group of software development process that based on incremental development or iterative development. In agile development process requirements and solutions are changed and evolved through collaboration and communication between organization and teams. Agile software process encourages rapid and flexible response to change [2, 3, 4].

One of the main and core characteristics of agile methodologies is that it addresses the problems of rapid change and it is based on best practices and their previous success and failures stories. In addition agile methodology team must be more effective and efficient in responding to changes. Theses days, agile software methodologies or lightweight methodologies generate a huge debate between traditional methodologies followers and agile methodologies followers. But actually there are many evidences and success stories from working with agile methodologies propose that agile methods are flourishing and appropriate for many situations and circumstances. [2, 3, 6, 7, 8, 9, 10]

Taking in consideration the increasing change and ambiguity in most software systems would lead to search for a process or methodologies that can deal with such circumstances. No doubt that agile methodology has the capability to deal with such change. [6, 7, 8, 9, 10]

This paper is organized as follows. In the next section, we review relevant literature; section three demonstrates agile approach, section four is about agile methodologies strength and weakness. The last segment of this paper presents our conclusion.

2- Previous Study

F. Maurer [11] studied the development of a web based system by nine full time employees in a small company that used XP and observed substantial productivity gains compared to their pre-XP timeframe.

Marchesi and Succi [12] demonstrate that there is increasing evidence that Agile Methodologies (AMs) have the potential to improve significantly the way software is produced.

Cho [13] demonstrated that in contrast to traditional SDLC, the Agile SDLC avoids ‘up-front’ requirement gathering as stakeholders often could not provide all requirements in sufficient details for implementation to occur at the beginning of a project.

Vijayasarathy [14] in his research said that although agile methodologies are better than traditional methodologies in many aspects, several difficulties are still there when it comes to practice. One among these is that agile methods significantly reduce the amount of documentation, and even claim that the code itself should act as a document.

An analytical study done by Tan [15] found confirmation of staffing stability and design compatibility that influence the success of software projects that use iterative or incremental development. The basic idea behind iterative development is to develop a software system incrementally, allowing the developer to take advantage of what was learnt during the development of earlier deliverable versions of the system. Therefore, iterative and incremental software developments are viewed as the keystone of the agile methodologies.

Uikey [16] in his research demonstrate that the informal communication among stakeholders and developers sometimes raises problems such as inability to cope with system complexity and rapidly changing requirements, or inability to add new members to the development team.

Morkel [3] examined in his research the adaptability of agile methodologies to software development standard laid by ISO. Also he said that Extreme Programming XP is used as main representative of different agile methodologies.

Aladwani [4] made an attempt to authenticate and integrate performance model of information system IS. In this research authors demonstrate that information systems project performance depends on six categories of variables. Also he said that information system project performance must consider the social context within which IS projects operates.

Demarco [5] in his research talked about Extreme Programming XP and demonstrated how XP create more much interest since it offers some fundamental alternative to traditional software engineering principles.

Steve [6] in his article talked about ongoing shift in which software is made from inhouse development to be ready to install products. Also he said that installation consists of several detailed analysis issues. In the traditional systems development life cycle (SDLC), analysis issues handled in the first stages, as part of the requirements analysis stage. In the market model of SDLC, detailed analysis comes after purchase. It is only during installation that users become deeply involved in the first time in assessing how the software meets their needs.

Wolfgang [7] in his research demonstrated that the support of software quality in a software development process depends on two points: the first point is providing some techniques that support the development of high quality software. The second point is providing techniques that assure the important quality attributes.

James [8] in his research demonstrates that Extreme programming XP is one of the most discussed subjects in the software development community. Also in this research authors show that XP uses a set of practices to build a successful software team that produce quality software.

Juyun Cho in [17] his research said that agile has several benefits but organization refuse to jump to agile methodologies for several reasons: 1) agile methods significantly reduce the amount of documentation and rely heavily on tacit knowledge, 2) agile methods have not been sufficiently tested for mission/safety-critical projects, 3) agile methods are not adequate for highly stable projects, 4) agile methods can be successful only with talented individuals who favor many degrees of freedom, and 5) agile methods are not appropriate for large-scale projects.

3- Agile Approach

Traditional or conventional methods are still used in large-scale projects particularly in industry. The reason for using traditional methods comes from their straightforward philosophy [18]. And its ability to provide predictability, constancy, and assurance. [19].

Agile software development (ASD) is relatively a new term used in software engineering. Agile processes or agile methodologies actually represent a new approach for planning and managing software projects. In spite of ASD which represents new methodologies in software development it is based on some existed philosophies and theories which both are inspired from the field of software engineering [12, 14, 18, 19].

Agile development processes have been analyzed from several experts and researchers. According to some researchers ASD is mainly focusing on four aspects: [1]

1. Agile development is nothing new, such practices have been in place in software development since2 the 1960s [20].

2. The lack of focus on architecture is bound to engender suboptimal design-decisions [21].

3. The practices in XP are rarely applicable [22].

4. Agile development methods are suitable for small teams, but for larger projects, other processes become more appropriate [23].

Nowadays; agile methodologies is one of the most important rising methodologies in software engineering. No doubt, that there is a high speed changing in market. This change enforces producers of software to take care of several required and not required features. Also new emergent issues required that producers have to answer several questions such as how companies can differentiate themselves? What technology must be used? Which type of technology can survive? To answer these questions companies must have different visions for the future and flexible methodologies. Taking in consideration that from hardware and software perspectives making long term prediction is unachievable.

By saying that agile development methodologies are based on the reality of rapidly change of today’s market. And taking in consideration that agile philosophy can deal with emergent issues which can appear rapidly in market environment, tools, and technologies. All these benefits and encouraged features which are a result of agile methodologies introduced through four basic values: [8, 9, 24, 25, 26]

Individuals and interactions over processes and tools.

- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

Agile methodologies core is people. People are the main factor for success or failure. One of the main important principles in agile methodologies is that people can respond earlier and faster. Also fast response to emergent issues in agile methodologies enhance the overall process since it doesn’t require following procedures for writing documents and calling several level of managers.

In agile methodologies team members and management have an equal place. This enhances collaboration and restricts constraints. And this doesn’t mean in any way that technical staff will take the role of managers. [27, 28, 29].

Also another main goal of agile methodologies is its focusing on working software with minimum effort. Simplicity is very important. The idea here is to never produce and write code more than required. To success in agile methodologies customer collaboration is a key aspect. Also in agile methodologies we do our business according to customer needs. So customer plays a great role in earlier phases. And in reality the customer is one member of the development team. [29, 30, 31].

One of the main reasons behind using iterative development its allowing the developer to take benefit of what was learnt through the development and earlier deliverables versions of the system [32].

Agile software philosophy based iterative and incremental development, customer collaboration, and frequent delivery. Agile process is done through light and fast development process [33].

Finally responding to change is one of success keys. In traditional methodologies team members and managers are following a written plan. Nowadays there is a great and rapid change in market which makes the process of following a written plan is inapplicable in most situations. So we have to follow flexible and unstable methodologies. [33, 34, 35, 36].

4- Agile methodologies: Strength, Weakness

In the following section authors will talk about agile methodologies strength, opportunities and weaknesses.

Actually there are many encouraging benefits in agile philosophy such as short development cycle, high customer satisfaction, and quick adaptation to change. No doubt that the strength of agile methodologies is based on its flexibility.

One of the strength points of agile methodologies is that its philosophy which based on the idea that projects are developed in short iterations. At the end of each iteration the user can see a working version of the software before moving ahead to the next iteration which means that the overall project will be more flexible. In traditional methodologies if the customer changes some requirement after long period such as (six-ten month) this require complete project rebuilding, but through applying agile philosophy it can be adapted. Another strength point is that the customer can evaluate the development after each iteration and suggest plans for future. Another advantage of agile methodologies is the constant testing and integration since testing is done after each iteration which means that faults can be corrected regularly. Another strength point is that progress can be viewed and measured after each iteration.

Also another strength point of agile methodologies is that it based on people and their creativity. This means that solving emergent issues and problems will be done immediately without following rigid policy and procedures. Another positive point in agile methodologies is that developers are often more pleased since they communicate with customers and respond to their requirements without following rigid policies in communicating with managers to take an approval for every simple modification. Flexibility does not means that developers can do modifications and respond to customers' needs without telling their supervisors but they can respond to customers' modifications before telling their managers. Also one of the main characteristics of agile methodologies is that it based on working on small teams. This means that there is always a continued channel for communication between team members which increase job satisfaction. Continued communications means identifying problems and mistakes earlier. Besides that working in small teams will accelerate the development process.

In agile methodologies customers are playing an important role in the working team. Agile methodologies recommend having a full time customer working with development team. This means that any misunderstand of customers' requirements can be solved and answered immediately. Also this means the time taken for solving any emergent issues will be reduce to the minimum.

Now we will start talking about agile weakness. There are several experts and practitioners say that "it's agile-not agile". Actually a very big mistake that many people assume that agile mainly based on flexibility and lack of flexibility. In reality this is not true, since agile methodologies has its own formal rules and polices. Another point is customer interaction. Agile followers consider this point as a merit. But in fact it may become a weak point in some circumstances such as the user or customer might not find enough time to spend with developers, or if the key customer is one of the high level managers. Another weakness point that agile methodologies recommend that the agile development team and end users are located in the same physical location for long periods daily which is not applicable under all situations.

One on the main weakness in agile methodologies is difficulty in coordinating between teams for large projects. Consider two teams working in separate locations and they are hundred of kilometers apart. Is there coordination equivalence to two teams sitting in the same physical location?

Another point we can talk about that agile methodologies lack of long term planning. This means that when various components of the project have been assembled definitely this will result in some problems. In agile methodologies this is called "refactoring". Also since agile methodologies based on a verbal communication with customers this will lead to a weak documentation.

Beside that one of the key successes in agile methodologies based on having strong and high qualified developers having the creativity and skills to works in teams and to communicate with other teams or customers. This type of developers is not easy to find and demand a huge cost. Also working in small teams guide to troubles in identifying individuals' contributions and how they can be pleased. Beside that lack of enough documentation for tracking overall progress make executive managers face huge challenges of not delivered goals.

5- Conclusion

In this paper authors try to show and highlight the strength and weakness of agile methodologies. Agile methodologies are great methodologies which appears as a respond to frequent changes in market. This paper demonstrate main strength points of agile methodologies such as it's philosophy based on short development cycle, high customer satisfaction and a quick adaptation to change. And how agile methodologies respond to change in markets and many organizations find these methodologies more suitable for market. This paper refers to some weak points in agile methodologies. After studying the weakness and strength points of agile authors found that there is a strength point which may be converted to weakness under some situations such as customer interaction.

Acknowledgment

The authors are grateful to the World Islamic Sciences and Education University, Amman-Jordan, for the financial support granted to cover the publication fee of this research article.

References

[1] Kuda Nageswara Rao, A Study of the Agile Software Development Methods**,** Applicability and Implications in Industry, International Journal of Software Engineering and Its Applications Vol. 5 No. 2, April, 2011

[2] M. A. Awad , A Comparison between Agile and Traditional Software Development Methodologies , This report is submitted as partial fulfillment of the requirements for the Honours Programme of the School of Computer Science and software Engineering, The University of Western Australia, 2005

[3] W.H. Morkel Theunissen, Derrick G. Kourie and Bruce W. Watson, Standards and Agile Software Development, Proceedings of SAICSIT, Pages 178–188, 2003

[4] Adel M. Aladwani, Journal of Management Information Systems /, Vol. 19, No. 1, pp. 185–210, Summer 2002.

[5] Tom Demarco, XP is the most important movement in our field today", IEEE computing and control engineering, June/July 2003.

[6] Steve Sawyer, Information System Development, Communications of the ACM /Vol. 44, No. 11, page 97-102, November 2001.

[7] Wolfgang Zuser, Stefan Heil, Thomas Grechenig, Software Quality Development and Assurance in RUP, MSF and XP - A Comparative Study, St Louis, Missouri, USA. ACM, 2005

[8] James Newkirk, Introduction to agile processes and extreme programming, Orlando, Florida, ACM, 2005.

[9] The Manifesto for Agile Software Development; http://agilemanifesto.org/;

[10] Richard Turner, Barry Boehm, People Factors in Software Management: Lessons From Comparing Agile and Plan-Driven Methods, The Journal of Defense Software Engineering, 2003

[11] F. Maurer and S. Martel, "Extreme Programming: Rapid Development for Web-Based Applications", IEEE Internet Computing, 6(1), pp. 86-91, Jan/Feb 2002.

[12] Marchesi and Succi, 2002 Proceeding of the 3rd International Conference on eXtreme Programming and Flexible Processes in Software Engineering (XP2002)

[13] Cho, Juyun. Issues and Challenges of agile software development with SCRUM. Issues in Information System. VOL IX, No. 2. 2008.

[14] Vijayasarathy, Leo R. Agile Software Development: A survey of early adopters. Journal of Information Technology Management Volume XIX, Number 2. 2008.

[15] Tan, T., et al., Productivity trends in incremental and iterative software development, in Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement. 2009, IEEE Computer Society. p. 1-10.

[16] Uikey, N., U. Suman, and A.K. Ramani, A Documented Approach in Agile Software Development.International Journal of Software Engineering (IJSE), 2011. **2**(2): p. 13-22.

[17] Juyun Cho ,A Hybrid Software Development Method For Large-Scale Projects: Rational Unified Process With Scrum**,** Issues in Information Systems, Volume X, No. 2, 2009

[18] Boehm, B. & Turner, R. (2003) Using risk to balance agile and plan-driven methods. IEEE Computer, 36(6), 57-66.

[19] Fruhling, A. & Vreede, G. (2006). Field experiences with extreme programming: Developing an emergency response system. Journal of Management Information Systems, 22(4), 39-68.

[20] H. Merisalo-Rantanen, T. Tuure, R. Matti, Is extreme programming just old wine in new bottles: a comparison of two cases, Journal of Database Management 16 (4) (2005) 41–61.

[21] M. Stephens, D. Rosenberg, Extreme Programming Refactored: The Case Against XP, Apress, Berkeley, CA, 2003, ISBN 1-59059-096-1.

[22] G. Keefer, Extreme Programming Considered Harmful for Reliable Software Development 2.0, AVOCA GmbH, Online Report, 2003.

[23] S.Nerur, V. Balijepally, Theoretical reflections on agile development methodologies, Communications of the ACM 50 (3) (2007) 79–83.

[24] M. Fowler and J. Highsmith. 2001. The Agile Manifesto. Software Development Magazine. August. http://www.drdobbs.com/the-agile-manifesto/184414755

[25] eWorkshop. 2002. Summary of the First eWorkshop on Agile Methods. http://fcmd. umd.edu/projects/Agile/Summary/SummaryPF.htm. April.

[26] P. Abrahamson et al. 2002. Agile software development methods – review and analysis. VTT Publications 478. http://www.inf.vtt.fi/pdf/publications/2002/P478.pdf

[27] Paul E. McMahon, Are Management Basics Affected When Using Agile Methods?, The Journal of Defense Software Engineering, November 2006.

[28] B. Henderson-Sellers, M.K. Serour, Creating a Dual-Agility Method: The Value of Method Engineering, Journal of Database Management, vol. 16, Issue 4, 2005.

[29] Alistair Cockburn, Agile Software Development: The People Factor, Software Managemen t, Page 131-133, November 2001.

[30] Carmel, E. American hegemony in packaged software trade and the “culture of software.” Info. Soc. 13, 1 (Jan. 1997), 125–142.

[31] Jim Highsmith, Using Adaptive Software Development to meet the challenges of a high-speed, high-change environment, Software Testing & Quality Engineering (STQE) magazine**,** July/August 2000

[32] Larman, C. and V.R. Basili, Iterative and Incremental Development: A Brief History. Computer,2003. **36**(6): p. 47-56.

[33] Standish Group (1994). The chaos report. Retrieved March 6, 2008, from http://www.standishgroup.com/sample_research/chaos_1994_1.php.

[34] Rombach, D. “Fraunhofer: The German Model for Applied Research and Technology Transfer.” In Proc. ICSE’00 (Limerick, Ireland, Apr. 2000), 531-7.

[35] Forrest Shull, Jeffrey Carver, Guilherme H. Travassos, An Empirical Methodology for Introducing Software Processes, 1-58113-390-1/01/09, ACM 2001

[36] Nicolas Anquetil , Kthia M. de Oliveira, Kleiber D. de Sousa, Mrcio G. Batista Dias , Software maintenance seen as a knowledge management issue, Elsevier, 2006