Basics

Agile	An iterative and collaborative approach to project management and software development.
Backlog	A prioritized list of tasks, user stories, and features yet to be completed
Burndown Chart	A visual representation of the progress made by the team in completing tasks over time
Burnup Chart	A visual representation of the total work completed (scope) and the work remaining over time.
Agile Manifesto	A set of four values and twelve principles that define the Agile approach to software development.
Minimum Viable Product (MVP)	The minimum set of features required to deliver value to customers.
Agile Transformation	The process of transitioning an organization from a traditional, waterfall approach to Agile practices.
Agile Project Management	An approach that combines Agile principles with traditional project management practices.
Cross-functional Team	A team composed of individuals with different skills and expertise needed to deliver a product.

Frameworks & Setups

Scrum	An Agile framework that focuses on delivering value through small, self-organizing teams.
Kanban	An Agile method that focuses on visualizing work, limiting work in progress, and maximizing flow
Lean	A set of principles and practices that aim to minimize waste, maximize value, and improve efficiency
DevOps	A set of practices that combines software development (Dev) and IT operations (Ops) to streamline the delivery process.
Lean Startup	An approach that applies Lean principles to the development of new products, focusing on rapid experimentation and validated learning.
Scrum of Scrums	A meeting where representatives from multiple Scrum teams come together to coordinate their work.
Scaled Agile Framework (SAFe)	A comprehensive framework that provides organizations with a structured approach to scaling Agile practices across multiple teams, enabling efficient collaboration, alignment, and delivery of large-scale projects or products.
The Large-Scale Scrum (LeSS)	A lightweight, principles-based approach to scaling Agile, designed to enable organizations to scale up their Agile practices while maintaining simplicity, transparency, and a focus on delivering value to customers.

Frameworks & Setups (cont)

Nexus	A scaled Agile framework that provides guidance for organizations to effectively scale Scrum across multiple teams, emphasizing collaboration, integration, and continuous improvement to ensure successful delivery of complex products or solutions.
Scrum@Scale	A framework developed by Jeff Sutherland, the co-creator of Scrum, that enables organizations to scale the Scrum framework across multiple teams, departments, and even entire enterprises, fostering collaboration, adaptability, and effective delivery of value at scale.
OKRs (Objectives and Key Results)	A goal-setting framework that helps organizations align their efforts and track progress by defining clear objectives and measurable key results.

Roles

Product Owner	The person responsible for defining and prioritizing the product backlog.
Scrum Master	The facilitator of the Agile process who helps the team stay on track and resolve any issues.
Agile Coach	Coach: An experienced Agile practitioner who guides and supports teams in adopting Agile practices and principles.
Flow Manager / Service Delivery Manager	The role responsible for overseeing the end-to-end delivery of services or products using the Kanban methodology, ensuring customer satisfaction and alignment with organizational goals
Service Request Manager	Is responsible for receiving, prioritizing, and managing incoming service requests, ensuring timely resolution and customer satisfaction while adhering to the Kanban principles and practices.
Release Train Engineer (RTE)	Plays a critical role in facilitating the coordination and alignment of multiple Agile teams within a release train, ensuring smooth execution, timely delivery, and effective communication across all levels of the organization.
Chief Product Owner (CPO)	Is the senior-level role responsible for defining the strategic direction and vision of the product, overseeing multiple product owners, and ensuring alignment between the product roadmap, business goals, and customer needs.

Events & Meetings

Scrum: Sprint	A time-boxed period (usually 1-4 weeks) where a team works to complete a set of tasks.
Scrum: Daily Stand-up / Daily Scrum /	A short, daily meeting where team members discuss progress, challenges, and plans for the day.
Scrum: Retrospective	A meeting held at the end of each sprint to review what went well, what didn't, and how to improve.
Scrum: Sprint planning	A collaborative meeting where the Agile team determines which user stories will be worked on during the upcoming sprint and defines a plan to accomplish them.
Scrum: Sprint review	A meeting at the end of each sprint where the Agile team showcases the completed work to stakeholders, receives feedback, and discusses any changes or next steps.
Kanban: Commitment meeting	A collaborative session where the team establishes and agrees upon the amount and type of work they will commit to completing within a specific timeframe, based on their capacity and available resources.
Kanban: Kanban Meeting	A regular gathering where the team discusses the progress of work, identifies any potential bottlenecks, and makes adjustments to optimize the flow of tasks on the Kanban board.

Agile practices

Test-driven Development (TDD):	A development approach where tests are written before the code is implemented.
Pair Programming	A practice where two developers work together on the same code, sharing a single workstation.
Agile Estimation	The process of estimating the effort required to complete a task or user story.
Continuous Delivery	The practice of continuously deploying software changes to production in a safe and sustainable manner.
Continuous Integration	The practice of frequently integrating code changes into a shared repository to avoid conflicts.
Continuous Improvement	The ongoing effort to identify and implement changes that result in better processes, products, and outcomes.
Velocity-based Planning	A planning approach that takes into account the team's historical velocity to forecast future work.
Agile Release Planning	The process of identifying and prioritizing features for upcoming releases or iterations.

Agile practices (cont)

Incremental Delivery	The practice of delivering a working product increment to stakeholders at regular intervals.

Work Item related

Epic	A large, high-level user story that is too big to be completed within a single sprint.
User Story	A brief, high-level description of a desired feature or functionality from the user's perspective.
Acceptance Criteria	The specific conditions that a user story must meet to be considered complete.
Definition of Done (DoD)	A set of criteria that must be met for a product increment or user story to be considered complete.
Definition of Ready (DoR)	The criteria that a user story must meet before it can be considered ready for development.
Retrospective Action Items	Specific action items identified in the retrospective to address areas for improvement.