

A SPRINT TO AGILE

SIMPLE AGILE DICTIONARY

Roles & Team Structure

1. **Scrum Master:** A facilitator who ensures the team follows Agile principles and removes "blockers" that slow down progress. They don't manage the people, but they do manage the process and health of the team.
2. **Product Owner (PO):** The person responsible for maximizing the value of the product by managing the backlog. They represent the voice of the customer and stakeholders to the team.
3. **Developers:** The cross-functional members of the team who do the actual work of building the product increment. They are self-organizing and decide how to achieve the technical goals.
4. **Stakeholder:** Anyone outside the core team with an interest in the product, such as customers, users, or managers. They provide feedback during reviews to ensure the product meets business needs.
5. **Cross-functional Team:** A group of people possessing all the skills (design, code, test) necessary to create a working product. This reduces dependency on outside departments and speeds up delivery.
6. **Self-Organizing Team:** A team that manages its own work and determines the best way to accomplish its goals. They don't wait for a manager to assign specific tasks.

Core Frameworks & Mindsets

7. **Agile Manifesto:** The 2001 document that defined the four core values and twelve principles of Agile software development. It prioritizes individuals and interactions over tools and processes.

8. **Scrum:** The most popular Agile framework, which uses fixed-length iterations called Sprints to deliver software. It relies on specific roles, events, and artifacts.
9. **Kanban:** A visual system for managing work as it moves through a process. It focuses on continuous delivery and limiting the amount of work-in-progress to avoid bottlenecks.
10. **Lean:** A methodology focused on "eliminating waste" and delivering value as quickly as possible. It encourages making decisions as late as possible to stay flexible.
11. **XP (Extreme Programming):** An Agile framework focused on technical excellence and high-quality code. It introduces practices like pair programming and test-driven development.
12. **Empiricism:** The philosophy that knowledge comes from experience and making decisions based on what is known. Scrum is built on three pillars of empiricism: transparency, inspection, and adaptation.

The Sprint Cycle (Ceremonies)

13. **Sprint:** A fixed time-box (usually 1–4 weeks) where a "Done" and usable product increment is created. Every Sprint has a specific goal that the team commits to.
14. **Sprint Planning:** A meeting where the team selects items from the Product Backlog to work on during the upcoming Sprint. They determine the "What" (Sprint Goal) and the "How" (the plan).
15. **Daily Stand-up:** A 15-minute daily sync for the team to coordinate their work for the next 24 hours. Members share progress, plans, and any obstacles in their path.
16. **Sprint Review:** A demo session at the end of a Sprint where the team shows their work to stakeholders. It is a collaborative event used to gather feedback and adjust the future backlog.
17. **Sprint Retrospective:** A private team meeting to discuss what went well and what could be improved in the next Sprint. It focuses on people, relationships, and processes rather than the product.
18. **Backlog Refinement:** The ongoing process of adding detail, estimates, and order to items in the Product Backlog. This ensures that the top items are "Ready" for the next planning session.

19. **Time-boxing:** The practice of setting a maximum amount of time for a specific activity or event. It helps teams maintain focus and prevents meetings or tasks from dragging on indefinitely.

Artifacts & Documentation

20. **Product Backlog:** An ordered, living list of everything that might be needed in the product. It is the single source of requirements for any changes to be made.

21. **Sprint Backlog:** The set of items the team has committed to finishing in the current Sprint. It includes a plan for how to deliver the work and meet the Sprint Goal.

22. **Increment:** The sum of all the Product Backlog items completed during a Sprint. Each increment must be in usable condition and meet the team's quality standards.

23. **User Story:** A short, simple requirement told from the perspective of the end user. It usually follows the format: "As a [user], I want [action], so that [value]."

24. **Epic:** A large body of work that is too big to be completed in a single Sprint. Epics are eventually broken down into smaller, manageable User Stories.

25. **Task:** A technical "to-do" item required to complete a User Story. Tasks are usually small enough to be finished in a day or less.

26. **Acceptance Criteria:** A specific list of requirements that a User Story must meet to be considered functional. They act as the "checklist" for a story to be accepted by the PO.

27. **Definition of Done (DoD):** A shared agreement on the quality standards a task must meet before it is called "finished." It ensures that everyone understands what "complete" actually means.

Metrics & Visualization

28. **Kanban Board:** A visual tool used to track work items as they move through different stages (e.g., To Do, Doing, Done). It helps teams see bottlenecks and flow in real-time.

29. **Burndown Chart:** A graph showing how much work is remaining versus how much time is left in the Sprint. It helps the team visualize if they are on track to finish their commitment.

30. **Burnup Chart:** A graph that tracks the total work completed against the total scope of the project. It is useful for showing progress and changes in the project's total size.

31. **Velocity:** A measure of the amount of work a team can tackle during a single Sprint. It is used for long-term forecasting, not as a way to compare different teams.
32. **Story Points:** A unit of measure for expressing the overall size and complexity of a User Story. They are relative values rather than specific hours or days.
33. **WIP (Work in Progress):** The number of tasks the team has started but not yet finished. Agile encourages "limiting WIP" to prevent multitasking and speed up delivery.
34. **Cycle Time:** The amount of time it takes for a single task to move from "In Progress" to "Done." Measuring this helps teams predict how long future tasks will take.
35. **Lead Time:** The total time from when a request is made until it is delivered to the customer. It includes the time the item spent waiting in the backlog.

ENJOY THE GAME!