**Definition of done** is an agreed upon list of the activities deemed necessary to get a product increment, usually represented by a user story, to a done state by the end of a sprint. i.e. programming, creating test data, actually testing, ensuring it's deployable, documenting. (means product quality).

**Definition of Ready** involves creating clear criteria that a user story must meet before being accepted into an upcoming iteration.

**Proof of Concept** (**POC**) is a demonstration, the purpose of which is to verify that certain concepts or theories have the potential for real-world application. **POC** is therefore a prototype that is designed to determine feasibility, but does not represent deliverables.

**Extreme Programming (XP)** is an agile software development framework that aims to produce higher quality software, and higher quality of life for the development team.

**Continuous deployment** aims to reduce the time elapsed between writing a line of code and making that code available to users in production. To achieve continuous deployment, the team relies on infrastructure that automates and instruments the various steps leading up to deployment, so that after each integration successfully meeting these release criteria, the live application is updated with new code.

**Continuous Integration** is the practice of merging code changes into a shared repository several times a day in order to release a product version at any moment. This requires an integration procedure which is reproducible and automated.

An **epic** is a large user story.

**Given-When-Then** formula is a template intended to guide the writing of acceptance tests for a User Story: (Given) some context, (When) some action is carried out, (Then) a particular set of observable consequences should obtain.

**Incremental Developmen**t is when each successive version of a product is usable, and each builds upon the previous version by adding user-visible functionality.

"**Integration**" (or "integrating") refers to any efforts still required for a project team to deliver a product suitable for release as a functional whole.

An **iteration** is a timebox during which development takes place. The duration may vary from project to project and is usually fixed.

**Planning Poker:** An approach to estimation used by Agile teams. Each team member "plays" a card bearing a numerical value corresponding to a point estimation for a user story.

A **sprint backlog** is the subset of product backlog that a team targets to deliver during a sprint in order to accomplish the sprint goal and make progress toward a desired outcome.

“**Behavior Driven Development**” is a practice where members of the team discuss the expected behavior of a system in order to build a shared understanding of expected functionality.

“**Test-driven development**" is a style of programming in which three activities are tightly interwoven: coding, testing (in the form of writing unit tests) and design (in the form of refactoring)

A **unit test** is a short program fragment written and maintained by the developers on the product team, which exercises some narrow part of the product's source code and checks the results

**Usability testing** is an empirical, exploratory technique to answer questions such as "how would an end user respond to our software under realistic conditions?"