AGILE GLOSSARY OF TERMS

Absolute Estimation	An estimation approach that uses hours, days or weeks; often contrasted with relative estimating approaches.
Acceptance Criteria	Acceptance Criteria are often used as tests of the completeness or behavior of a feature being developed.
Acceptance Test Driven Development (ATDD)	An approach for testing that begins with the customer acceptance criteria.
Acceptance Testing	Tests that are performed by the end-users or client to determine whether an application or feature fulfills its purpose.
Adaptation	One of the 3 pillars of empiricism; adaptation is the changes that are adopted by the team based on what is learned in inspection.
Adaptive Approach	Often used in contrast to predictive approaches, adaptive approaches introduce flexibility and responsiveness to change.
Agile	Agile is a mindset and set of values that centers around people and using incremental and iterative steps to deliver value.
Agile Champions	Agile Champions are advocates for agile ways of working in an organization; they support change within the organization and remove impediments.
Agile Manifesto	Formally called the Manifesto for Agile Software Development, the Manifesto is a set of 4 Values and 12 Principles developed by thought leaders in 2001.
Agile Mindset	A flexible way of thinking that enables people to respond quickly and adapt to change.
Agile Principle	There are 12 Agile Principles that were created as part of the Agile Manifesto. Formally called the Principles behind the Manifesto.
Agile Software Development	A development approach where small, self-organizing teams leverage close collaboration and short delivery cycles to reduce cost and speed development.
Agile Transformation	The process of transitioning the process and culture of an organization away from traditional or waterfall to an approach based on agile principles and thinking.
Agile Values	The 4 Value statements that were part of the Agile Manifesto created in 2001 when agile was formally launched.
Agile Working Group	Advocates for agile ways of working in an organization; they support change within the organization and remove impediments. See also Agile Champions.
Artifacts	A term in Scrum for the 3 tools that support team development; Product Backlog, Sprint Backlog and Increment.
Automated Build	Automated builds include retrieval of source code, compilation into binary code, automated tests, and publishing the build to a common repository.
Automated Test	A key feature of XP, automated tests are unit level tests of functionality that are run on code check-in or a set times throughout the day.
Backlog	A prioritized list of items, features, or requirements that an agile needs to complete. See Product Backlog and Sprint Backlog.
Batch Size Reduction	Reductions in the size of work items. Small batches go through the system more quickly which reduces risks and speeds feedback and learning.
Bottleneck	A process or operation that has limited capacity and reduces the capacity of the entire chain of events.
Burndown Chart	A burndown chart is a visual tool for measuring and displaying team progress for completing a sprint or a release.
Burnup Chart	A burnup chart is a visual tool for measuring and displaying team progress; typically used for progress of a release.
Business Agility	The ability to compete and thrive by quickly responding to market changes and emerging opportunities with innovative solutions.
Business Value	The perceived worth of a backlog item or feature from the perspective of the customer.
Capacity	A measure of the scope of work an agile team can take on based on available team members and work days.
COD - Cost of Delay	The potential loss from not delivering value to the market at the right time
Collective Code Ownership	All code is jointly owned; any developer can change any line of code to add functionality, fix bugs, improve designs or refactor removing a potential bottleneck.
Collocated Team	Teams that can sit and work together in the same space.
Communities of Practice	A common interest group who collaborate to share knowledge and tackle challenges. Frequently used in agile when functional orgs migrate to cross-functional teams.
Continuous Delivery	A software development practice that leverages short cycles and small batches to release a steady stream of changes to production.
Continuous Deployment	A practice that builds on continuous delivery to push new features into production without human intervention.
Continuous Integration	A technical practice where new code changes are integrated into the main codebase in small batches rather than BIG ones. Typically includes automated testing for defects.
Cumulative Flow Diagram (CFD)	A chart showing the number of work items in various queues of development. Frequently used with Kanban.
Cycle Time	The time it takes for a team to deliver a work item to a customer once they begin working on it.
Daily Scrum	The Daily Scrum is a short meeting of the Developers of a Scrum team to inspect their progress toward the Sprint Goal; it is timeboxed at 15 minutes.
Daily stand-up | Daily meeting	A short daily meeting of a team where members standup so that the meeting is kept short.
Definition of Done (DoD)	The Definition of Done is a team quality standard or agreement on what constitutes done for a backlog item.
Definition of Ready (DoR)	A quality standard agree by the team and based on those characteristics of a backlog item that would make it ready to bring into a sprint to be worked on.
Developers	In the Scrum Guide, Developer is the generic name for a team member committed to creating a usable Increment each Sprint.
DevOps	A set of technical practices that remove boundaries between teams, shortens delivery cycles, and improves quality.
Disciplined Agile	Disciplined agile is an agile decision-making toolkit created by Scott Ambler and Mark Lines and acquired by PMI in 2019.
Distributed Teams	Teams that are not co-located are distributed or remote.
Dot Voting	A democratic technique to allow participants to choose from several alternatives.
Emergence	The process of the coming into existence or prominence of new facts or new knowledge of a fact, or knowledge of a fact becoming visible unexpectedly.
Empirical Process Control	An approach based on inspecting the results of the process and making regular adjustments. Often contrasted with predictive approaches.
Empiricism	Approach where decisions are based on observation, experience and experimentation rather than speculation. Relies on transparency, inspection and adaptation.
Epic	A term for a very large user story that is eventually broken down into smaller stories.
Extreme Programming (XP)	A lightweight agile approach based on a set of technical development practices popularized by Kent Beck in the 1990's.
Feature Driven Development	An agile framework that organizes software development around making progress on "features" which are similar to user stories.
Fibonacci Sequence	The sequence of numbers used for estimating in story points. The next number is derived by adding together the previous two (1,2,3,5,8,13,21...)
Flow Metrics	Metrics that measure the rate of business value delivery for software products through the lens of your customers.
Frequent Releases	A concept mentioned in the 12 Agile Principles that fosters short feedback cycles with customers.
Impediment	Things that slow team progress or prevent the team from meeting their goals.
Increment	The Increment is one of 3 artifacts in Scrum. It represents a valuable subset of the overall solution delivered by the Scrum Team in a Sprint.
Incremental Development	A method of developing solutions piece-by-piece. The system is broken down into small elements which are designed, built and tested independently.
Information Radiators	An up to date display of team progress posted in a visible place. Passers by can tell the status of the team without interrupting them.
Information Refrigerators	A display of team information that requires interested parties to rummage around looking for it, often in an online tool.
INVEST	An acronym used for characteristics of effective user stories; Independent, Negotiable, Valuable, Estimatable, Small and Testable.
Iteration	A timeboxed period (usually two weeks) in which a development team completes a set amount of work. In Scrum the iteration is called Sprint.
Iterative Development	An approach to delivering technical solutions that involves using improved versions of the solution. Often coupled with Incremental development.
Kaizen	Kaizen is a combination of two Japanese word that translate as good change. It has come to stand for "continuous improvement" in lean and agile ways of working.
Kanban	A card-based method of managing work using a visual board with columns/boards representing each stage of the work process.

Kanban Board	An information radiator used in Kanban to model the workflow of a group of people and the stages or queues that the work goes through.
Lean Software Development	An agile framework based on optimizing development time and resources, eliminating waste, and delivering only what the product needs.
Lean Thinking	A framework focused on organizing human activities to deliver more benefits to society and value to individuals while eliminating waste.
Minimum Marketable Feature (MMF)	A minimum marketable feature is the smallest set of functionality in a product that must be provided for a customer to recognize value.
Minimum Viable Product (MVP)	The smallest version of a product that has sufficient features to be usable by early customers to gain feedback and insights about customer needs.
Mob Programming	A software development approach where the whole team works on the same thing, at the same time, in the same space, and on one computer.
Nexus	An agile scaling framework developed by Ken Schwaber that includes up to 9 Scrum teams developing from a single product backlog.
Pair Programming	An Agile software development technique introduced in Extreme Programming in which two programmers work together at one workstation.
Pairing	Pairing is a variation of pair programming extended outside developers. People with various skills work together on one computer at the same time.
Participatory Decision-making	A democratic technique that gives ownership of decisions to the whole group, finding effective options that everyone can live with.
Personas	User archetypes that help teams have empathy for and understand the needs of end-users.
Planning Poker	A team-based technique for estimating based on relative size of backlog items.
Product Backlog	An emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the Scrum Team.
Product Backlog Item	A product backlog item is an individual feature or need that is included in the product backlog.
Product Backlog Refinement	The ongoing act of breaking down and further defining Product Backlog items into smaller more precise items. Formerly called backlog grooming.
Product Owner	Product Owner is a member of the Scrum team that represents the voice of the customer and is accountable for ensuring that the team delivers value.
Product Roadmap	A high-level plan that outlines the direction, priorities, and progress of a product over time, typically 4-5 quarters.
Program Increment Planning	Technique for planning multiple dependent teams delivering an increment of work. Also called Big Room Planning.
Pull System	A lean manufacturing strategy that relies on downstream pull signals to move materials thereby minimizing inventory, work in process, and waste.
Queue	A holding place for items as they wait for the next action in a work stream. Agile teams avoid queues by reducing batch sizes and addressing bottlenecks.
Ready	Similar to the Definition of Done, some teams use a Definition of Ready as a checklist to ensure items can be started and finished in the same sprint.
Refactoring	Refactoring is an XP technical practice. It is the process of improving software design, without changing the functionality.
Relative Estimation	An estimation approach that uses relative measure like story points or t-shirt sizes to make fast estimates that are precise enough.
Release	An incremental delivery of a product or solution; internal releases are for internal use only and external releases go to customers.
Release plan	The rough agile equivalent to a project plan showing the work items and sprints needed to deliver a release of your product or solution.
Release Train (RT)	Same as Agile Release Train in SAFe.
Release Train Engineer (RTE)	Similar to the Scrum Master in Scrum, the RTE is a servant leader and coach for a group of agile teams participating in a Release Train.
Retrospective	A Retrospective in Scrum is the last event to be held during a Sprint. Teams work together to identify and prioritize action steps they can take to improve their process.
Scaled Agile Framework (SAFe)	A scaling technique popularized by Dean Leffingwell that includes a set of organizational and workflow patterns for implementing agile practices at enterprise scale.
Scaling	Scaling in Agile is the process of translating established team-based agile approaches like Scrum and Kanban to larger groups of people.
Scrum	Scrum is a lightweight framework developed by Ken Schwaber and Jeff Sutherland that helps teams explore value through adaptive development.
Scrum Event	A Scrum Event is a formal opportunity to inspect and adapt Scrum artifacts. Previously called meetings, Scrum Events also include the Sprint itself.
Scrum Master	One of the accountabilities of the Scrum Team; a servant leader accountable for fostering Scrum and removing team impediments.
Scrum of Scrums	A technique popularized by Jeff Sutherland to scale Scrum to multiple teams working on the same product.
Scrum team	A cross-functional and self-organizing group responsible for delivering the product. The team includes all the skills needed to deliver end to end.
Scrum Values	Scrum values include Commitment, Focus, Openness, Respect, and Courage. These give direction to the Scrum Team with regard to their work, actions, and behavior.
Scrumban	Scrumban is the term used for teams that use elements of Scrum (the events) along with the Kanban approach.
Self-Organization	In agile, empowering teams to self-manage. A self-organizing agile team is solely responsible for assigning and tracking their own work and progress.
Spike	A specific type of user story that represents a short, time-boxed piece of research or a technical proof of concept.
Sprint	A fixed length timebox or iteration that serves as a container for all the Scrum events.
Sprint Backlog	The Sprint Backlog is an output of Sprint Planning, where the team forecasts the backlog items and tasks that they will complete during the sprint.
Sprint Goal	An overarching objective for the Sprint that communicates why the Sprint is valuable to stakeholders.
Sprint Planning	A Scrum event where the Scrum Team initiates the Sprint by laying out the work to be performed during the Sprint
Sprint Retrospective	The Retrospective is the Scrum event that happens at the end of every Sprint to review the team process and to discuss how the team can be more effective in the future.
Sprint Review	The Scrum event held at the end of each Sprint where stakeholders review and provide feedback on the product Increment.
Stakeholder	Those individuals who are impacted by or invested in the success or failure of a project. May be internal or external to the organization.
Story Points	Relative units of measurement used by some agile teams to quickly estimate the effort needed to complete items in the product backlog.
Sustainable Pace	An concept introduced in XP and memorialized in the agile principles whereby all team members and stakeholders work at a pace they can continue indefinitely.
Task	Tasks are an attribute of Product backlog items. Teams will generate tasks to complete the backlog items during Sprint Planning.
Taskboard	A visual representation of the work of an agile team. Originally this was a wall chart with cards and/or sticky notes; most teams today use online tools.
Team	An agile team is a cross-functional and self-organizing group responsible for end to end delivery.
Technical Debt	Technical Debt is the cost associated with maintaining code. It represents the overhead that results from poorly designed code that is risky or costly to maintain.
Three Questions	A prescribed check-in for the Daily Scrum, the questions are 1) what did I do since we last me, 2) what am I going to do next and 3) do I have any impediments.
Timebox	An agreed period of time, such as a Sprint or iteration, during which a person or a team works steadily towards completion of some goal.
Transparency	One of the 3 pillars of empiricism; transparency means everything is open and available for inspection. Nothing is hidden.
User Story	A simple expression of a business need, that describes the who, what and why of the need and serves as a placeholder for a future conversation.
User Story Mapping	A technique popularized by Jeff Patton that uses lightweight methods to map out the interactions that users to go through when using a product.
Value Stream Map	A tool from lean that involves charting out the flow of an existing process to identify bottlenecks and delays and to propose improvements.
Value Stream Mapping	A process allows you to create a detailed visualization (value stream map) of all the steps in your work process.
Velocity	Velocity is a relative throughput measure for a team which reflects the number of items or story points a team completed in a particular period.
Wait time	A lean concept, it represents steps in a process that don't add value since work items are simply waiting for someone to do something.
Waterfall	A popular project management approach that emphasizes a linear progression from beginning to end of a project.