**NIKITA**

**2017UCO1611**

**SCRUM**

Scrum is a framework that helps teams work together. It encourages teams to learn through experiences, self organize while working on a problem and reflect their wins and losses for continuous improvement.

Scrum relies on **self-organizing** which means: there is no overall team leader who decides for everyone and the problem to be solved. These issues are decided by the team as a whole. And a **cross-functional** team which means: everyone is needed to take a part from idea to implementation of the project.

In Scrum, instead of providing a complete and detailed description of how everything is to be solved and done on solving a problem, almost everything is left up to the team.

*Process of Scrum* :

According to Scrum, projects progress via a series of **sprints**. At the start of a sprint meeting team members figure out how many items they can commit to and then create a **sprint backlog**. The decided work is done, tested and integrated into the evolving product.

On each day of the sprint, all team members should attend a daily scrum meeting. During the meeting, team members share the work they have done, will do on that day and identify the obstruction to the progress of the project. At the end of the meeting, reviews are also taken from scrumMaster and Product Owner.

The meeting is an opportunity to reflect on the sprint and identify the ways and opportunities to improve.

*Pros :*

* Increase the quality of projects to be delivered.
* Coping from the changes in a better way.
* Providing better estimates while spending less time in organising them.
* Be more in control of the project schedule and state.

**LEAN DEVELOPMENT**

Elimination of waste is the main principle of lean development.

Lean software development is an agile framework based on optimising development time and resources, eliminating waste, and ultimately delivering only what the product needs.

Also known as Minimum Viable Product (MVP).

In MVP, the team releases a basic version of it’s product to the market and takes feedback from the users about the product and then iterates back on the given feedback.

The structure of lean products is iterative as it follows agile methodology.

Developers pay great attention to testing as their main focus is to deliver high quality products.

*Pros :*

* The approach followed allows more functionality to be delivered in less time.
* Unnecessary activities are eliminated and thus the cost can be reduced.
* It gives power to the development team to take decisions which hereby boost their morale.

*Cons :*

* Lean development heavily depends on the development team, reducing its scalability with respect to other frameworks.
* It depends on strong documentation and faults in the same can result in development errors and mistakes.

**EXTREME PROGRAMMING (XP)**

Extreme programming is an agile development methodology whose main aim is to improve software quality and its ability to properly adapt to the changing needs of the customers and clients.

Extreme programming aims to provide frequent releases throughout the life cycle of a project and follows an iterative approach allowing both customers and team members to examine and review the project’s process throughout the lifecycle of software development.

User stories are maintained as a mandatory part in XP as the whole development and process is done following these maintained user stories.

*Rules :*

* Planning
* Managing
* Designing
* Coding
* Testing

*Pros :*

* Saves time and cost required for project realisation.
* Whole process of XP is transparent and accountable.
* Constant feedback.
* Simple design

*Cons :*

* Doesn’t measure code quality assurance
* Defect documentation is not always good.
* It is not the best option if programmers are geographically distant.

**ADAPTIVE SOFTWARE DEVELOPMENT (ASD)**

It is a direct outgrowth of an earlier agile framework, Rapid Application Development (RAD).

Adaptive software development is a software development process whose principle and aim is to quickly and effectively adapt to changing requirements by evolving their products with lightweight planning and continuous learning.

It replaces the traditional waterfall cycle with a repeating series of three phase processes : speculate, collaborate and learn cycles.

It’s characteristics are : mission focused, feature based, iterative, risk driven and change tolerant.

*Pros :*

* Focused on the end users.
* Allows delivery to be on-time or even earlier.
* Encourages more transparency between clients and developer teams.

*Cons :*

* Demands excessive involvement of users.
* Integrates testing at every phase which increases the cost of the project.
* Focus on rapid iterating and continuous feedback can lead to deviation in the scope of the project to be developed.

**FEATURE DRIVEN DEVELOPMENT (FDD)**

FDD is an incremental and iterative software development process.

It is a lightweight or Agile method for developing software. It is customer centric with a goal of delivering efficient results often.

FDD encourages status reporting at all levels, which helps to track progress and results.

Features are a fundamental piece of FDD as user stories are to scrum.

FDD values documentation more than other methodologies, which also creates differences in meeting’s roles.

FDD is typically used in large-scale development projects. Some steps that are followed are:

* Develop the overall model
* Build feature list
* Plan by feature
* Design by feature
* Build by feature

*Pros :*

* Rapid development as it is a simple five step process.
* Larger teams allow them to move forward with continuous success.
* Pre-defined development standards, so development is quick.

*Cons :*

* Doesn’t work efficiently for smaller projects.
* Highly dependent on lead developers.