SCRUM

Scrum is a lightweight software development methodology that focuses on having small time-boxed sprints of new functionality that are incorporated into an integrated product baseline. Scrum places an emphasis on customer interaction, feedback and adjustments rather than documentation and prediction. Instead of phases, Scrum projects are broken down into releases and sprints. At the end of each sprint you have a fully functioning system that could be released. With scrum projects, the requirements for the project do not have to be codified up-front, instead they are prioritized and scheduled for each sprint. The requirements are composed of 'user stories' that can be scheduled into a particular release and sprint.

The Scrum team consist of:-

- Product owner = manages the product backlog and accepts completed increments of work.
- Scrum master = removes impediments, facilitates meetings, and helps the product owner groom the backlog.
- Development team = deliver shippable increments at the completion of each sprint.

Scrum events:-

- Sprint = time-boxed period during which specific work is completed and made ready for review. It is usually 2-4 weeks long but can be as short as one week.
- Sprint planning = time-boxed events that determine which product backlog items will be delivered and how the work will be achieved.
- Daily stand-up = short communication meeting (no more than 15 minutes) in which each team member quickly and transparently

covers progress since the last stand-up, planned work before the next meeting, and any impediments that may be blocking their progress.

- Sprint review = review delivery after completion of a sprint by product owner and stakeholders.
- The retrospective = final team meeting in the sprint to determine what went well, what didn't go well, and how the team can improve in the next sprint

Scrum artifacts:-

- Product backlog = document that outlines every requirement for a system, project, or product
- Sprint backlog = specific list of items taken from the product backlog which are to be completed in a sprint.
- Increment = sum of all product backlog items that have been completed since the last software release

Benefits of Scrum:-

- Transparency throughout the development cycle
- Quick releases keep the team motivated and users happy
- Rules, artifacts, events, and roles are easy to understand

KANBAN

Kanban is a visual system for managing work as it moves through a process. It visualizes both the process (the workflow) and the actual work passing through that process. The goal of Kanban is to identify potential bottlenecks in the process and fix them so work can flow through it cost-effectively at an optimal speed or throughput. It allows the software to be developed in one large development cycle. It controls the limited number of tasks active at any one time. Kanban is applicable in situations where work arrives in an unpredictable fashion and/or when you want to deploy work as soon as it is ready, rather than waiting for other work items.

Core principles of Kanban methodology:-

- Initiate with the existing workflow
- Limit the existing tasks
- Respect existing roles and responsibilities
- Encourage leadership at all levels

Core practices of Kanban methodology:-

- Visualization of workflow
- Reduction of Work In Progress
- Efficient workflow management
- Explicit management policies
- Take feedback

Benefits of using Kanban:-

- Enhanced flexibility
- Continuous improvement

Agile Unified Process

The Agile Unified Process (AUP) is the agile version of the Rational Unified Process (RUP).

AUP is an iterative-incremental process consisting of following disciplines:-

- Modeling = know their specific problem domain in order to identify a viable solution
- Implementation = model (or models) is written out into a code. The code is then executed and tested using a basic level of testing (i.e. unit testing) only.
- Testing = perform objective evaluations to validate that the code works, find defects and flaw, verify that the project requirements are met
- Deployment = plan the system delivery to ultimately make the software available to your end users.
- Configuration Management = manage your team's access to project artifacts to track the changes and versions made to the project in question
- Project Management = manage risks, assign tasks, track progress, and coordinate with the necessary parties and systems to ensure deliverables meet the deadline and budget.
- Environment = supporting the collaborative efforts of your development teams by making sure the tools, standards, and guidelines are immediately available for them as needed.

Principles of AUP:-

- Workers know what they are doing, because people do not need to

read the detailed documentation about the processes, but they should be given some simple design

- Ease of description, which means that everything is described concisely using smaller instructions, instead of manual
- Acceleration is one of the basic principles of this process
- Focus on important activities and not on others that may have occurred during the project
- Independence on the tools you use enables you to use any of the tools, those that are best suited for a particular job
- Adaptability of AUP model to the development team needs

FEATURE DRIVEN DEVELOPMENT

Feature-driven development (FDD) is an agile framework that organizes software development around making progress on features. FDD is customer-centric, incremental, and iterative to deliver tangible software results often and efficiently. It employs status reporting at all levels, thus helps to track results and progress. It allows teams to update the project regularly and identify errors quickly.

Five steps of FDD methodology:-

- Developing an overall model = The overall model is created by the chief architect, or another professional leading the project, by identifying the scope and context of the system.
- Building a feature list = Developers brainstorm a list of potential items that would be useful to users and could be completed along a set timeline for release. Each feature should be manageable within a timeframe of around two weeks.
- Planning by the feature = Features are organized by how long they take to create and how important they are to the client. Ownership of each feature is also assigned.
- Designing by the feature = The actual details of each feature are produced, inspected and finalized.
- Building by the feature = After the design is improved, the completed

feature is added to the official build for delivery to the client.

Benefits:-

- Enables larger teams to move products forward rapidly with continuous success.
- Efficient team communication
- Allows teams to regularly keep their projects up-to-date, observe any errors, and provide users/clients with valuable information at any time.
- The five well-defined methods make it easier for new team members or new hires to work on the project quickly.