

Study on Agile Software Development Methodologies

Shalini Gautam

M. Tech. Student

Department of Computer Science & Engineering

*Ambedkar Institute of Advanced Communication Technologies
& Research, Delhi*

Dr. Manoj Kumar

Head of Dept.

Department of Computer Science & Engineering

*Ambedkar Institute of Advanced Communication Technologies
& Research, Delhi*

Abstract

In Today's challenging environment it is very much dynamic and organizations are rapidly changing their software requirements to adjust with present environment. Therefore they also need quick delivery of software products as well as for change in the requirements. In case of agile software development approaches, there has been a spectacular importance in the field of software development in recent few years. As a result of the fastness in agile approaches it brings rapid delivery of the final outcome for the life cycle of software development. With respect to the change in numerous areas that have been overblown, there is a need to realize that the components of the agile methodologies are different from the traditional one. In this paper, we have tried to describe the importance of agile software development methodologies and how the agile process can result in increasing the efficiency of the tasks.

Keywords: Agile Software Development, Types of Agile Methodologies, Agile Principles, Agile Manifesto, Traditional vs. Agile Process

I. INTRODUCTION

The term Agile refers to easy and quick movement, fast gathering and quick response. An agile process helps to divide the work into various number of sub tasks for adapting useable plans that can be used for frequent reassessments. Agility refers to the ability of creating and responding to change for making profit in a business environment. For example: Companies need to determine the amount of agility as to stand in a competitive environment Agile software development is a technique that provides a conceptual framework for promoting development iterations in the life-cycle of the project. Iteration is the process of developing a single unit of software per at a time, which may last from one to four weeks. To understand the work flow of agile process we need to visualize the figure given below:

Table – 1
Cause and Benefits of agile process

<i>S. No.</i>	<i>Causes for the Failure of Agile Process</i>	<i>Benefits Off Agile Process</i>
<i>1.</i>	<i>Due to Lack of experience</i>	<i>Believes in the on time or before product delivery policies.</i>
<i>2.</i>	<i>When Companies philosophy or culture not matches with the core agile values</i>	<i>Helps in enhancing the ability to support changing priorities</i>
<i>3.</i>	<i>Because of the lack of management support</i>	<i>Results in increasing the productivity</i>
<i>4.</i>	<i>Due to the external pressure to follow the traditional methods</i>	<i>Helps in improvising the software quality</i>
<i>5.</i>	<i>Because of communication problems</i>	<i>Helps in enhancing the predictability of the delivery</i>

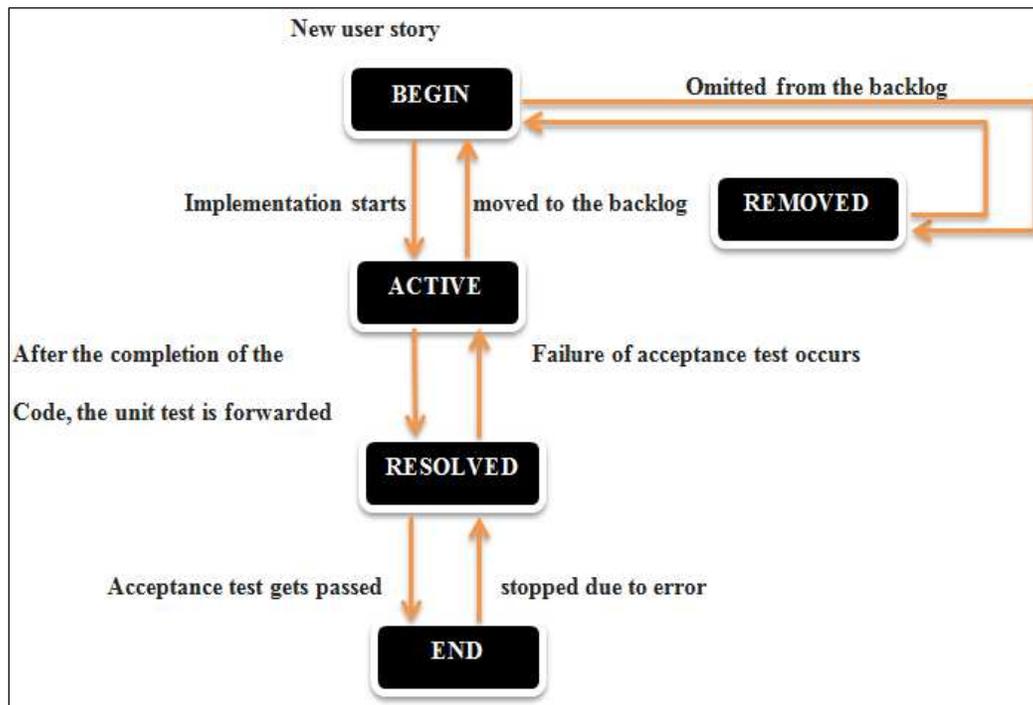


Fig. 1: Flow Chart of Agile Process

II. RELATED WORK

This article is divided into various sections to increase the understanding towards the approach. In section I: Introduction of the topic is well summarized. When we move to section III: Description of various authors view about their articles is shown in a tabular form. Section IV& V: Here, details of agile manifesto and 12 principles of agile modelling are presented. Section VI: describes the properties of traditional model and agile model. Section VII: deals with the comparison of different types of agile methodologies in a table format and in the last Section VIII: The conclusion is shown.

III. LITERATURE SURVEY

According to various authors, agile process is compared with different traditional software development methods which will help in deep understanding the need of agile software development methodologies. After observing various articles we have concluded some of the papers as follows:

S. No.	Paper Title	Goal/ Aim	Published In	Author Name
1.	Review of Agile Software Development Methodologies [2].	The aim of this paper to consistently review the agile software development techniques with their merits, demerits, and how traditional process is different with respect to agile process.	Feb, 2013	Kaushal Pathak, Anju Saha
2.	A Comparison between Agile Project Management and Traditional Project Management Methods [3].	In this article, the author has tried to show the comparison among traditional and agile process using various useful practices.	Oct, 2014	Hanadi Salameh
3.	Comparing Agile Software Processes Based on the Software Development Project Requirements [5].	The motive of this paper is to analyze the characteristics of various agile process for the developing a software project, which will help the project managers and software engineers to select an appropriate agile process that is most useful step in the software projects.	2008	Malik Qasaimah, Hossein Mehrfard, Abdelwahab, Hamou-Lhadj
4.	A Study of the Agile Software Development Methods, Applicability and Implications in Industry [4].	The main aim is to deal with the current agile methods and practices, and understanding the strengths and weaknesses using various issues of their applicability.	Apr, 2011	Kuda Nageswara Rao, G. Kavita Naidu, Praneeth Chakka
5.	Review of agile methodologies in software development [10].	The aim of the article is to analyze three agile approaches i.e., Extreme Programming, Agile Modeling, and SCRUM and describing the differences between them and recommends when to use them.	Oct, 2009	MALIK HNEIF, SIEW HOCK OW

IV. AGILE MANIFESTO

A. Communication with Individuals over Processes and Tools

Communication and interaction is considered as more important than raw talent. Correct tools selection will result in smooth functioning for the team. Let the team build the environment on the basis of need, as the more focus is on the development of a team rather than the development of the environment.

B. Working Software over Comprehensive Documentation

The team must provide human readable data/ documents that will help in understanding the system and design decision rationale. Code is the only simplest procedure that will help in understanding the source of information. Team memorizes each and every change occurred during the execution process of any system in their heads which cannot be visualized on paper.

C. Customer Collaboration over Contract Negotiation

The main focus is on intense collaboration with customer rather than details of scope and schedule.

D. Responding to Change over following a Plan

Better planning strategy are those which make detailed plans in the next few weeks, very rough plans in the next few months, and extremely crude plans beyond that.

V. PRINCIPLES OF AGILE MANIFESTO

The Agile Manifesto is based on twelve principles that are as follows:

- 1) First and main priority is to fulfill customer's satisfaction through fast and continuous delivery of useful product.
- 2) Agile teams work to keep the software structure flexible thus changes in the requirement are allowed even at the later stages for development.
- 3) Focus is on Delivering the Working Software Frequently within a couple of weeks with a preference to the shorter time period.
- 4) The Business analyst and Developers Must Work Together Daily throughout the Project. For successful modelling, Interactions among customer, developers and stakeholders plays a key role.
- 5) Build Projects around Motivated Individuals. That is the employees involved in the tasks are fully trusted and supported to provide
- 6) The Most Efficient and Effective Method of Conveying Information to and within a Development Team is face-to-face Communications.
- 7) Working software is considered as first and foremost selective measure of progress. Progress is not measured by the phase we are working on, or by the volume of the documentation we have produced or by the amount of code which coder has created.
- 8) Agile Processes promote sustainable development. The sponsors, developers and customers should be able to maintain the same speed indefinitely.
- 9) Continuous Attention to Technical Excellence and Good Design describes how much agility can be seen in your model.
- 10) Simplicity is considered as the art of maximizing the amount of work not done which is an essential part for any model that has to be built because they focuses on today's work rather than holding it for tomorrow.
- 11) The Best Architectures, Requirements, and Designs emerge from Self-Organizing teams as the Responsibilities are communicated to the team as a whole, and the team determines the best way to fulfill them.
- 12) At regular Intervals, the teams reflect on how to become more effective and cooperate during the time for completion of the target. The team knows well on how continually adjusts its organization, rules, conventions, relationships, etc.

VI. TRADITIONAL PROCESS VS AGILE PROCESS

Table – 3
Traditional process and agile process

S. No.	Features	Traditional Process	Agile Process
1.	Requirements	Requirements are well known to the users i.e. size is well understood therefore no changes can be made.	Changes can be made according to the requirements like what is the scope of the project and Are the available requirements correct?
2.	Resources (Money, Infrastructure, People etc.)	Considered as already available and approved by the higher authorities. The budget of the project is funded and sufficient. People are aware of working of tools and tasks.	May or may not be fully approved initially. Due to uncertain budget, money problem can arise. It requires new ideas, innovations and skills for better performance.
3.	Time	Time is already defined to clear the given milestones.	Time and milestones are not clearly defined. They may work according to the condition provided.

4.	Risks	Due to well understood scope impact of risk is minor in this case.	As new technologies are welcomed, unknown risks may occur that will give rise to major risks.
5.	Communication	In this case, formal communication is followed among the employees.	In agile process, communication is informal, i.e. interaction among customer and leaders.
6.	Control	Process centric control is carried out.	People centric control is followed.
7.	Customer's Role	They plays a very important role.	Critical roles are visualized in agile process.
8.	Development Model	They follow the life cycle models for the development of the task.	The incremental and iterative process cycle is followed for the tasks completion.
9.	Example	Waterfall model, spiral model etc.	Scrum, DSDM etc.

VII. COMPARISON AMONG AGILE SOFTWARE DEVELOPMENT METHODOLOGIES

The most popular agile methods are as follows: Rational Unified Process and Dynamic Systems Development Method (DSDM) (1994), Scrum (1995), Crystal Clear and Extreme Programming (1996), Adaptive Software Development, Feature Driven Development (1997). These are now collectively referred to as agile methodologies, after the Agile Manifesto was published in 2001.

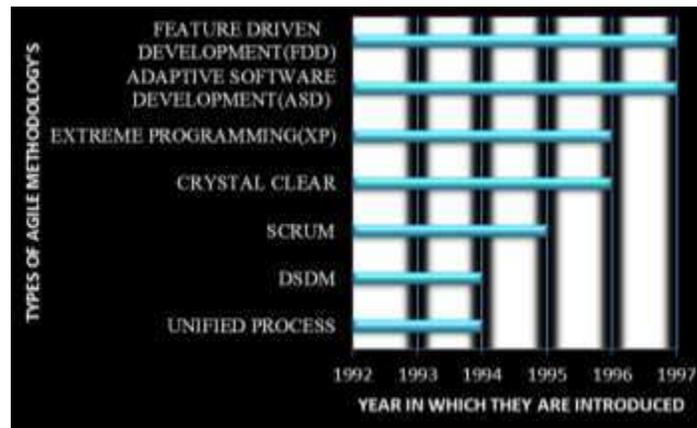
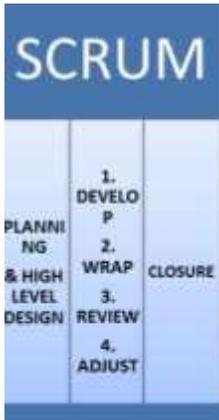
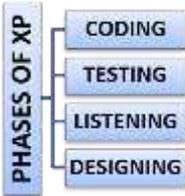


Fig. 2: types of agile methodologies

Table – 4
Comparison among Agile Methodologies

S No.	Name of Agile Methodology	Year	Description	Activities	Practices
1.	Unified Process	1994	<p>The Unified Software Development Process or Unified Process is a famous framework for the development of an iterative and incremental. Unified Process is popularly known as the Rational Unified Process (RUP). There are four phases of Unified Process for the development of an working software as:</p> <ul style="list-style-type: none"> - Inception Phase - Elaboration Phase (Milestone) - Construction Phase (Release) - Transition Phase 	<p>Fig. 1 : Phases of Unified Process</p>	<ul style="list-style-type: none"> - It helps in Developing iterative process - It deals with the criteria for the management of the requirements. - It makes use of the components. - Provides final results when the products quality is verified. - Manages the changes.
2.	Dynamic Systems Development Method (DSDM)	1994	<p>Dynamic systems development method (DSDM) is used as a framework to deliver fast and rapid software. DSDM provided as a base to the rapid application development (RAD) method. There are three phases of DSDM which are described as follows:</p> <ul style="list-style-type: none"> - Pre-Project Phase - Project Life-Cycle Phase - Post- Project Phase. <p>A project life-cycle phase is further classified into 5 different stages:</p>	<p>Fig. 2 : Phases of DSDM Method</p>	<ul style="list-style-type: none"> - Active involvement of the End-user. - Teams should be motivated. - Update for the frequent releases should be there. - Development driven mode should be activated to meet the business needs. - Changes are always welcomed.

			<p>Study for the Feasibility of software.</p> <ul style="list-style-type: none"> - Study for the improvement business perspectives. - Model iteration on the basis of their functionalities. - To build iteration after designing them. - Implementation phase. 		<ul style="list-style-type: none"> - Priorities must be given to the initial requirements. - Integration between development phase & testing phase must be done efficiently. - Procedure for the Collaboration and cooperation of teams is an important task to be considered.
3.	SCRUM	1995	<p>Scrum is also considered as a framework for the management of the product release through an iterative and incremental agile software development approach. It checks features such as flexibility, strategies for the development of holistic products where a development team works in a collaborative and cooperative manner to reach their desired targets. The success for the scrum methodology is that customer can change their minds regarding what are their requirements and what requirements they are expecting during the process of the production that is why it is called as volatility of the requirements and as result changes cannot be easily analyzed using traditional predictive or planned manner.</p> <p>The phases of scrum method involves:</p> <ul style="list-style-type: none"> - Sprint Planning Event - Daily Scrum (Or Stand-Up) - Sprint Review 	 <p>Fig. 3: Phases of Scrum</p>	<ul style="list-style-type: none"> - Improves the deliverables quality. - Concept of the changes through review or feedbacks must be established. - Must provide better view and ideas at minimum time for the releases. - Be more in control of the project schedule and state.
4.	Crystal Clear	1996	<p>Crystal Clear is a methodology that can be applied to the teams of up to 6 or 8 members or developers working together on a system that does not going to harm any of the individual. Their main focus is on the efficiency and habits they adapt these features for the safety of the project. Crystal Clear main motive is to fulfill the needs people and not the processes or artifacts.</p>		<ul style="list-style-type: none"> - Fast deliverables. - Reflective improvement - Communication must be one to one and doubts should be clarified. - Safety of the individuals. - Much focused and clear regarding their achievements. - Expert users can easily access the requirements. - Deals with the technical features like tests automation, configuration management and frequent integration

5.	Extreme Programming (XP)	1996	<p>Extreme programming is also called as XP implies a software development methodology that can be used to improve the quality of the software and its response to the changing environment with respect to the requirements of the customers. The methodology works on the principle of considering the important practices done by traditional software must be taken as the initial base as to reach the "Extreme" phase.</p> <p>There are main four basic activities of XP for software development process are:</p> <ul style="list-style-type: none"> - Coding - Testing - Listening - Designing 	 <p>Fig. 4 : Phases of Extreme Programming Method</p>	<ul style="list-style-type: none"> - Fine scale feedback - Programming is done by formation of pairs and the planning is done on how to move forward and deal with the situations. - Software is released after performing certain tests which are very much necessary for the software quality development. - Process works continuously until target is achieved. - Continuous Integration - Improvement in the designs. - Products are released at small intervals. - Mutual understanding among the team members is always required. - Coding Standards - Collective Code Ownership - Designs are simpler.
6.	Adaptive Software Development (ASD)	1997	<ul style="list-style-type: none"> - Adaptive software development (ASD) is a process that follows the policies of rapid application development (RAD) work developed by Jim Highsmith and Sam Bayer. There are three activities that are followed by ASD approach are as follows: - Speculate: It means to planning paradox that implies that all stakeholders are comparably incorrect for certain amount of time during the production of the project. - Collaboration: It refers to the managing of the task based on predictable parts of the environment (planning and guiding them) and considering the improper surrounding mixtures which occurred due to factors like requirement of the technology, number of stakeholders and software vendors. - Learning cycles: It means challenges by the stakeholders which will work on the small iterations with the help of designing phase, build phase and testing phase. 	 <p>Fig. 5: phases of ASD process</p>	<ul style="list-style-type: none"> - The main principles on which ASD life cycle works are that it focuses on the targets, follows feature based approaches that are iterative, time-boxed, risk driven, change tolerant.

7.	Feature Driven Development (FDD)	1997	<ul style="list-style-type: none"> - Feature-driven development (FDD) is referred as an iterative and incremental software development process. It is one of the framework which helps to handle a number of lightweight approaches for the development of the software. Their main aim is to deliver correct and working software every time. FDD consists of five basic activities for developing a tangible working software are: - Overall development of the model. - Building feature list. - Feature based planning. - Feature based designing. - Feature based building. 	 <p style="text-align: center;">Fig. 6: Phases of FDD Method</p>	<ul style="list-style-type: none"> - Modeling with the help of objects domain. - Feature based development. - Follows Ownership using Individual classes. - Team works on feature based approach. - Managing the Configuration. - Regular updates. - Progress results are much clear and visible.
----	----------------------------------	------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

VIII. CONCLUSION

Agile approaches are meant to increase fastness and flexibility in the software projects. The reasons for adopting agile methods are adaptability to change, releases product in short period of time, customers review and feedbacks are given importance, high-quality product as outcome and software is always error free. In this paper, we presented our analysis of three agile software processes. The objective is to help software engineers to understand the need of agile methodologies with respect to traditional methodologies and therefore select the most suitable process for developing the high quality software projects. In the comparative study of agile software development with other traditional models we conclude that agile project provides better results in terms of productivity, performance, faster time cycles, risk analysis. Also, we have analyzed various types of agile software development methodologies on the basis of different features in a form of table to get them more clearly. Thus, after the analysis of each and every requirement we conclude that agile process can be a winning formula to deal with various kinds problems.

ACKNOWLEDGEMENT

I Shalini Gautam pursuing M.Tech in Computer Science and Engineering from Ambedkar Institute of Advanced Communication Technologies & Research, GGSIPU (Delhi), take this opportunity to express my deep gratitude and thanks to Dr. Manoj Kumar, Head of Computer Science and Engineering Department, Ambedkar Institute of Advanced Communication Technologies & Research GGSIPU (Delhi) for his encouragement, inspiration and continuous support for completing this paper. I express my sincere thanks to him for his valuable suggestions.

REFERENCES

- [1] Harleen K. Flora, Swati V. Chande, "A Systematic Study on Agile Software Development Methodologies and Practices", International Journal of Computer Science and Information Technologies, 2014.
- [2] Kaushal Pathak, Anju Saha, "Review of Agile Software Development Methodologies", International Journal of Advanced Research in Computer Science and Software Engineering, 2013.
- [3] Hanadi Salameh, "A Comparison between Agile Project Management and Traditional Project Management Methods", International Journal of Business and Management Review, 2014.
- [4] Kuda Nageswara Rao, G. Kavita Naidu, Praneeth Chakka, "A Study of the Agile Software Development Methods, Applicability and Implications in Industry", International Journal of Software Engineering and Its Applications, 2011.
- [5] Malik Qasaimah, Hossein Mehrfard, Abdelwahab, Hamou-Lhadj, "Comparing Agile Software Processes Based on the Software Development Project Requirements", IEEE: International Conference on Conference: Computational Intelligence for Modelling Control & Automation, 2008.
- [6] Qumer, A., & Henderson, "A framework to support the evaluation, adoption and improvement of agile methods in practice", The Journal of Systems and Software, 2008.
- [7] Hassan Hajjdiab and Al Shaima Taleb, "Adopting Agile Software Development: Issues and Challenges", International Journal of Managing Value and Supply Chains, 2011.
- [8] Laurie Williams, "Chapter: A Survey of Agile Development Methodologies" 2007.
- [9] Gaurav Kumar, Pradeep Kumar Bhatia, "Impact of Agile Methodology on Software Development Process", International Journal of Computer Technology and Electronics Engineering, 2012.
- [10] MALIK HNEIF, SIEW HOCK OW, "REVIEW OF AGILE METHODOLOGIES IN SOFTWARE DEVELOPMENT", International Journal of Research and Reviews in Applied Sciences, 2009.
- [11] Pekka Abrahamsson, Outi Salo, Jussi Ronkanien, Juhani Warsta, "Agile Software Development Methods" VTT Publications, 2002.
- [12] Juyun Cho, "ISSUES AND CHALLENGES OF AGILE SOFTWARE DEVELOPMENT WITH SCRUM", Issues in Information Systems, 2008.
- [13] Tore Dyba, Torgeir Dingsoyr, "Empirical studies of agile software development: A Systematic Review", Information and Software Technology- Elsevier, 2008.