

# Popular Agile Methods in Software Development: Review and Analysis

R. Vijay Anand, Dr. M. Dinakaran

School of Information Technology and Engineering, VIT University, Vellore-632014, Tamil Nadu, India.

**Abstract**—Software development plays a very major role in this industrial world. Previously organisations used to follow traditional software models but now there is a shift to agile methodologies. The principles and methodologies of Agile Software development have gained a lot of popularity in these recent times. In this paper, a comparison of six different agile methodologies has been made based on some characteristics. The objective of this paper is to provide a clear understanding and complete overview of Agile software development culture and practices. The main aim is to help the project managers or software engineers in choosing out the best agile methodology that suits their project based on their needs and requirements..

## I. INTRODUCTION

Agile organizations can more easily adapt their project portfolios due to changing business priorities. Agile projects deliver working software on a regular schedule, typically weekly or biweekly. The features delivered within each "iteration" are determined by the business[1], which will typically prioritize the highest value features first. These practices combine to reduce the risk of shifting resources away from an agile project. Because there isn't much half-done work at any point, the work that is finished is high value, and the state of the project is always unambiguous.

In today's fast-paced business environment, companies are constantly pressured to adapt to changing market conditions. Companies that produce software increasingly turn to agile development practices to help them stay competitive. Agile processes are highly collaborative, iterative and all focused on the rapid and repeatable delivery of software. Many practices, such as pair programming and regular retrospectives[2], may appear on their surface to slow teams down. However, the benefits to organizations of "going agile" far outweigh the costs.

### Agile Methodologies

In this paper, we have taken six different agile methodologies that are mostly used and a comparison is made between them based on some characteristics. A neat and clean explanation of every methodology is given. The six agile methodologies that are chosen in this paper are listed [2] below.

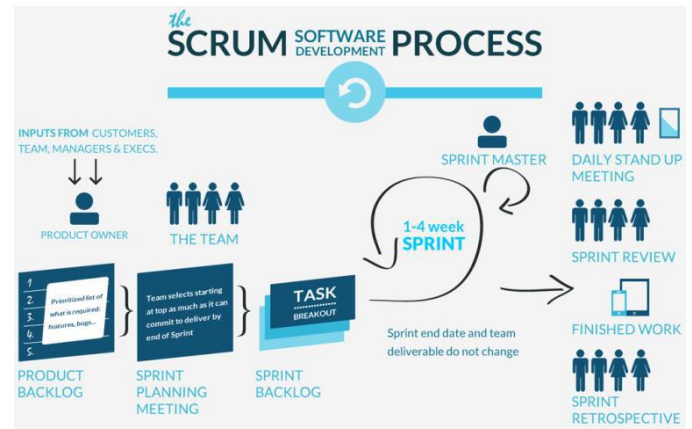
- Scrum
- Extreme Programming(XP)
- Feature Driven Development(FDD)
- Test Driven Development(TDD)
- Dynamic Software Development Method(DSDM)
- KANBAN

### Scrum

Scrum is one of the popular agile software development framework. Scrum is primarily for developing complex products and system. According to Ken Schwaber“Scrum employs an interactive and incremental approach to optimise

predictability and control risk”. There are mainly three people involved

1. product owner
2. scum master
3. project team or scrum team



### The Scrum Process

The process of scrum starts with the product owner [3]. The product owner gets the input from end-users, customers, stakeholders etc and come up with the product backlog/ list of features prioritised by the product owner in the sequence of business requirements and they are kept alive and owned by the product owners. Product owner constantly revises the product backlog and the only person who can make changes to it. From the product backlog, the team pulled something called Spring backlog and this happens for every sprint. Then team executes this building of product in sprints. Sprint is a fixed period of time that the team commits to work in a course of developing the product. The length of the Sprint is decided by the team and the product owner. During the sprint, what the team committed to deliver does not change and the end date of sprint doesn't change. If something major comes up, product owner can direct the team to terminate and start the new one. Scrum team ideally consist of 7+/-2 people (5-9 people). The team is self-organising and self-managing and has to be cross functional (containing members from different verticals required for developing the product). The team will select what

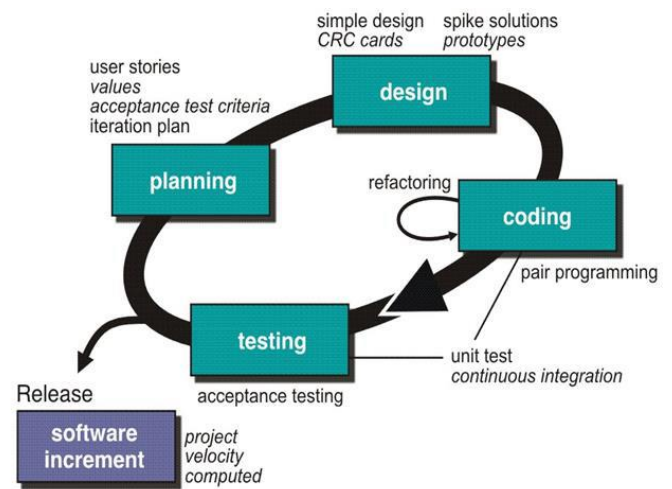
it will commit to deliver by the end of the Sprint. The team create a task level plan for how they will deliver. Everyone in the team takes part regardless of experience level. Each day team updates a simple chart called burndown chart that shows the task remind and total hours left for all tasks. Each day the team has a short meeting to update each other on progress and blocks. This is usually a casual and informal meeting of less than 15 minutes where everyone stand up to keep it fast and report just 3 things

1. done since yesterday
2. done by tomorrow
3. blocks and obstacles

Scrum master makes a notice of these blocks/obstacles [4] and help the team to resolve them. The role of the scrum master can be played by an existing person such as former project manager or team member. He is responsible for protecting and serving a team. He helps the team to remove any impediment and protects the team from any internal or external interference and guides the team. Without a Scrum Master, the team as a high risk of failure. The aim for the team is to complete 100% of what they committed ideally an increment which should be potentially shippable for the project. Every potentially shippable increment must be fully implemented and tested with no major defects. At the end of every sprint, there are two reviews that takes place. First one called "Sprint review meeting" where product owner, team, scrum master, stakeholder, etc. come together and see the demo of working software and feedback is taken. The other one called "Retrospective meeting" primarily only for the team, product owner and scrum master. They meet at the end of each sprint to review their way of working and look for ways to improve their effectiveness. Sprint review is a "product review" and Retrospective is a "process review".

**Extreme Programming** It is one of the most popularly and widely used agile practices for developing a high quality software. It mainly focuses and concentrates on improving the quality of software and on improving the responsiveness to changing requirements. According to Kent beck "XP is a light weight methodology for small to medium sized teams developing software in the face of rapidly changing requirements". Team work is emphasized in [5]extreme programming. There are five essential ways in which the extreme programming improves the software project. They are Communication, Simplicity, feedback, Respect and courage. Extreme programmers constantly stay in contact with the customers and with their fellow programmers. They keep out their design as simple as possible and keep it clean. Feedback is taken starting from day one by continuously testing the software. System or the working software is delivers to customers as early as possible and implement the changes as suggested even late in the life cycles. Every small success deepens their respect for the unique contribution of each and every team member. Other elements of extreme programming include: programming in pairs or doing extensive code review, unit testing of all code, avoiding programming of features until they are actually needed, a flat management structure, simplicity and clarity in code, expecting changes in the

customer's requirements as time passes and the problem is better understood, and frequent communication with the customer and among programmers. The methodology takes its name from the idea that the beneficial elements of traditional software engineering practices are taken to "extreme" levels. As an example, code reviews are considered a beneficial practice; taken to the extreme, code can be reviewed continuously [6], i.e. the practice of pair programming.

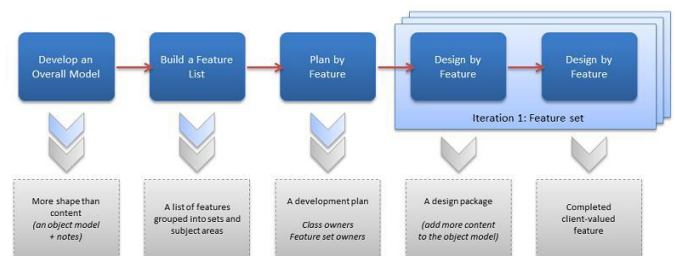


### Feature Driven Development

FDD is one of the agile software development process which uses a short-iterative and incremental model. It is one of the lightweight agile methodologies for developing a software. Combination of key benefits from various agile methodologies and other industry recognized best practices are provided by FDD. It is best suited to scale up larger projects and teams [7].

A feature is a small valued function expressed in client valued terms. If the given set of features are said to be small they can be completed within a two week time. If the features take more time to complete then it has to broken into smaller set of features.

The primary roles of FDD are Project Manager, Chief architect, Development manager, Domain experts, Class owners, Chief programmers.



These are the steps to be followed in the FDD process

**Step 1:** Develop an overall model (team members should gain knowledge and shared understanding of the problem domain)

**Step 2:** Build a features list (The features are to be organized with the hierarchy such as Business activity, Domain subject area, Features)

**Step 3:** Plan by feature (Constructing the initial schedule, Assigning responsibilities to team members)

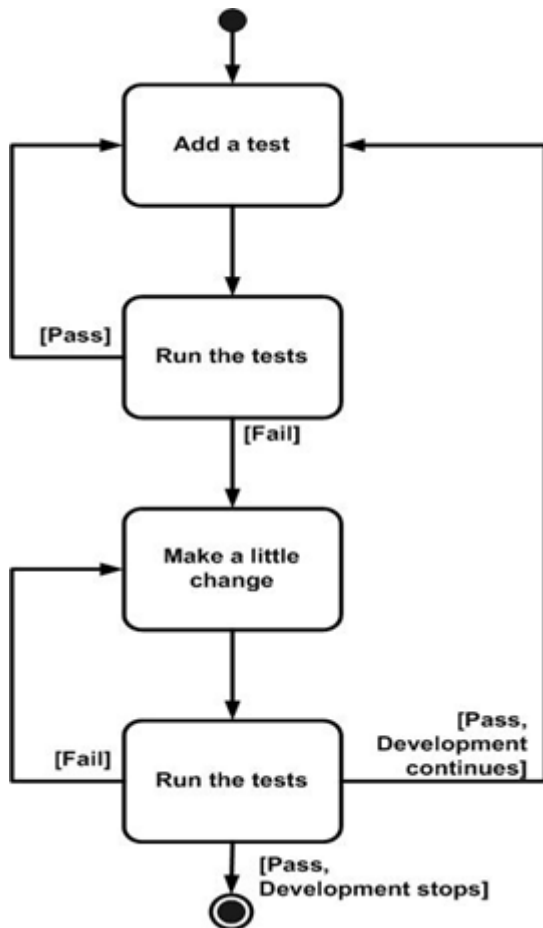
**Step 4:** Design by feature (Forming feature teams, team collaboration, updating the model Artifacts to support the change)

**Step 5:** Build by feature (Implementing the designed feature, Testing the feature, mandated code inspections)[8].

*Test Driven Development*

Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle [9]. The process will be carried out using some set of steps [10]:

1. Add a test
2. Run all tests and see if the new one fails.
3. Write some code.
4. Run tests
5. Refactor code.



*Dynamic Software Development Method*

DSDM is an agile project delivery framework. It is a RAD approach to the software development with iterative and incremental methodology including customer involvement. In 2007, DSDM became a generic framework for project management and solution delivery [11].

The public and private sectors came with an aim to develop the industry standards to develop the new method.

DSDM is the longest established full project agile approach which is tried and tested for business related projects.

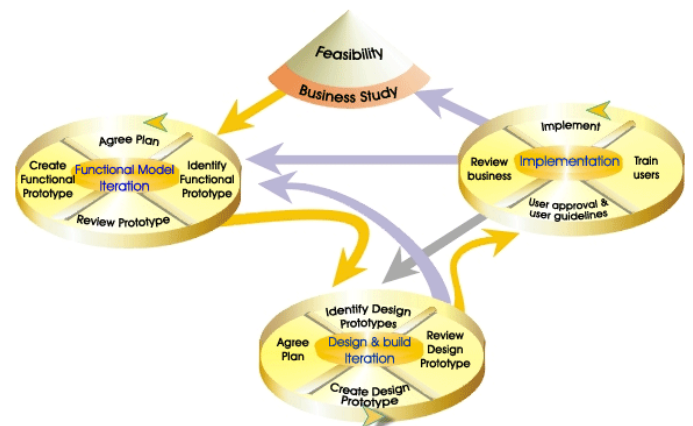
Some of the techniques of DSDM are MoSCoW, Timeboxing, Prototyping, Testing, Configuration management.

DSDM follows some of the principles in its development process such as follows [12]:

1. Focusing on the customer needs
2. Active user involvement
3. Rather than maximizing the quality focus on giving frequent releases.
4. If any solution to be re corrected, the iterative development is essential.
5. Testing can occur throughout the project.
6. Every stakeholders much get involved and communicate.

These are the steps to be followed in the DSDM process:

- Step 1:** Feasibility study
- Step 2:** Functional model iteration
- Step 3:** Design and build iteration
- Step 4:** Implementation



*Kanban*

Kanban is a method for managing knowledge work with an emphasis on just in time delivery while not overloading the team members. Development process is transparent in this methodology from definition of a task to its delivery to the customer, is displayed for participants to see[13]. The Kanban Method is as an approach to incremental, evolutionary process and systems change for organizations. It uses a work-in-progress limited pull system as the core mechanism to expose system operation (or process) problems and stimulate collaboration to continuously improve the system. Visualisation is an important aspect of Kanban as it allows to understand the work and the workflow. Kanban method consists of four basic principles : Start with existing process

The Kanban method does not prescribe a specific set of roles or process steps. The Kanban method starts with existing roles and processes and stimulates continuous, incremental and evolutionary changes to the system[14]. The Kanban method is a change management method.

**1. Agree to pursue incremental, evolutionary change** The organization (or team) must agree that continuous, incremental

and evolutionary change is the way to make system improvements and make them stick.

**2. Respect the current process, roles, responsibilities and titles** It is likely that the organization currently has some elements that work acceptably and are worth preserving. The Kanban method seeks to drive out fear in order to facilitate future change. It attempts to eliminate initial fears by agreeing

to respect current roles, responsibilities and job titles with the goal of gaining broader support.

**3. leadership at all levels** Acts of leadership at all levels in the organization, from individual contributors to senior management, are encouraged.

Comparison of Agile Methodologies:

Characteristic	SCRUM	XP	FDD	TDD	DSDM	KANBAN
Approach of Development	Iterative and incremental	Iterative and incremental	Iterative and incremental	Iterative	Iterative and incremental	Incremental
Time period for one iteration	2-4 weeks	1-6 weeks	<2 weeks	Reasonable Amount of Time based On product Built	2-3 days	Time taken From the start of work until Bugfix is live
Team Size	7+/-2	<20	Many Members (More than One team)	First 2/3 developers Get used to TDD. Later members are added accordingly	Independent Teams of any size	Any skilled or cross-functional team can work
Project Size	All types of projects	Small Projects	Complex projects	Small scale projects	All types of projects	Small projects usually with < 3000 LOC
Team Communication	Informal	Informal	Based on Documentation	Open communication b/w managers and developers about issues	Based on documentation	Informal face to face
Customer Involvement	Product owner acts on behalf of customer throughout the process	Throughout the process	Customer through reports	Involved only at acceptance testing	Involve in frequent releases	Customer through incremental release

## II. CONCLUSION

Through this paper, a clean and neat analysis of six different agile methodologies that are mostly used is given. This would help out the project managers and software engineers to understand the key characteristics, culture and practices of agile methodologies. This in turn helps them to choose the best practices that suit their software project according to their need.

## REFERENCES

- [1] Campanelli AS, Parreiras FS. Agile methods tailoring-A systematic literature review 2015. The Journal of Systems and Software.2015 Dec; 110:85-100.
- [2] Othmane LB, Angin P, Weffers H, Bhargava B. Extending the agile development process to develop acceptably secure software. Dependable and Secure Computing.IEEE Transactions. 2014 Nov-Dec; 11(6):497-509. DOI:10.1109/TDSC.2014.2298011.
- [3] Vlaanderen K, Jansen S, Brinkkemper S, Jaspers E. The agile requirements refinery: Applying SCRUM principles to software product management. Information and Software Technology. 2011 Jan; 53(1):58-70.
- [4] Danevaa M, Veena EVD, Amrita C, Ghaisasb S, Sikkela K, Kumarb R, Ajmerib N, Ramteerthkarb U, Wieringaa R. Agile requirements prioritization in large-scale outsourced system projects: An empirical study. The Journal of Systems and Software. 2013 May; 86(5):1333-53.
- [5] Popli R, Chauhan N, Sharma H. Prioritizing user stories in Agile Environment. International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT). 2014 Feb 7-8. p. 515-19. DOI:10.1109/ICICT.2014.6781336.
- [6] Shrivastava SV, Rathod U1. Categorization of risk factors for distributed agile projects. Information and Software Technology.2015 Feb; 58:373-87.
- [7] Ghosh GK. Challenges in Distributed Scrum.IEEE Seventh International Conference on Global Software Engineering. 2012. p. 200. DOI: 10.1109/ICGSE.2012.46.
- [8] Parsons D, Susnjak T, Lange M. Influences on regression testing strategies in agile software development environments. Software Qual J. 2014 Dec; 22(4):717-39. DOI: 10.1007/s11219-013-9225.
- [9] Anita, Chauhan N. A Regression test selection technique by optimizing user stories in an agile development. IEEE International Advance Computing Conference (IACC). 2014 Feb 21-22; 1454-58. DOI:10.1109/IAdCC.2014.6779540.
- [10] Ekbal R, Srikanta P, Vandana B. Estimation and Evaluation of Change in Software Quality at a Particular Stage of Software Development. Indian Journal of Science and Technology.2013 Oct; 6(10).
- [11] Akif R, Majeed H. Issues and challenges in Scrum implementation. International Journal of Scientific and Engineering Research. 2012 Aug; 3(8):1-4.1. ISSN 2229-5518.