**AGILE SOFTWARE DEVELOPMENT**

Agile is an interactive approach to software development that helps teams to deliver faster results for their customers with fewer obstacles. Following are the 4 key pillars that holds the Agile software development upright.

**Agile Manifesto:**

* Individuals and interactions over process and tools.
* Working Software over comprehensive documentation.
* Customer collaboration over contract negotiation.
* Responding to change over following a plan.

**12 key principles of agile software development:**

* Early delivery of the software
* Embrace change
* Frequent delivery
* Cooperation
* Motivated individuals
* Face-to-face communication
* Working Software
* Sustainable development
* Technical brilliance
* Simplicity
* Self-organizing teams
* Reflection and adjustment

**SCRUM**

Scrum divides the development into short cycles called sprint cycles. It advocates daily team meetings for coordination and integration.

The roles of scrum are as follows:

* **Product Owner** (the client or the owner)
* **Scrum Master** (responsible for controlling, handling, communication and removing any barriers during scrum process)
* **Team Member** (includes designers, programmers, quality analysts, architects; usually consists of 4-7 members)

**Scrum Sprint:** A sprint is **a short, time-boxed period when a scrum team works to complete a set amount of work**.

**User Stories:**

A user story is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

Stories give the team an important context and associate tasks with the values those tasks bring. User stories usually facilitates the following benefits:

* Keep the focus on user
* Enable collaboration
* Drive creative solutions
* Keep the momentum on

User stories together form the Epic or the bigger project.

**Product backlogs and estimations:**

In agile development, the product owner is tasked with prioritizing the **backlog** - the ordered list of work that contains short descriptions of all desired features and fixes for a product.

Agile teams usually cover story points. Story points are the units of measure for expressing an estimate of the overall effort required to fully implement a product backlog item or any other piece of work. Teams assign story points relative to work complexity, the amount of work, and risk or uncertainty. Estimations may or may not always be accurate.

**Retrospectives** are a time for the team to incorporate insights from **past iterations** - including the accuracy of their estimates.

**Metrics:**

Agile metrics provide insight into productivity through the different stages of a software development lifecycle. This helps to assess the quality of a product and track team performance.

**Burndown Chart:** Burndown charts track the progress of development over a larger body of work than the sprint burndown, and guide development for the scrum team.

**Velocity:** Velocity is the average amount of work a scrum team completes during a sprint, measured in either story points or hours, and is very useful for forecasting. The product owner can use velocity to predict how quickly a team can work through the backlog, because the report tracks the forecasted and completed work over several iterations–the more iterations, the more accurate the forecast.

Metrics are just one part of building a team's culture. They give quantitative insight into the team's performance and provide measurable goals for the team. Listening to team’s feedback during retrospectives is equally important in order to deliver quality product and maintain development speed during the release process.