



# Contribution of Agile Methodology for Software Process Improvement

Dr. Kirti Mathur

International Institute of Professional Studies, D. A. University, Indore

## ABSTRACT

Most software companies today aim to produce quality software in short time with minimal cost, and accepting changing environments. For this agile methodology is used. Unlike traditional methods, it employs short iterative cycles, and relies on tacit knowledge within a team to meet the new requirements of the software development companies. This paper presents how software process improvement is also ensued with the use of the agile approaches.

**KEYWORDS :** Agile methodologies; extreme programming; SCRUM

## INTRODUCTION

Software has embedded into many diverse fields, and is becoming more complex to develop. Changing requirements makes it even more difficult. As a result, new software development approach evolved, as agile methodologies, to solve such problem. This methodology includes modifications to Software Process, making it more productive and flexible.

Agile methodology is an alternative to waterfall method, typically used in software development. It helps teams respond to unpredictability through incremental, iterative work. [17]

In 1970, Dr. Winston Royce criticized the sequential development model. He asserted that software should not be developed like an automobile on an assembly line, in which every phase of the project is completed in lock-steps and because there was lack of communication possible among the specialized groups to complete each workflow. [13]

## 2. HISTORY OF SOFTWARE DEVELOPMENT APPROACHES

Software development approaches have evolved with time. The first development approach that came into existence in the 1970s [10] was structured methodologies, which divided the software development process into phases, to allow the developer focus their efforts on one phase at a time and resulted in successful software projects. Later software development approaches evolved into Object-Oriented methodologies and then Agile.

## 3. AGILE METHODOLOGIES

Earlier in software development, user requirements were static, and during development never accepted any changes. However, as software development involved more critical and dynamic industrial projects, new difficulties emerged according to the growth of companies. These difficulties were [1]:

- Evolving requirements: due to evolving business needs and experience.
- Lack of Customer involvement lead to project failure.
- Unrealistic deadlines and budgets.
- Miscommunications: between developers and customers due to use of jargon.

To overcome these problems new development methodologies were required.

A number of IT professionals started to work individually on new approaches and in 2001 they created the so called: Agile Manifesto. These approaches were developed based on the same rule that the best way to verify a system is to deliver working versions to the customer, and update it according to their notes. Agile authors built their methodologies on four principles:

- Develop software satisfying the customers need, through continuous delivery and feedback.
- Accepts changes in requirements.
- Promotes cooperation between the developers and the customers on a daily basis throughout the project development.
- Develops test-cases prior to implementation. [1].

Agility in short means stripping away the heaviness, commonly associated with traditional software development methodologies, in order to promote quick response to changing environments and user requirements to accelerate project deadlines [7]. Agile methodologies prefer software development not documentation. Their philosophy is to deliver many working versions in short iterations, and update the software on customers' feedback.

## Agile methodologies include:

- Extreme Programming
- Agile Modeling
- SCRUM

The main agile methodologies that are being used include XP, Agile Modeling, and SCRUM.

- XP is the coding of what the customer specifies, and the testing of that code.
- Agile Modeling defines a collection of values, principles, and practices which describe how to streamline modeling and documentation efforts.
- SCRUM, on the other hand, supports management role in software development.

## Extreme Programming (XP)

Extreme Programming was introduced by Kent Beck in 2000. XP offers a number of practices, values and principles for software development project [3]. Extreme Programming was in fact targeted especially at small co-located teams developing non-critical products [16]. Currently, however, it has been used by companies of all different sizes and industries worldwide [8].

XP provides a list of simple, specific, and seemingly naïve principles and values that guide the software development process throughout the main four phases of software development and also emphasizes teamwork; experiences from all stakeholders to meet goals within the given constraints [8].

## Software development phases are:

- Planning : Begins with the idea that describe functionality of would be software.
- Design : It provides implementation guidance (Refactoring).
- Coding : XP recommends that two people work together for code (Pair programming).
- Testing : Test cases are specified by customer, focusing on overall functionality.

This agile methodology or XP values, improves software development in four ways[8]:

S.N.	Value
1.	Rigorous communication between developer and customer.
2.	Design simplicity.
3.	Continuous modification on customer's feedback.
4.	Encouraging the customer and developers to correct requirements through develop-feed-back cycle.

**Table1. XP Values**

The programming principles that are encouraged by XP are:

- Its simplicity and flexibility to reduce maintenance costs of the software.
- The intensive and robust testing mechanism that reduces the number of defects after delivery.
- Embraces changes, during the development process.
- Moreover, encourages creating high quality code [8].

### 3.2 Agile Modeling (AM)

Modeling is important and enables software developers to think about complex issues prior to its implementation. Agile Modeling (AM) was established by Scott Ambler in 2002. It is a collection of values, principles, and practices for modeling software for development project in an effective and light-weight manner [2].

The values of AM are:

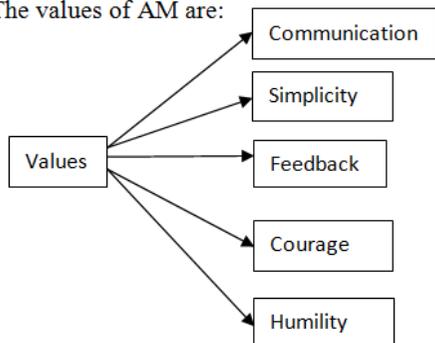


Figure 1. AM values.

Humility means to accept that you may not know everything; others may also provide useful contribution to the project [2].

Again, the principles of AM are quite similar to those of XP, such as assuming simplicity, accepting changes, incrementally of the system, and rapid feedback. In addition to these principles, AM include the knowledge of the purpose for modeling; having multiple effective models; the content is more important than the representation; keeping open and honest communication between parties involved in the development process; and finally, to focus on the quality of the work [2].

The practices of AM have some commonalities with those of XP, too. AM practices highlight on active stakeholder participation; focus on group work to create the suitable models; apply the appropriate artifact as UML diagrams; verify the correctness of the model, implement it and show the resulting interface to the user; model in small increments; create several function models in parallel; apply modeling standards; and other practices [2].

Agile Model Driven Development (AMDD) is the agile version of model driven development. To apply AMDD, an overall high level model for the whole system is created at the early stage of the project. During the development iterations, the modeling is performed per iteration, along with other methodologies, such as Test Driven Development (TDD), and Extreme Programming (XP), to get the best results [2].

AM basically creates a mediator between rigid methodologies and lightweight methodologies, by suggesting that developers communicate architectures through applying its practices to the modeling process [7].

### 3.3 SCRUM

SCRUM methodology was introduced by Ken Swaber in 1995 and was practiced before the announcement of Agile Manifesto. Later, it was included into agile methodology. SCRUM is used with the objective of simplifying project control through simple processes, easy to update documentation and higher team iteration over exhaustive documentation [6].

Scrum is a management framework for incremental product development using one or more cross-functional, self-organizing teams. It provides structure of roles, meetings, rules, and artifacts. Teams are responsible for creating and adapting their processes within this frame-

work. Scrum uses fixed-length iterations, called Sprints.

SCRUM comprises project management as part of its practices to guide the development team.

For the team of SCRUM roles are:

Roles	Function
product owner	Who mainly would be the voice of business.
SCRUM team(developers, testers)	Identify the need for a new product.
SCRUM master	Responsible for keeping team focused.

Table2. SCRUM roles.

Briefly, SCRUM is considered an iterative, incremental methodology of software development which can be used as a program management approach.

### 4. LIMITATIONS OF AGILE

Agile development aims to support early and quick development of operational modules that meets the user needs ignoring documentation and modeling and its limitations are [16]:

- It is not suitable for green-field engineering and maintenance, since there exist not much documentation for the systems.
- Secondly it depends heavily on the user involvement, and thus, the success of the project will depend on the customers.
- Another limitation is it concentrates work quality on the skills and behaviors of the individual developer, as the design of the modules and sub-modules are created mainly by single developer. When developing software to be reusable, then agile methodologies will not provide the best way.
- Agile methodologies work best for teams with relatively small number of members.

To get the advantages of applying agile methodologies in the development, there is a set of assumptions, assumed to be true. To mention some of them are: cooperation and face to face relation between the customers and the development team; evolving and changing requirements of user; developers having good individual skills and experiences.

### 5. CONCLUSION

Agile methodologies came into existence after the need to accommodate changing requirements was felt. It provides some practices that facilitate, communication between the developer and the customer, and undergoes develop-deliver-feedback cycles, to have more specific view of the requirements, and be ready for any change at any time. The main aim of agile methodologies is to deliver what is needed when it is needed.

Agile methodologies are not suited for all project types. When communication between the developer and the customer is difficult and development team has beginners, then agile methodology is not suitable. This methodology exhibits optimum results when there is strong communication between the developer and the customer, and the development team has skilled members. When there exists chances of misunderstanding customer's requirements, or when the deadlines and budgets are tight, then Agile methodologies is the best software development approaches to apply and will definitely improve the software process.

**REFERENCES**

- [1] Beck. K. and Andres. C.(2004), *Extreme Programming Explained: Embrace Change* (2nd Edition), Addison-Wesley, Boston. || [2] [www.agile-modeling.com](http://www.agile-modeling.com) || [3] <http://www.agilemanifesto.org>. || [4] <http://en.wikipedia.org> || [5] [http://en.wikipedia.org/Crystal\\_Clear\\_](http://en.wikipedia.org/Crystal_Clear_) || [6] Cristal. M, Wildt. D. and Prikladnicki. R (2008) Usage of SCRUM Practices within a Global Company. In IEEE International Conference on Global Software Engineering, pp 222-226. || [7] Erickson. J, Lyytinen. K and Siau. K, (2005), Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research. In *Journal of Database Management*, 16(4), pp 88-100. || [8] [www.extremeprogramming.org](http://www.extremeprogramming.org) || [9] Ferreira. J, Noble. J, and Biddle. R.(2007), Up-Front Interaction Design in Agile Development. In *Agile Processes in Software Engineering and Extreme Programming*, Springer, Berlin / Heidelberg, pp 9-16. || [10] Pressman (2009) R. S., "Software Engineering", 7th edition, McGraw Hill Education. || [11] Maurer. F. and Martel. S (2002) Extreme programming.Rapid development for Web-based applications.In *Internet Computing*, IEEE, 6(1), pp 86-90. || [12] [http://en.wikipedia.org/wiki/Object\\_oriented](http://en.wikipedia.org/wiki/Object_oriented) || [13] Schwaber. C. and Fichera. R. (2006), Corporate IT leads the second wave of agile adoption. Forrester Research. || [14] Singh. M. (2008) U-SCRUM: An Agile Methodology for Promoting Usability. In *Ag.AGILE'08. Conference*, Toronto, pp555-560. || [15] Tessem. B. (2003), Experiences in Learning XP Practices: A Qualitative Study. In *Extreme Programming and Agile Processes in Software Engineering*, pp131-137. || [16] Turk. D, France. R and B. Rumpe.(2002) Limitations of agile software processes. In *Proceedings of the Third International Conference on Extreme Programming and Flexible Processes in Software Engineering*. || [17] <http://agilemethodology.org/> ||