Adoption of Agile Methodology in Software Industry

Satyavan Singh[1], Nitin Kumar[2] and Varun Bansal[3]

[1] Assistant Professor, Department of Computer Science & Engineering, Shobhit University, Gangoh, Saharanpur, Uttar Pradesh, India

[2] Assistant Professor, Department of Computer Science & Engineering, Shobhit University, Gangoh, Saharanpur, Uttar Pradesh, India

[3] Assistant Professor, Department of Computer Science & Engineering, Shobhit University, Gangoh, Saharanpur, Uttar Pradesh, India

[1]*satyavans8@gmail.com*, [2]nitin.ch.45@gmail.com, [3] bunty_varun@yahoo.com

Keywords- XP(Extreme Programming), AUP(Agile Unified Process).

Abstract- Agile methods provide a team or an organization with the flexibility to adopt a selected subset of principles and practices based on their values, their culture, and the systems that they develop. Agile software development these days becomes very popular. Webster university has started a graduate level course on Agile software development in this course main idea is discussed and implemented via student projects Traditional software development methods have many problems and limitations. To overcome the problems addressed in traditional development methods, with great advancement in development process and quality of software, agile methodologies were introduced. Traditional methods depend on document on records. The aim of agile processes is to satisfy the customer and faster development.

1. Introduction

The agile methodology provides an organization or team, the flexibility to adopt a selected subset of principles and practices based on their needs. Agile methodologies are used to produce higher quality software in a shorter period of time. Agile methodologies streamline the development process and remove challenges to accept business requirement changes during the development process. Agile methodologies do not need that customer requirements and design details be engaged for the duration of development. Agile SDMs have several features including iterative development, prototyping and minimal documentation. Agile software development is an iterative and time boxing approach to software development. Iteration can be of 1 week, two weeks or three weeks but rapid releases should be delivered to the customers. They include Scrum, XP, feature-driven

development, and others. Agile methods using various tactics, try to overcome the limitations of the systems development projects.

Paper starts with describing Agile, Lean and Scrum methodology discussing the difference between traditional methods and new approach Agile in software development.
Second chapter deals Assessment of agile methodology, the third chapter deals with implementation of agile methodology.
The final chapter deals with future work and used practices while applying this new way of working.

ABOUT AGILE METHODOLOGIES

A classically linear and sequential approach to software design and systems development is the waterfall model, where the development starts from analysing and defining the requirements and ends to operating maintaining the software. Actual coding is only a minor part of the entire process, whereas there is much emphasis on defining, designing, documentation, testing and maintenance the software system.
Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

Manifesto [5] for Agile Software Development

- Individuals and interactions over processes and tools,
- Working software over comprehensive documentation,
- Customer collaboration over contract negotiation,
- Responding to change over following a plan.

Agile and Lean are relatively broad concepts. Agile methods include Scrum, Extreme Programming (XP) and Agile Unified Process (AUP), among others.

A. Extreme Programming

Originally described by Kent Beck, has emerged as one of the most popular and controversial agile methodologies. XP is a disciplined approach to delivering high-quality software quickly and continuously. It promotes high customer involvement, rapid feedback loops, continuous testing, continuous planning, and close teamwork to deliver working software at very frequent intervals, typically every 1-3 weeks.
The original XP recipe is based on four simple values – simplicity, communication, feedback, and courage – and twelve supporting practices:

- Planning Game
- Small Releases
- Customer Acceptance Tests
- Simple Design
- Pair Programming
- Test-Driven Development

B. Scrum

Scrum is a lightweight agile project management framework with broad applicability for managing and controlling iterative and incremental projects of all types. Ken Schwaber, Mike Beedle, Jeff Sutherland and others have contributed significantly to the evolution of Scrum over the last decade. Scrum has garnered increasing popularity in the software community due to its simplicity, proven productivity, and ability to act as a wrapper for various engineering practices promoted by other agile methodologies.

C. Feature Driven Development

The FDD variant of agile methodology was originally developed and articulated by Jeff De Luca with contributions by M.A. Rajashima, Lim Bak Wee, Paul Szego, Jon Kern and Stephen Palmer,.

FDD is a model-driven, short-iteration process. It begins with establishing an overall model shape. Then it continues with a series of two-week "design by feature, build by feature" iterations. The features are small, "useful in the eyes of the client" results. FDD designs the rest of the development process around feature delivery using the following eight practices:

- Domain Object Modelling
- Developing by Feature
- Component/Class Ownership
- Feature Teams
- Inspections
- Configuration Management

2. ASSESSMENT METHODOLOGY FOR AGILE SOFTWARE DEVELOPMENT METHODS

There is a lack of comprehensive approach for assessing the agile methods. We assess the effectiveness of agile methods. Assessment is based on

1. Method`s effectiveness
2. Organisation support to implement the agile method (Capability)

3. Adequacy

We define these as below

1. Method`s effectiveness-Producing the intended or expected results. The existence of necessary process artefacts and product characteristics indicate levels of effectiveness.
2. Adequacy-Sufficiency of the method with respect to meeting its stated objectives.
3. Ability of an organization to provide an environment supporting the implementation of its adopted method. Such ability is reflected in the characteristics of an organization's people, process and project.

3. IMPLEMENTATION OF AGILE METHODOLOGY

Management teams have lot of discussion over the way to adopt the agile methodology. The methodology is accepted according to business environment, operations and requirement of customers.

We should start the journey towards becoming an agile software company in following steps

1. Low level planning we need a detailed plan to start an agile project we understand and solve all major problems this planning should be carried out into a strategic planning process and should be responsible for change
2. Acute learning
3. Innovation
4. Change
5. Agile Practices used

4. FUTURE WORK

We address the adoption of agile methodology from various perspectives and also address the assessment of agile methods from three perspectives – adequacy, capability and effectiveness. A comprehensive approach is applied to assessment of agile methods. A long term goal is to validate the framework and assessment process of framework.

REFERENCES

[1] Sheetal Sharma, Darothi Sarkar, Divya Gupta, “Agile Processes and Methodologies: A Conceptual Study” in (IJCSE) Vol. 4 No. 05 May 2012

[2] Shaweta Kumar, Sanjeev Bansal, “Comparative Study of Test Driven Development with Traditional Techniques” in (IJSCE) Volume-3, Issue-1, March 2013

[3] Felipe Carvalho, Leonardo Guerreiro Azevedo "Service Agile Development Using XP" IEEE Seventh International Symposium on Service-Oriented System Engineering, 2013

[4] Jeffrey A. Livermore " Factors that Impact Implementing an Agile Software Development Methodology" IEEE, 2007

[5] Sami Kollanus "Test-Driven Development - Still a Promising Approach?" Seventh International Conference on the Quality of Information and Communications Technology IEEE, 2010