

AGILE METHODOLOGY

In terms of software development basically means “the ability to respond to changes i.e. changes from requirements technology and people”.

- It is an iterative and incremental process.
- It offers direct collaboration with their customers.
- Delivers multiple software increments.
- Here each iteration last from one week to three weeks.
- Engineering actions are carried out by cross functional teams(a team expert in their respective work assigned)
- A team of software developers publish the agile manifesto in 2001, highlighting the importance of the development team, accommodating changing requirements and customer involvement.
- The principles of agile methodology were defined by manifesto(a public declaration of policy and aims).

Values of Agile:

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

Principle of Agile Process Models:

Following are the principles of agile process models,

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project. (here business ppl is those ppl who has invested in this project, and they daily have to collaborate with the developers so that the project will always up to the mark)

- Build projects around motivated individuals (stay away from demotivated individuals). Give them the environment and support they need and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation (to convey information regarding software, sending email shouldn't be our first priority, In agile face to face conversation is preferred).
- Working software is the primary measure of progress (we have to run the software to check whether it is working or not).
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. (sustainable development means if in one month 30% of the work has done so in the next month, we have to expect 30% work again to maintain the constant pace and if we are expecting less than the previous value that's not good for the project)
- Continuous attention to technical excellence and good design enhances agility (technical excellence and good design means software should be able to absorb/adapt requirement changes).
- Simplicity--the art of maximizing the amount of work not done--is essential (task should be defined in a simple way so that it can be done Which will minimize the amount of work).
- The best architectures, requirements, and designs emerge from self-organizing teams (each team should be self-organized and worked the same way as they planned).
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly (teams should collaborate and discuss the product's efficiency and how to improve product's quality).

Advantages:

- Deployment of software is quicker and thus helps in increasing the trust of the customer.
- Can better adapt to rapidly changing requirements and respond faster.
- Helps in getting immediate feedback which can be used to improve the software in the next increment.
- People – Not Process. People and interactions are given a higher priority rather than process and tools.
- Continuous attention to technical excellence and good design.

Disadvantages:

- In case of large software projects, it is difficult to assess the effort required at the initial stages of the software development life cycle (Difficult to predict how much this product will cost).
- The Agile Development is more code focused and produces less documentation.
- Agile development is heavily depended on the inputs of the customer. If the customer has ambiguity in his vision of the final outcome, it is highly likely for the project to get off track.
- Face to Face communication is harder in large-scale organizations.
- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence, it's a difficult situation for new programmers to adapt to the environment.

Example: Let's go through an example to understand clearly about how agile actually works.

A Software company named **ABC** wants to make a new web browser for the latest release of its operating system. The deadline for the task is 10 months. The company's head assigned two teams named **Team A** and **Team B** for this task. In order to motivate the teams, the company head says that the first team to develop the browser would be given a salary hike and a one-week full sponsored travel plan. With the dreams of their wild travel fantasies, the two teams set out on the journey of the web browser. The team A decided to play by the book and decided to choose the Waterfall model for the development. Team B after a heavy discussion decided to take a leap of faith and choose Agile as their development model.

The Development plan of the Team A is as follows:

- Requirement analysis and Gathering – 1.5 Months
- Design of System – 2 Months
- Coding phase – 4 Months
- System Integration and Testing – 2 Months
- User Acceptance Testing – 5 Weeks

The Development plan for the Team B is as follows:

- Since this was an Agile, the project was broken up into several iterations.
- The iterations are all of the same time duration.
- At the end of each iteration, a working product with a new feature has to be delivered.

- Instead of Spending 1.5 months on requirements gathering, they will decide the core features that are required in the product and decide which of these features can be developed in the first iteration.
- Any remaining features that cannot be delivered in the first iteration will be delivered in the next subsequent iteration, based in the priority
- At the end of the first iterations, the team will deliver a working software with the core basic features.

Both the team have put their best efforts to get the product to a complete stage. But then out of blue due to the rapidly changing environment, the company's head come up with an entirely new set of features and want to be implemented as quickly as possible and wanted to push out a working model in 2 days. Team A was now in a fix, they were still in their design phase and did not yet started coding and they had no working model to display. And moreover, it was practically impossible for them to implement new features since waterfall model there is not reverting back to the old phase once you proceed to the next stage, that means they would have to start from the square one again. That would incur them heavy cost and a lot of overtime. Team B was ahead of Team A in a lot of aspects, all thanks to Agile Development. They also had the working product with most of the core requirement since the first increment. And it was a piece of cake for them to add the new requirements. All they had to do is schedule these requirements for the next increment and then implement them.