Agile Software Development Methods, Comparison with Traditional Methods & Implementation in Software Firm

Neeraj Bhadoriya[1] ,
Neha Mishra[2] , Alok Malviya[3]
[1,2] Dept. of Information Technology
Institute of Engineering & Technology, Devi Ahilya Vishwavidhalaya
Indore-452001 (M.P.)
[3] Institute of Management Studies, Devi Ahilya Vishwavidhalaya
Indore-452001 (M.P.)

Abstract: **In today's world business is very much dynamic in nature and organisation are changing their requirements very quickly and for that there is a tremendous importance of agile methods in software development , because of its nature of being fast as compare to traditional method it is used very rapidly in software development process. Though traditional methods like waterfall, object oriented approaches have dominated past decades but Agile software development brings its own set of novel challenges that must be addressed to satisfy the customer through early and continuous delivery of the valuable software .The paper carries the distinct objectives to define current agile method understanding different agile methods, the uses of agile and drawbacks. To meet the set goals and objectives, we used qualitative and research methodologies. Apart from taking references and review, we performed research from a software firm named Incubation Centre which is in Indore**.

Keywords: Agile methods, scrum, Extreme programming (xp), DSDM, Crystal, Feature driven development, Waterfall, software development life cycle (SDLC) etc.

I. INTRODUCTION

[3].Software development is a process to develop our software in a fast, efficient and cost effective way .To fulfils this requirement there is a need of some method which is faster over changing business requirement. This method is defined as agile methods. Agile is a method, which emphasis on iterative software design and development, allow team to release software features at frequent intervals.

Why should we use Agile Methodology to develop software?

Agile software development methods are an iterative approach to software specification, development and delivery. Now a day's business requirement or customer requirement changes frequently so, ASD methods are applicable on those application where system requirement changes (may be possible) many times within a day. These methods are intended to develop and deliver working software to the software owner as fast as possible, after the first version of the software the customer will propose new changed requirement that will be implemented by the software development team in next iteration.

Our paper focused on the agile software development, as agile methods is mostly used in current industry. Most commonly this examined the advantages, drawbacks and their adoption in industry. In order to study and analyse there must be need of taking data from a software industry to see what their expectation from it are. This will lead us to find benefits, limitations and difficulties in transition from traditional to agile software development. The rest of paper is as follows: section 2 contains background & related works, section 3 contains different agile methods, and section 4 contains Comparison of agile with traditional method of software development 5 contains research design, section 6 contains findings of our paper, and section 7 contains conclusion and future work. The last section contains the references.

II BACKGROUND & RELAYED WORK

[2]Agile software development (ASD) is major paradigm, in field of software engineering which has been widely adopted by the industry, and much research, publications have conducted on agile development methodologies over the past decade. The traditional way to develop software methodologies follow the generic engineering paradigm of requirements, design, build, and maintain. These methodologies are also called waterfall–based taking from the classical software development paradigm. They are also known by many other names like plan–driven, (Boehm and Turner, 2004); documentation driven, heavyweight methodologies, and big design upfront, (Boehm, 2002).

A. *Agile manifesto*. [1]*:*

Twelve principles behind the Agile Manifesto

1: Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

2: Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3: Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4: Business people and developers must work together daily throughout the project.

5: Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6: The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

7: Working software is the primary measure of progress.

8: Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9: Continuous attention to technical excellence and good design enhances agility.

10: Simplicity--the art of maximizing the amount of work not done--is essential.

11: The best architectures, requirements, and designs emerge from self-organizing teams.

12: At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

III. DIFFERENT METHODS OF AGILE SOFTWARE DEVELOPEMENT METHOD

- Extreme programming (Beck 1999, Beck 2000)
- Adaptive Software Development (Highsmith, 2000)
- Scrum (Schwaber and Beedle, 2001)
- Crystal (Cockburn,2001)
- Dynamic System Development Method (DSDM) (Staplaton, 1997)
- Features Driven Development (FDD) (Palmer and Felsing, 2002)

There are different types of agile methods. they are differ from each other in some manner naming iteration time, team size, working rules, collection of user requirement, values, practices, and principles of agile methods. But all these methods share some common principals that are given in the table: **[10]**

TABLE I
PRINCIPALS FOR AGILE METHODS

Principals	**Description**
Customer involvement	Customer should be closely involved throughout the Development process. Their role is provide and priorities New system requirements and to evaluate the system iteration.
Incremental delivery	The software is developed in increments with the customer Specifying the requirements to be included in each increment.
People not process	The skills of the development team should be recognized and Exploited. Team member should be left to develop their own ways Of working without prescriptive processes.
Embrace change	Expect the system requirement to change, so design the system to Accommodate these changes.
Maintain simplicity	Focus on simplicity in both the software being developed and in The development process. Wherever possible, actively work to Eliminate complexity from the system.

- Customer involvement in the software development process was new in agile method. Customer can give the detailed information of the user requirement for the intended software. If project stakeholder (customer) is not able to convey the proper requirement then it will become a problem.
  - If team members does not interact with the other members of the team then situation may be critical to handle.
  - To prioritize the changes requirement of the stakeholder is another difficulty with agile practices. This problem arises when

number of stakeholder are more, and there requirement are different.

- Simplicity of the product is not easy task; it does require extra time and effort to maintain. But there is a time constraint to deliver the working software to customer then it will create a pressure on development team.

Every method has some advantages and disadvantages over other methods. Agile also have some limitations. Agile is only suited for small and middle size of business applications. They are not suited for the large scale and complex system development. Another problem with agile is we cannot work on distributed development team means team is in different places. We cannot use agile methods for the complex and critical system where deep analysis of the requirement is needed. Because most of the critical system is related to safety and security, so to proper understanding of this problem deep analysis is necessary.

A. *Extreme Programming:*

Extreme programming as name suggested its takes an extreme approach to develop software. The incremented version delivered one and more times within day and complete developed software delivered within two weeks. The development team develops new version or new incremented software they run all existing tests software to insure that there is no error in the incremented version before delivering the software to the software owner. If all tests executed successfully then we deliver the new version to the customer.

Extreme programming (XP) is most widely used of the agile methods. In extreme programming the incremental delivery and customer involvement is in 'extreme' level. In extreme programming product stakeholder give the requirement as scenario or as a user stories. These stories further divided into task by development team and assign this task to team members. The test code is written first before the actual code is written by the development team. And after that these prewritten test code is applied to the actual product code to test the error in the software code. When new requirement is added to the system the prewritten test code is executed if this task is successfully implemented it means there is no error in the system. [10]

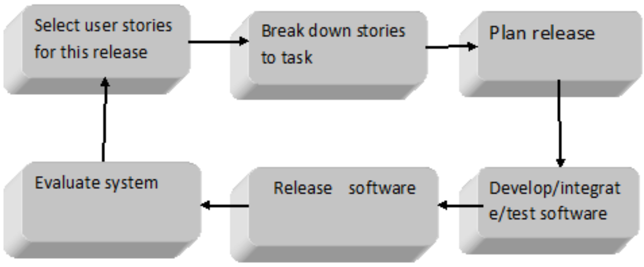

Select user stories for this release
Break down stories to task
Plan release
Develop/integrate/test software
Release software
Evaluate system

Fig. 1 shows the XP process to produce an increment of the system that is being developed[10].

3) ***Practices that involved in extreme programming are summarized in the following table:*** Delivery of the working software is incremental approach this approach followed by small, and frequent release of the software by recognizing the user requirement as user stories or scenario that is our basis for process planning.

Customer representative communicate with the development team to give the detailed information of the user requirement. They engaged with the development team during the development process.

The whole development team is owner of the code. It is known as pair programming.

Changes are implemented in the system by integration of the new changes to the system. And test code is applied to the new incremented version of the system.

Simplicity is maintained by the refactoring method. The repeated code is removed from the code and using the coding quality technique we can maintain the simplicity of the system.

In an XP process, customer is always available to specify and prioritize the user requirement. The customer is member or part of the development team. Customer shares the requirement with the other members of the team. The new requirement of the customer is integrate into the new incremented version of the software. If new requirement of the customer are totally different from the previous one then developed story card, if they are not implemented in the software then they are discarded and new story card are created according to the new requirement. Story card that are developed by the team are further breakdown into the task, and estimate the resource and efforts that are required to implement these task**.**

TABLE II

EXTREME PROGRAMMING'S PRINCIPLES & PRACTICES[10]

Principal or Practices	**Description**
Incremental planning	Requirement are recorded on Story cards and the Stories to be Included in a release are determined by the time available and their Relative priority. The developers break these Stories into development 'tasks'.
Small Releases	The minimal useful set of functionality that provides business values is developed first. Releases of the systems are frequent and incrementally add functionality to the first release.
Simple design	Enough design is carried out to meet the current requirements and no more.
Test-first development	An automated unit test framework is used to write tests for a new piece of functionality before that functionality

	itself is implemented.
Refactoring	All developers are expected to refractor the code continuously as soon as possible code improvements are found. This keeps the code simple and maintainable.
Pair programming	Developers work in pairs, checking each other's work and providing the support to always do a good job.
Collective ownership	The pairs of developers work on all the areas of the system, so that no island of expertise develops and all the developers own all the code. Anyone can change anything.
Continuous integration	As soon as work on a task is complete it is integrated into the whole system. After any such integration, the entire unit test in the system must pass.
Sustainable pace	Large amount of overtime are not considered acceptable as the net effect is often to reduce code quality and medium term productivity.
On-site customer	A representation of the end user of the system (the customer) should be available full time for the use of the XP team. In an extreme programming process the customer is a member of the development team and is responsible for bringing system requirements to the team for implementation.

B. *Adaptive method:*

Adaptive Software Development is another agile software development process that is developed by Jim Highsmith and Sam Bayer (2000). In Adaptive software development first we specifies the user requirement prepare a document that tells about the feasibility and risk assessment during the software development. If this prepared document is approved by the customer then analysis of the software start. The analyst analysed the requirement and quality of the software and produces the software specification document. After SRS document incremented design process started. In ASD prototype method is used to design the system. Testing process of the software follows the unit test of each incremented version. The test code developed first before the actual code. The final test of the software is acceptance test that is done by the customer. Customer provides some input to test the developed software. If acceptance test is successful then software is delivered to the customer [14].

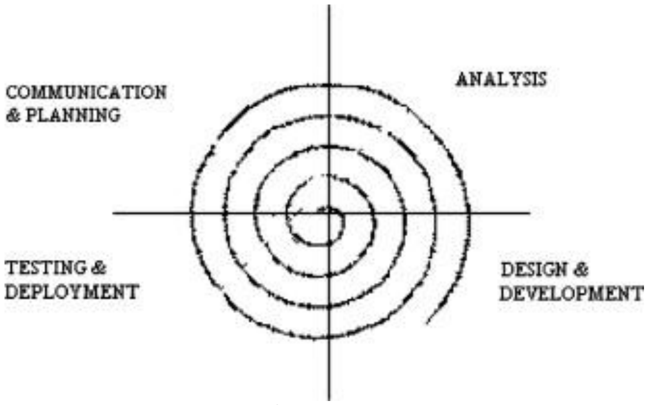


Fig. 2 : Adaptive software development method [4].

C. SCRUM*:*

Scrum (Ken Schwaber and Mike Beedle. 1990s) is iterative incremental process that is used for development of software. Its principle based on the fact that small teams working cross functionally produces good results. Scrum is more revenue centric with attention on improving revenue and quality of the software. Since being lightweight it can adapt to changing requirements and releases the software in small release cycles called sprints. Development team is divided up to 7 persons team. Iteration length varies from 1 week to 6 weeks. Project may consist of multiple teams that could be distributed [13].

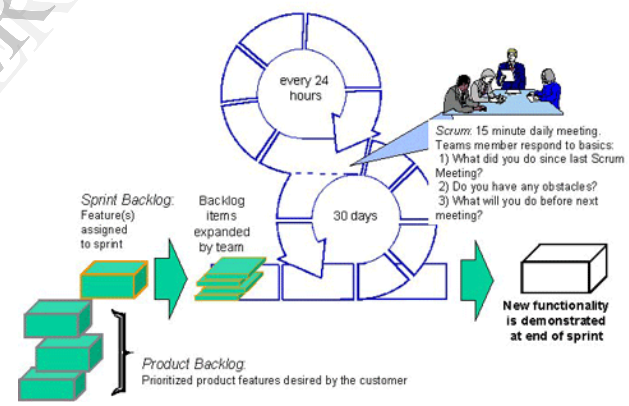

every 24 hours
Scrum: 15 minute daily meeting.
Teams member respond to basics:
1) What did you do since last Scrum Meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?
Sprint Backlog:
Feature(s) assigned to sprint
Backlog items expanded by team
30 days
New functionality is demonstrated at end of sprint
Product Backlog:
Prioritized product features desired by the customer

Fig. 3 scrum software development process [4]

D. *Crystal:*

It is most adaptable approaches to software development. This method comprises of family of agile methodologies and having scaling according to team size, system criticality , different colors denote weight like crystal clear ,crystal yellow , crystal orange , crystal orange web , crystal Red , crystal diamond. This family is specially designed to meet the project's unique characteristics [13].

E. *Dynamic System Development Method (DSDM):*

This is agile development framework which is used as development method. This is framework which embodies knowledge about the current project. It is based on nine key principles that primarily revolve around business needs, active user involvement, empowered teams, frequent delivery, integrated testing and stake holder collaboration[13]

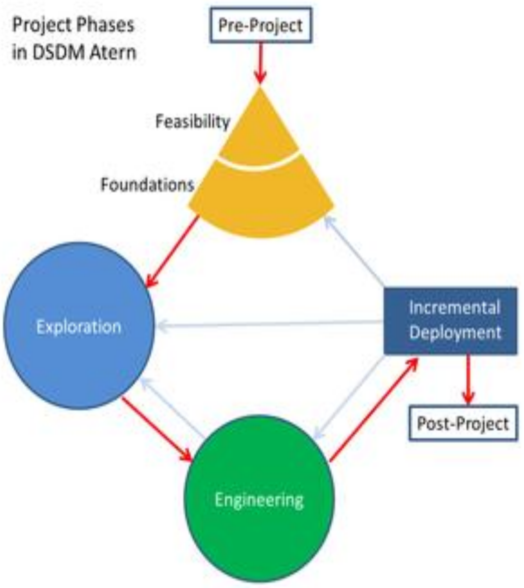
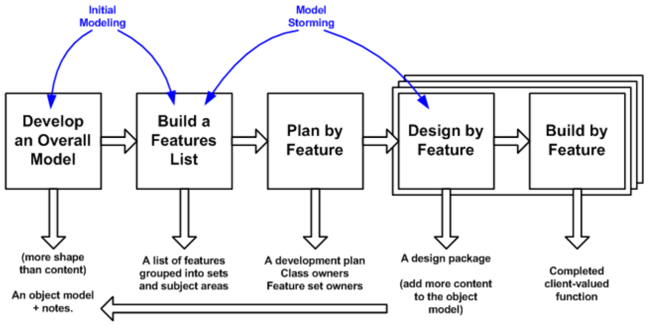


Fig. 4Dynamic software development process [4]

F. *Feature Driven Development (FDD):*

Feature Driven Development (FDD) was introduced in 1997 by Jeff De Luc when he was working in a software development project for a large Singapore bank. The FDD is highly adaptive software development process that is highly and short iterative, emphasis quality at all steps, delivers frequent, tangible working results at all steps so it is like by client, managers and developers [12].


Initial Modeling
Model Storming
Develop an Overall Model
Build a Features List
Plan by Feature
Design by Feature
Build by Feature
(more shape than content)
An object model + notes.
A list of features grouped into sets and subject areas
A development plan Class owners Feature set owners
A design package
(add more content to the object model)
Completed client-valued function

Copyright 2002-2005 Scott W. Ambler
Original Copyright S. R. Palmer & J.M. Felsing

Fig. 5 Feature Driven Software Development process [4]

IV. COMPARISION of AGILE with TRADITIONAL METHOD of SOFTWARE DEVELOPEMENT[5], [6], [7], [8], [9]

Traditional Approaches	**Agile Approaches**
Deliberate and formal, linear ordering of steps, rule-driven.	Emergent, iterative and exploratory, beyond formal rules.
Optimization is the goal.	Adaption, flexibility, responsiveness is the goal.
In this type the environment is taken as stable and predictable.	In this type the environment is taken as turbulent and difficult to predict.
Sequential and synchronous process.	Concurrent and asynchronous process.
It is work centered process because people will change according to different phases.	It is people centred process, as the same team is developing throughout.
Project lifecycle is guided by tasks or activities.	Project lifecycle is guided by product features.
Documentation is substantial.	Documentation is minimal.
Developers do waiting until the architecture is ready.	The whole team is working at the same time on the same iteration. Good coordination between team members
Too slow to provide fixes to user.	Provide quick responds to user feedback
Change requirements is difficult in later stages of the project	Can respond to customer requests and changes easier
More time is spent on design so the product will be more maintainable. The "what ifs" arise earlier	There is no time for the what ifs
No communication within the team, novices stay in their rooms and try to understand things	High level of communication and interaction, reading groups, meetings
Restricted access to architecture	The whole team influences and understands the architecture. Everybody will be able to do a design presentation
Documents and review meetings are needed to solve an issue	5 minutes discussion may solve the problem
Everything is up front, everything is big before you start	The focus is on whether customer requirements are met in the current iteration

V. RESEARCH DESIGN

In this section of research design , a research is conducted using qualitative method. The data was collected from a software firm named as Incubation Center , which are developing software for various multinational companies. There are having around 30 teams in their groups. Each of having 10 members. The main focus of these teams is to develop software according to fast complex set of rapidly changing requirement of their customers. They are mainly working on php and java application.

They are using agile methods in most of their projects and even extreme programming methodology of agile more specifically , we can say so because we have taken data of there some of the projects like MPEB (Madhya Pradesh electricity board) , static project. In MPEB project , they are been given initial requirement as to develop html pages of the register of 1000 pages , means simply converting hard copy data into softcopy format. Then after completing this phase in one week, the customer dealer of their firm then shows this project to client. Then they will give other suggestion or modification in the project . They said to have add and delete option of pages . so they will develop this in next phase . In this way we can say that they are actually following extreme programming methodology , in which they are firstly collecting user requirement as user stories , develop it in phase to phase and finally deliver it.

Secondly, we have to focus to ask them what changes agile method has brought as compare to traditional method. The purpose was to know the perspective of the team member , the change in work they experienced and to know the change from the customer's point of view.

VI. FINDINGS

As we have studied that they are using agile methodology in their software development work. By taking views of various teams of Incubation Centre, there teams gives some benefits and drawbacks of using agile methods.

A. *Benefits:*

As we have already said that they are implementing XP method in most of their projects, one benefit could be that it is capable of quickly adopting changing requirements of their clients.

Team 3 believe that because of implementing XP programming a high quality of data can be made as compare to traditional method like waterfall.

Taking views from the company manager, he mentioned some benefits of agile methods.

Firstly, because of agile method there is cost effectiveness in developing their software it is so because in agile , they don't have to follow whole SDLC so it saves their time and money as documentation and analysis is not required .

Secondly, because of agile method there is customer involvement in every phase of development , like by giving initial requirement to check whether the software is up to their type or not for that they are constantly in the loop , suggesting improvement and reviewing in each phase one of the project mention above of MPEB , we have seen that after developing html pages , they had suggesting other improvements as to have add and delete option also , so it is having great benefit as compare to traditional method, in traditional method when their team was using it previously what they see is , client gives its requirement with detail explanation of what he wants then it up to develops how they develop , after completing whole project the client will see it , and that time changes cannot done and at time we have to fully change the software so it waste our hard-work .

Team 5 says that on implementing agile , small and medium sized system are perfect for agile development since they offer more compatibility and because of that there is no problem of bulkiness.

B. *Drawbacks:*

Every coin has two sides; agile too has some drawbacks mentioned by team members.

Firstly, Team 4 faces problem that they software requirement are to fluctuating that proper scheduling is difficult , project have to deliver very quickly that sometimes create problem in software.

Secondly, managers believes that sometimes agile process could bring their deficiencies to light, so always there is pressure on the person . As the project requirement change so it is absolutely necessary to have experienced and senior resources in the team or at least experienced resources mentoring new resources.

In the review of company , we say that what changes they faced in working with agile is that firstly, now they had

Very small time for developing software , which is somewhat increasing their programming ability to deliver software properly in time.

Secondly , because of agile dealing with clients give happiness by seeing more satisfaction which is not possible in traditional waterfall method .

VII. CONCLUSION

There is no unique recipe for success in software development field. To overcome the some limitation of waterfall model, the new method of software development was developed that is known as agile. There is many software development processes naming water fall model, rapid software development process, traditional, prototyping, agile. But our paper only focuses on agile method. Agile is an iterative or incremental approach to specify user requirement, develop and deliver the working software to the customer.

If any organization is using agile for software development, it is very necessary to follow every specification, practices and process that is prescribed by that process. In agile team communication is must. An expert or an experienced person is needed to support the development team. Because an experienced person can think about a solution before the problem arises. In agile there is many practices i.e. followed by and common in many agile methods.

We have already explained the agile methods which are used by any software development team to develop software. There are many principles that are defined by agile manifesto which are necessary for the software development team and customer. Or we can say that it is necessary for every stakeholder to follow these principles which is the part of the process.

As we have already seen the advantages and disadvantages of the agile process. This is up to the organization what method they use to develop the software. Agile is the best method for small and medium size business application. With the frequent changing requirement we can develop the working software with many iteration or increment of the complete software. The complete software may take two or four weeks to deliver to the product owner. The team communication, team co-ordination, and a manager who, manage the development team, customer representative who communicate with the development team play very important role in agile process.

VIII. FUTURE WORK

This paper concentrates on the explanation of the agile methods ,its benefits and drawbacks. There are many research areas in agile which are left. For example, can we use agile method in distributed environment? Can we use this method for complex applications? Can distributed team can work together to develop one single software? We have found some answers during work on this paper that are agile does not support distributed environment. It is true the risk of software failure is less in agile but we cannot use this method for complex system because in complex system a deep analysis in each part of the system is very necessary.

REFERENCES

[1] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Humt, A., Je®ries, R., Kern, J., Marick, B., Martin, R.C., Mellor, S., Schwaber, K., Sutherland, J., Thom, D.: *Manifesto for agile software development. Website (2001) http://agilemanifesto.org/*.

[2] A B M Moniruzzaman, Dr. sayyad Akhter Hossain *" comparative study on agile software development methodologies".*

[3] kuda nageswara Rao , G. kavita Naidu , Praneeth Chakka Dept. co Computer Science & Systems Engineering ,College of Engineering , Andra university vishakapatanam , A.P. india *" A study of agile software development Methods, Applicability and Implications in industry".*

[4] http://www.controlchaos.com/images/diagram/flow.gif

[5] S. Nerur and V. Balijepally*, "Theoretical Re -fl ections on Agile Development Methodolo-gies,"* Comm. ACM, vol. 50, no. 3, 2007, pp. 79–83.

[6] S. Nerur, R. Mahapatra, and G. Mangalaraj, *"Challenges of Migrating to Agile Methodolo-gies,*" Comm. ACM, vol. 48, no. 5, 2005, pp. 72–78.

[7] P. Schuh, Integrating Agile Development in the Real World, Charles River Media, 2004.

[8] Mikio Aoyama, "*Agile Software Process Model",* IEEE Software, pp 454-455

[9] Aoyama, M.*, "Agile Software Process model",* IEEE Software, 2002

[10] Software engineering book seventh edition, published by Pearson education, ISBN 81-297-0867-1

[11] http://www.versionone.com/Agile101/Agile-Development-methology-Scrum-Kanban-Lean-XP/

[12] Kaushal Pathak Anju Saha *USICT, Guru Gobind Singh Indraprastha University, " Review of Agile Software Development Methodologies"* 2013, IJARCSSE Volume 3, Issue 2, February 2013

[13] http://www.umsl.edu/~sauterv/analysis/6840_f09_papers/Nat/Agile.l