IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Progressive Product Development with Agile

Y. Padma[1], Dr. P.V.S. Lakshmi[2], A. Haritha[3], G. Venu Gopal[4]

Assistant Professor, Department of IT, P.V.P. Siddhartha Institute of Technology, Vijayawada[1, 3, 4]

Professor, Department of IT, P.V.P. Siddhartha Institute of Technology, Vijayawada[2]

Abstract: In the rapidly growing world, the concept of the traditional development needs to evolve. The agile methodology is a more acceptable practice, based on producing software at a more rapid pace, while still maintaining efficiency. This being particularly useful for smaller software production firms with limited resources. The agile methodology emphasizes on the quality issue and provides a very stable backbone for today's software development. In this report we discuss the history of agile methodology, in a general context, with listing of the agile manifesto and the agile principles. We explain four of the existing agile methodologies with more focuses on the famous and mostly known agile process: the Extreme Programming. The report includes a discussion about the critical success factors, benefits and weakness of the agile methodologies based on a number of existing surveys with some real world examples that shows some of Agile methodology's advantages and disadvantages.

Keywords: Agile Methodology, Progressive Product Development, Extreme Programming, Non-Agile Projects.

WHAT IS AGILE?

Agile methodology is one of the development processes to build a new software substitutes the conventional strategies. This differs widely from other software process models like Waterfall model, V-Model, Iterative model etc. Agile means ‘ability to think quickly and clearly’ and responding speedily to change, is a key mark of agile software development. It helps teams respond to changeability through incremental, iterative work measures, known as sprints.

OVERVIEW

- In conventional software development methodologies a project can take more than a few months or years to complete and the customer need to wait until the end of the project.
- Non-Agile projects will assign lot of time for requirements elicitation, design, coding, testing and UAT, before the deployment of a project.
- In contrast to this, agile projects have sprints which are shorter in duration (Sprints/iterations can vary from 2 weeks to 2 months) during which predestined features are developed and delivered.
- Agile projects can have one or more iterations and deliver the complete product at the end of the final iteration.

The five values of Agile Modelling are:

1. **Communication**. Models help us to communicate between the team and project stakeholders and also between developers in the team to the stakeholders.

2. **Simplicity**. It’s very important for developers why the models are essential for simplifying both software and the software process.

3. **Feedback**. By using models/diagrams we can effectively communicate our ideas through, and can get the feedback easily and quickly.

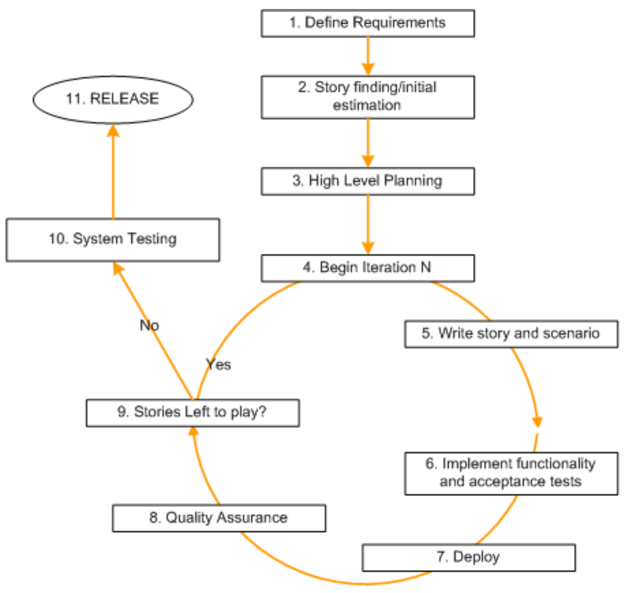


Fig: Agile Development

4. **Courage**. It’s very important because we need to take important decisions and be able to change the way by either refactoring or discarding our work, if anything goes wrong.

5. **Humility**. Respect every individual’s idea involved with the project has equal value and they have their own areas of expertise and have value to add to a project.

10 Key Principles of Agile

1. Active user involvement is imperative.
2. The team must be empowered to make decisions.
3. Requirements evolve but the timescale is fixed.
4. Capture requirements at a high level; lightweight & visual.
5. Develop small, incremental releases and iterate.
6. Focus on frequent delivery of products.

7. Complete each feature before moving on to the next.
8. Apply the 80/20 rule.
9. Testing is integrated throughout the project lifecycle – test early and often.
10. A collaborative & cooperative approach between all stakeholders is essential.

With Agile Methodology

- In Agile approach, at first the project is divided into a number of Iterations.
- Each iteration should be developed within the same.
- A working product with the expected functionality will be delivered at the end of each iteration.
- Rather than taking more time to requirements collection, in Agile, the project team will identify the basic features of the product that can be met with and will plan what features can be developed in the first iteration.
- Test frequently each iteration for making sure of no defects.
- Follows Collaborative approach

When to use Agile model:

- The agile gives us freedom to change. Changes can be implemented at very low cost because of the increments which are produced.
- To implement a new feature the developers need to work for few days, or even only hours sometimes.
- In agile model limited planning is required to initiate the project unlike the waterfall model. Agile accepts the end user's needs are continuously changing in a dynamic business world. Changes can be discussed and features can be either added or removed based on response. This can deliver an effective product to the customer what they require or desire.

Types of Agile Methodologies
A variety of agile methodologies that can have the same characteristics, but differs in implementation point of view, each has its own practices and vocabulary. Here are a few types:

1. **Scrum**
Scrum is one of the Agile movement. In Scrum, projects are divided into succinct work cadences, known as sprints, which are typically one to three weeks in duration. At the end of each sprint, stakeholders and team members meet to assess the project's advancement and plan its next steps. This allows project's direction to be adjusted or reoriented based on the work completed, not speculation or predictions.

Scrum Roles
Scrum has three roles:
Product Owner, Scrum Master, and Team.

- **Product Owner:** The Product Owner is a person with vision, authority, and availability. He is responsible for constantly communicating the vision and priorities to the development team. He must also represent the customer's interests through requirements and priorities.

- **Scrum Master:** The Scrum Master acts as a medium between the Product Owner and the team. He will never manage the team. Instead, he works to remove any impediments that are obstructing the team from achieving its sprint goals. This helps the team remain creative and productive while making its successes visible to the Product Owner. The Scrum Master also works to advise the Product Owner about how to maximize ROI (return on investment) for the team.

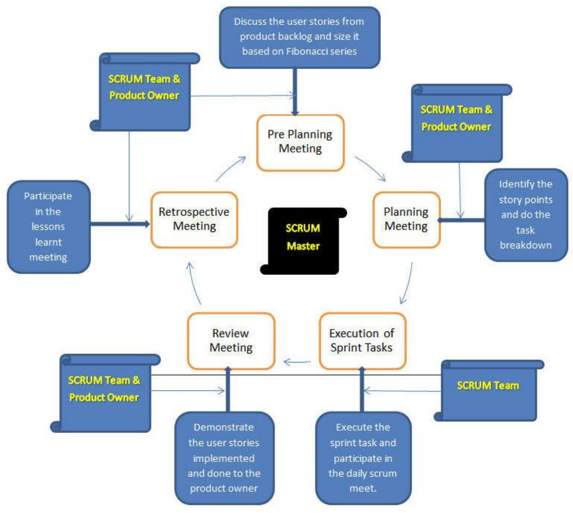

Discuss the user stories from product backlog and size it based on Fibonacci series
SCRUM Team & Product Owner
Pre Planning Meeting
SCRUM Team & Product Owner
Participate in the lessons learnt meeting
Retrospective Meeting
SCRUM Master
Planning Meeting
Identify the story points and do the task breakdown
Review Meeting
Execution of Sprint Tasks
SCRUM Team & Product Owner
SCRUM Team
Demonstrate the user stories implemented and done to the product owner
Execute the sprint task and participate in the daily scrum meet.

Fig: Scrum Process

- **Team:** In the Scrum methodology, the development team is responsible for self organizing to complete work. A Scrum development team contains about seven cross-functional members, plus or minus two individuals. For any software projects, a team includes a mix of analysts, software engineers, UI designers, architects, programmers, testers and QA experts, and. To each sprint, the team is responsible for determining how to complete the work. The team has a good independence and responsibility to meet the goals of the sprint.

EXTREME PROGRAMMING (XP)

XP is also one of the most popular agile methodologies. XP is a well-organized approach to deliver high-quality software quickly and continuously and openness to changing customer needs. It promotes high customer involvement, rapid criticism, constant testing and planning, and close teamwork deliver working software at very frequent intervals, usually every 1-3 weeks.

The XP is based on four simple values – simplicity, communication, feedback, and courage and twelve supporting practices:

1. Planning Game
2. Small Releases
3. Customer Acceptance Tests
4. Simple Design
5. Pair Programming
6. Test-Driven Development

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

7. Refactoring
8. Continuous Integration
9. Collective Code Ownership
10. Coding Standards
11. Metaphor
12. Sustainable Pace

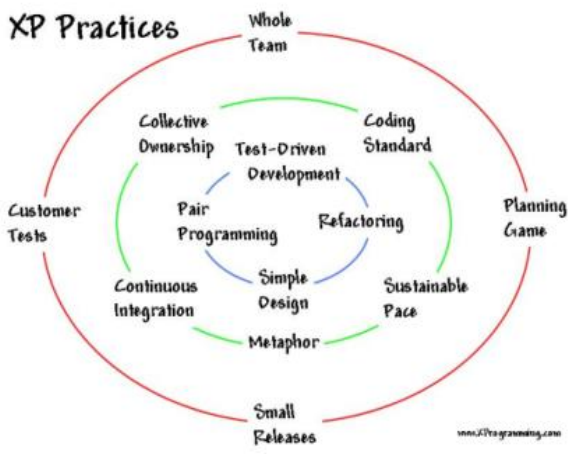
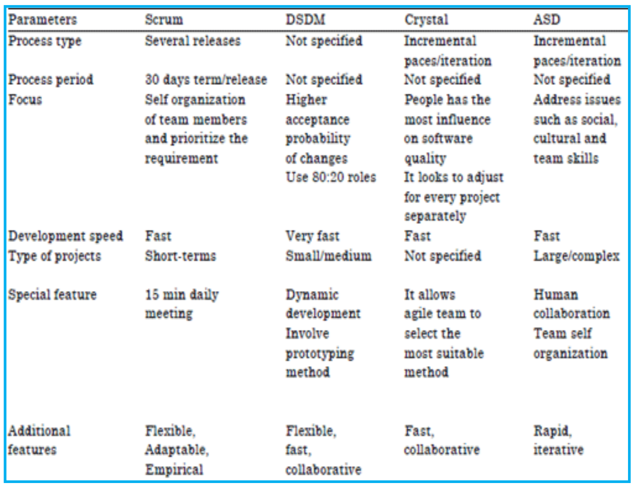


Fig: Extreme Programming Practices (XP)

In XP, the "Customer" communicates directly with the development team to describe and prioritize granular units of functionality referred to as "User Stories" need the software should fulfil. User stories help the team to estimate the time and resources necessary to build the release and to define user acceptance tests.

To create a release plan, the team breaks up the development tasks into iterations. The release plan defines each iteration plan, which drives the development for that iteration. At the end of an iteration, users perform acceptance tests against the user stories. If they find bugs, fixing the bugs becomes a step in the next iteration. Iterative user acceptance testing can result in release of the software. To guarantee high quality software and to get the better yielding, some supportive, trivial framework should be there to guide the team.



Parameters	Scrum	DSDM	Crystal	ASD
Process type	Several releases	Not specified	Incremental paces/iteration	Incremental paces/iteration
Process period	30 days term/release	Not specified	Not specified	Not specified
Focus	Self organization of team members and prioritize the requirement	Higher acceptance probability of changes Use 80:20 roles	People has the most influence on software quality It looks to adjust for every project separately	Address issues such as social, cultural and team skills
Development speed	Fast	Very fast	Fast	Fast
Type of projects	Short-terms	Small/medium	Not specified	Large/complex
Special feature	15 min daily meeting	Dynamic development Involve prototyping method	It allows agile team to select the most suitable method	Human collaboration Team self organization
Additional features	Flexible, Adaptable, Empirical	Flexible, fast, collaborative	Fast, collaborative	Rapid, iterative

Table: Summarizing and comparison of different agile techniques

DSDM: Dynamic system development method, XP: Extreme programming, ASD: Adaptive software development.

BENEFITS OF AGILE METHODOLOGY

- Customer satisfaction with quick, unremitting delivery of software.
- Rather than process and tools people and interactions are emphasized
- Clients, designers, developers and testers continuously interact with each other.
- Working software is delivered frequently.
- Easily adapts to changing conditions.

LIMITATIONS OF AGILE METHODOLOGY

- As the requirements are changeable, it's difficult to predict the expected result.
- Difficult to estimate the effort required for the project at the initial stage of the software development.
- Lack of importance on necessary designing and documentation.

CONCLUSION

Agile development methodology addresses the need of the today's business environments. The methods offer a fast development strategy for the software development houses with a higher level of the product's quality, performance and control. Furthermore, the agile method looks primarily to satisfy the customer. This is a direct goal of the development process. Agile method achieves customer satisfaction more than other traditional methods. It makes the customer part of the team, so that the customer will be able to see the work's progress and he/she can be satisfied from early stages. There are some principles we should follow when we develop using the agile method. Implementation of these principles depends on the nature of the project.

Some of the agile methods that are already in existence represent agile principles in different ways for different type of views. The six agile methods include: Feature Driven Development (FDD), eXtreme Programming (XP), Adaptive Software Development (ASD), Crystal, SCRUM and Dynamic Software Development Method (DSDM). These methods share some of the practices and features but still represent a unique way of developing the system.

REFERENCES

[1] Agile Manifesto, 2001. Agile manifesto and principles: Manifesto for agile software development. http://agilemanifesto.org.
[2] Pressman, R.S., 2005. Software Engineering: A Practitioners Approach. 6th Edn. McGraw Hill, New York.
[3] http://agilemethodology.org/
[4] http://www.allaboutagile.com/
[5] B. Boehm, Get ready for agile methods, with care, IEEE Computer 35 (1) (2002) 64–69.