Agile and understanding its principles

Introduction

In the world of software development and engineering, Agile method of development refers to a technique which was created in the 2000s in order to ensure that new products do not fail. It is considered to be one of the most versatile methods of software development and is used not only in the field of technology, but also in the world of business in a huge amount. The process means software developments must be based on iterative techniques along with cross functional teams. According to Wikipedia, Agile software development comprises various approaches to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their end users.

The term Agile was made popular by a famous event on February of 2001, where the concept of Agile Manifesto first emerged. The reason for this term being popularized was the frustration of engineers due to the rapidly changing market and absence of a unified development technique.

The Principles of Agile

The signatories of the famous 2001 event settled on the following points:

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

These four values of Agile form the 12 principles of software development

1. Customer satisfaction by early and continuous delivery of valuable software.
2. Welcome changing requirements, even in late development.
3. Deliver working software frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the primary measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Best architectures, requirements, and designs emerge from self-organizing teams
12. Regularly, the team reflects on how to become more effective, and adjusts accordingly

Understanding Agile further

The concepts of Agile rest on three major aspects, i.e., iterations, increment and evolution. Defining these in the terms of development, the process has to be iterative since the first version code might not meet the entire requirements of the project. As long as the product development is iterated more changes can be brought forth which further polishes it. Unlike linear methods of development, the iterative process has been more efficient regarding dynamic development.

The Agile factors of development allow the works to be segmented in small portions and with each iteration, there can be an increment of work. This allows the expertise to be utilized across all segments and progress is visible. Due to such segmentation, the changes in the final product is swift and smooth. Since the final goal of the Agile to find a working software/output, this technique is the best.

The Agile Philosophy

Agile has distinct purpose across its dimension. It is to achieve the best possible outcome. In the practical side of things, the processes of a development flow are divided into two categories: one being the adaptive side and another being the predictive one. The Agile principles fall on the adaptive side of things which according to its value can allow the product to be evolved based on the current circumstances.

One of the key adaptive development techniques is the use of rolling wave and also the Burnup chart. The chart shows the dynamics of created for client value in PM simulation.



Adoption of Agile

The method of Agile is mostly used in OOP environments. The initial users of the technique were SMEs working with requirements using languages like Java and Lisp. To use the methods of Agile, there are several tools and practices defined as:

1. Internal Assessments
2. Public Surveys

What are the cons of Agile?

1. Lack of overall product design
2. Adding stories to an iteration in process
3. Lack of sponsor support
4. Insufficient Training
5. Product owner role is non filled properly
6. Unfocused teams
7. Extensive planning and preparation
8. Problem solving in daily standup techniques like SCRUM
9. Assignment of tasks
10. Scrum master as a contributor
11. Lack of test automation
12. Allowing technical debt build up
13. Doing much within an iteration
14. Fixated time and resources
15. Fix quality
16. Developer burnout