

A Comparative Study of Agile Methods: Towards a New Model-based Method

Soukaina MERZOUK, Sakina ELHADI, Hassan ENNAJI, Abdelaziz MARZAK, Nawal SAEL
Hassan II University-Faculty of Sciences Ben M'sik, Laboratory of Information and Modeling
Morocco
merzouk.soukaina@gmail.com
elhadi.sakina1993@gmail.com
hassan.ennaji.17@gmail.com
marzak@hotmail.com
saelnawal@gmail.com



ABSTRACT: Agile development is an approach for managing and building software projects and teams, in order to satisfy customers by providing adaptive planning, evolutionary development, early delivery and continuous improvement. It is a collection of methodologies such as Extreme Programming (XP), Scrum and Feature Driven Development (FDD). Even if agile methods are rapid and flexible to change, they still suffer from some problems example, difficulty of estimating the effort required at the beginning of the development software life cycle, especially for large projects. Also, it is important to keep design phases sufficiently long to allow for a real maturation of needs and avoid reflection based only on reactions to what has been developed, which can lead to a lot of reworking. As a first step towards a new agile method, this paper aims to provide a comparative study of available agile methods in order to propose a new model-based method.

Keywords: Agile Software Development, Model-based Method, XP, Scrum, FDD

Received: 21 June 2017, Revised 28 July 2017, Accepted 5 August 2017

© 2017 DLINE. All Rights Reserved

1. Introduction

Agile development is a philosophy for managing software projects and teams, aiming at satisfying customers by providing adaptive planning, evolutionary development, early delivery and continuous improvement. It declines in several methodologies such as Extreme Programming (XP), Scrum and Feature Driven Development (FDD). Since 1960, the software industry faced many problems while using the traditional software development methods. Especially when the environment and the customer's requirements arrive to change, i.e. the customer does not have a clear idea about his project. However, traditional software development based on the famous sequence "*specification, design, realization and validation*", focuses more on decisions at the beginning of the project [1]. The project's success rates are low and the management processes must be adapted. In many traditional environments, the project manager is held accountable for the success or fail of the project, whatever the circumstances

if he is really responsible for success/failure or not. These changes have led since 1990 to a new philosophy called agile. Its main goal is to make business people and developers work together daily throughout the project to deliver operational releases to customer regularly. That enables all the people involved to check that it works fine [2]. Those methods decrease the failure probability significantly.

This paper seeks to provide comparative study of available agile methods. To achieve this goal, a set of criteria will be used to shed light on which aspects should be improved to propose a future successful model-based agile method. The remainder of the paper is organized as follows. Section 2 defines ‘Agile software development’, principles and values. Section 3 describes some of existing agile software development methods. Related work of several comparative studies is presented in Section 4. Section 5 identifies set comparison criteria, the goal of this choice, presents a comparative study of these methods and discussion. Finally, Section 6 concludes our research work.

2. Agile Software Development

Agile software development consists of methods to be compatible with constant and distributed environment [3]. In agile development, rather than long development cycles of conventional SDLC, the development cycle is split up into smaller segment, called “increments” or “iterations”, in which each of these increments touches on each of the conventional phases of development [4].

2.1 Values and Principles

The concept of agile development was written on Agile Manifesto in 2001 by 17 software practitioners. The focal values respected by the software practitioners are [5]:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

The 12 principles that have been written in this manifesto are [6]:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

2.2 Difference between Agile and Traditional Approaches

One major difference between agile development and conventional development methods is that the former methodology

possesses the ability to successfully deliver result quickly and inexpensively on complex projects with ill-defined requirements. Agile methods emphasize on teams, working software, customer collaboration, and responding to change; while the conventional methods stress on contracts, plans, processes, documents, and tools. Table 1 illustrates the differences between Agile and traditional approaches in several aspects [4].

Method Criteria	Agile	Traditional
User requirement	Iterative acquisition	Detailed user requirements are well defined before coding/implementation
Rework cost	Low	High
Development direction	Readily changeable	Fixed
Testing	On every iteration	After coding phase completed
Customer involvement	High	low
Extra quality required for developers	Interpersonal skills & basic business knowledge	Nothing in particular
Suitable Project scale	low to medium-scaled	Large-scaled

Table 1. Comparison between agile and traditional approaches [4]

On the other hand, some of the most common Agile methodologies (Agile methods) include [7]:

- Iterative strategies for managing software development projects, such as Scrum, the Dynamic Systems Development Method (DSDM), Feature Driven Development (FDD), the Agile Unified Process (AUP) and Lean Development.
- Strategies for optimizing software development work, such as extreme programming (XP) and the Rational Unified Process (RUP).
- Strategies for managing software maintenance and support activities, such as Kanban.

3. Various Agile Development Methods

Agile is an incremental and iterative approach of software development using smaller development cycles. Agile methodologies are adaptive rather than prescriptive. This research work includes eight agile methods, viz; XP, Scrum, DSDM, FDD, Crystal Family, LSD, Kanban and ASD.

3.1 Extreme Programming (XP)

Extreme Programming has evolved from the problems caused by the long development cycles of traditional development models. It breaks the process of the last-mentioned into smaller manageable pieces. That makes doing planning, analyzing and designing on each individual piece rather than the entire project. Consequently, reduce the cost of changing software [8]. Fundamental principles of XP are communication, simplicity, feedback, courage and respect [9], [2].

3.2 Scrum

Scrum approach has been developed for managing the systems development process. Scrum concentrates on how the team members should function in order to produce the system flexibly in constantly changing environment. The main idea of scrum is that system variables involves several technical and environmental variables (e.g. Requirements, time frame, and resources) that is likely to change during a process. A scrum is a set of guidelines that govern the development process of a product, from its design stage to its completion [10, 11].

3.3 System Development Method (DSDM)

The Dynamic System Development Method (DSDM) framework can be implemented for agile and traditional development processes. It is based on 9 principles such as Active User Involvement is Imperative, Teams Must be Empowered to Make Decisions, Focus on Frequent Delivery, Fitness for Business is Criterion for Accepted Deliverables, Iterative and Incremental Development is Mandatory, All Changes During Development Must Be Reversible, Requirements are Baselined at High-Level, Testing is Integrated Throughout the Lifecycle and Collaborative and Co-operative Approach [12]. DSDM consists of five phases: feasibility study, business study, functional model iteration, design and build iteration, and implementation [2].

3.4 Feature Driven Development (FDD)

Feature Driven Development (FDD) is an agile, highly adaptive software development process that is highly and short iterative, emphasizes quality at all steps, delivers frequent, tangible working results of all steps, provides accurate and meaningful progress and status information, with the minimum of overhead and disruption for the developers and Is liked by client, managers and developers [13].

FDD based on a set of “best practices” such as Domain Object Modeling, Developing by Feature, Individual Class (Code) Ownership, Feature Teams, Inspections, Regular Builds, Configuration Management and Progress reporting. The FDD approach focuses on the design and building phases [2], [14].

3.5 Crystal Family

Crystal is a family of methods because Cockburn believes that there is no “one-size-fits all” development process. The different methods are assigned colors arranged in ascending opacity, the most agile version is Crystal Clear, followed by Crystal Yellow, Crystal Orange, and Crystal Red. The need for distributed documentation, practices and tools is caused by the difficulty to manage the process via face-to-face communication when the team gets larger [15]. With Crystal Methods, organizations only create and use as large a methodology as their project and business needs demand [2], [16].

3.6 Lean Software Development (LSD)

Lean Software Development (LSD) is an iterative methodology that concentrates on optimizing the entire process to attain the maximum possible gain. It is more of a management philosophy than a development process, for that reason team size and iteration lengths are not specified. Lean Software Development offers seven Lean principles [17].

3.7 Kanban

The notion of “Kanban” comes from Japanese which means ‘signboard’. It is a method for managing the creation of products with an emphasis on continual delivery while not overburdening the development team. It is like Scrum, a process designed to help teams by working together more effectively. The six principles of Kanban are Visualized the workflow, Limit WIP (Work in Progress), Manage the workflow, Make processes/policies explicit, Implement feedback loops, Improve collaboratively [17].

3.8 ASD Method

Adaptive Software Development (ASD) is a methodology that was developed as a response to an economy that is increasingly changing and evolving [18]. It focuses mainly on the problems in developing complex, large systems. The method strongly encourages incremental, iterative development, with constant prototyping [2].

ASD is fundamentally a management philosophy for agile projects. An Adaptive Software Development project is realized in three-phase cycles which are Speculate (initiation and planning), Collaborate (concurrent feature development) and Learn (quality review). Lifecycle Characteristics of ASD are Mission focused, Time boxed, Risk driven, Iterative, Change tolerant and Feature based [17].

4. Related Work

As surveys of the previous comparatives studies of agile methods.

The work done in [19] is a comparison between five methods viz; XP, Scrum, FDD, DSDM and ASD. It comes to help future developers to choose the right method for their work.

The work presented by Mihai Liviu DESPA [11] makes a comparison of Software development methodologies including agile

methodologies by highlighting strengths and weaknesses. Its main objective is to formalize a software development methodology dedicated to innovation oriented IT projects. Moreover, the purpose of the comparative study done in [3], is to find out the similarities and difference between XP and Scrum, in order to know how to use XP and Scrum in a project.

This paper offers a solution for future developers, in order to help them in choosing the right methodology for their projects. Based on several criteria, such as, project, environment, team, etc. criteria.

In this paper, we aim to compare seven agile methodologies (extensible) in order to know their characteristics, advantages and disadvantages. Furthermore, to know if any of these methods is a model-based method or not. Also, we explore the possibility of proposing a general model of agile methodologies.

5. Comparison of Agile Methods

5.1 Criteria of Comparison

Agile development methodology is a conceptual framework for any software project. In general, agile approach provide a shorter development cycle, higher customer satisfaction, and quicker adaptation to rapidly changing requirements [17].

The choice of method for a project is very important, although there are as many methods based on the same concepts, but each with its own characteristics, processes, advantages and disadvantages, and they are not applicable in all the situations. Then, at the implementation level, each method proposes its own practices with a greater or less degree of involvement of the client. The project itself has characteristics (for example, the size of the project, its complexity, etc.) that can be used to choose the methodology to follow during all the phases of its realization. The success of a project depends on the correct choice and understanding of the method. So, for this study, we cited some criteria that gives us a general view on the method such as team size, project size, Iteration length, Roles and responsibilities, Process centric, People centric, Virtual team support, High Risk Mitigation, Medium Risk Mitigation, Documentation, Daily Meeting, Information sharing via document, Meeting face-to-face and Model-Based.

- **Team size:** The team size is the number of members of a development team required for maximizing workplace productivity of this Agile method. Each of these methods, supports a type of project to realize it and it needs a team, it defines the number of members per the roles to play in the development lifecycle.
- **Project size:** The size of the project determines the relative size of a project effort; which project management practices are applied. It is a very important in the management of a project because it is responsible for estimating the number of people to work on this project which also involves estimating the budget. The agile methods assert that any size of the project has found a light method that realizes the project with as few as possible of person in the team. Then there are three categories such as a small, medium and large project.
- **Iteration length:** It is the shorter iterations help to get better faster to release a product increment to the customer for this Agile method.
- **Roles and responsibilities:** It is the degree to which specific roles & responsibilities are defined for this Agile method.
- **Process centric:** It indicates to if this Agile method is process centric.
- **People centric:** It indicates to if this Agile method is more people centric.
- **Virtual team support:** It is the degree to which this Agile method supports virtual team communication and coordination.
- **Risk Mitigation:** A systematic reduction in the extent of exposure to a risk and/or the likelihood of its occurrence. Also, called risk reduction [20]. In this comparison, we use two levels of Risk Mitigation such as High-Risk Mitigation and Medium Risk Mitigation
- **Documentation:** It is written text or illustration that accompanies software increment. It either explains how it works or how to use it.

- **Daily Meeting:** They are organized to monitor the progress of the team continuously by the questions: what has been done since the last meeting and what needs to be done before the next one. Also, the problems and other variable issues are reviewed and monitored in this meeting.

- **Information sharing via document:** In a company, we think of two tools: document management and file sharing.

- **Document Management:** The tool supports the storage, sharing and retrieval of electronic documents. Managing the entire lifecycle of a document: search, version, comment, instant discussion, etc.

- **File Sharing:** The tool keeps the file in a space to allow remote sharing.

- **Meeting face-to-face:** Oral and face-to-face communication occupy within the company, the major part of the relational time of the actors of life in the company. The know-listen and know-speaking represent a preliminary to a good control of the telephone relation and the technique of the meetings.

- **Model-Based:** If the method is based on a model developed by the MDA approach.

5.2 Comparative Study

The research question that guided this study was, Which method to choose for which project? This question led to another question that is what is an agile method? The definition is different from one to another and when the developers team need to use a method, they must study and compare all the existing methods. That is why we effect an in-depth study of agile methods (i.e. study their processes, practices, etc.), in order to facilitate knowing the advantages and disadvantages of all these methods. This study is summarized in a comparison table below.

Method \ Criteria	SCRUM	XP	DSDM	FDD	LSD	KANBAN	CRYSTAL FAMILY	ASD
Team size	5 – 9	2 – 10	2 – 10	4 - 20	-	-	All	Small to Large Teams
Project size	All	Small	All	Large	All	All	All	Small
Iteration length (week)	4	2	-	2	-	1	Project Specific	4 – 8
Roles and responsibilities	√	√	√	√	X	√	X	X
Process centric	X	X	√	√	√	√	X	X
People centric	√	√	X	X	X	X	√	√
Virtual team support	√	X	X	√	X	√	√	√
High Risk Mitigation	√	X	√	X	X	X	√	√
Medium Risk Mitigation	X	√	X	√	√	√	X	X
Documentation	Basic	Basic	Exist	Important	X	X	Important	Basic
Daily Meeting	√	√	X	X	X	√	X	√
Information sharing (via document)	X	X	√	√	X	X	X	X
Meeting face-to-face	X	X	X	X	√	X	√	X
Model-based	X	X	X	X	X	X	X	X

-: Not specified X: Not defined √: Defined

Table 1. Comparative table of agile methods

5.3 Discussion

Agile methods are light for project management, they come to solve many problems due to conventional development (long of lifecycle, not response to change, etc.).

Several studies have already been carried out to evaluate the adoption rate of these emerging approaches. Among these studies, an international survey was conducted between July and December 2016. The results show that 25% of organizations use “agile” practices for about two years. However, 60% of respondents said that less than half of the teams in their organizations use agile practices. The “agile” methods used are the “scrum” method (58%), the “scrum” and “extreme programming” combination (10%), the combination “scrum” and “kanban” named Scrumban (8%), the combination of multiple methods (8%), Kanban (5%), DSDM / Atern methods, FDD, Lean Startup, Lean Development, XP, and AgileUP (5% for each of these methods) [21].

This study shows that these methods, even if have several benefits, they still suffer from some inconveniences.

The Scrum, XP and ASD methods are centered on the person. This criterion is among the principles of agile development. Also, they have another benefit, that is the highest level for the Risk-Mitigation. However, Scrum and ASD have a long duration of delivery of an iteration, then XP (4 weeks for Scrum and 4 to 8 weeks for ASD). Furthermore, we find a lack of communication between team members in DSDM and FDD methods, they share all information via documents, unlike the others, which use communication as information sharing, such as, daily meeting and face-to-face communication. Finally, none of these methods is model-based. This study is extensible to other agile methods.

6. Conclusion

To sum up, this paper presented a comparative study of the main agile methods. This study shows that although these methods present the advantages, they present also some of the disadvantages viz; they have high level of risk mitigation and vulnerabilities, which may lead to an attack on the company’s computer systems; none of these methods have the model-based that is leading us to model each method; and so on. To overcome these drawbacks, we are planning in our future work to make a new model-based agile method.

References

- [1] Li, J. (2010). Agile Software Development (2010).
- [2] Pekka Abrahamsson, Ed. (2002). Agile software development methods: review and analysis. Espoo: VTT.
- [3] Almseidin, M., Alrfou, K., Alnidami, N., Tarawneh, A. (2015). A Comparative Study of Agile Methods: XP versus SCRUM. *IJCSSE*, 4, 126–129 (May).
- [4] Leau, Y. B., Loo, W. K., Tham, W. Y., Tan, S. F. (2012). Software development life cycle AGILE vs traditional approaches. *International Conference on Information and Network Technology*, 37, p. 162–167 (2012)
- [5] Manifesto for Agile Software Development, <http://agilemanifesto.org/>.
- [6] Principles behind the Agile Manifesto, <http://agilemanifesto.org/principles.html>.
- [7] Cooke, J. L. (2012). Everything you want to know about Agile: how to get Agile results in a less-than-agile organization. In: Ely, Cambridgeshire, U.K.: IT Governance Pub.
- [8] Beck, K. (1999). Embracing change with extreme programming. *Computer*, 32 (10), p. 70–77.
- [9] Beck, K., Andres, C. (2005). Extreme programming explained: embrace change, 2nd ed. In: Boston, MA: Addison-Wesley.
- [10] Malhotra, C., Chug, A. (2013). Agile testing with Scrum-A survey. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3 (3) 452-459.
- [11] DESPA, M. L. (2014). Comparative study on software development methodologies, *Database Syst. J. BOARD*, 5 (3).
- [12] Voigt, B. J., Glinz, M. (2004). D.-I. C.Seybold. Dynamic System Development Method. Dep. Inf. Technol. Univ. Zurich Zurich
- [13] Goyal. S. (2008). Major seminar on feature driven development. Jennifer Schiller Chair Appl. Softw. Eng.
- [14] Palmer, S. R., Felsing. J. M. (2002). A practical guide to feature-driven development. In: Upper Saddle River, NJ: Prentice Hall PTR.
- [15] Boehm. B. (2007). A survey of agile development methodologies. Laurie Williams.
- [16] Livermore. J. A. (2008). Factors that Significantly Impact the Implementation of an Agile Software Development Methodology.

- [17] Harleen, F., K. C., Swati, V. (2014). A Systematic Study on Agile Software Development Methodologies and Practices. *IJCSIT*, vol. 5 (3).
- [18] Riehle, D. (2001). A Comparison of the Value Systems of Adaptive Software Development and Extreme Programming: How Methodologies May Learn from Each Other. *Extreme Program. Explain.*, p. 35–50.
- [19] Shelly. (2015). Comparative Analysis of Different Agile Methodologies. *International Journal of Computer Science and Information Technology Research*, India, p. 199–203 (March)
- [20] “Read the full definition.,” BusinessDictionary.com, <http://www.businessdictionary.com/definition/risk-mitigation.html>
- [21] The 11th annual state of agile report, <https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2> (June 2017)