
Advanced Skill Certificate in Team Leadership for Software Development (United Kingdom)

Agile Team Dynamics and Facilitation

Agile Manifesto

Related terms: Agile Principles, Scrum, Kanban

The Agile Manifesto is a set of four values and twelve principles that guide agile software development. It emphasizes individuals and interactions over processes, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan.

***Example*:** A team adopts the manifesto by prioritising frequent releases that deliver usable features rather than waiting for a large, monolithic release.

***Practical application*:** Use the manifesto as a reference during sprint retrospectives to evaluate whether the team's behaviours align with the values.

***Challenges*:** Translating high-level values into day-to-day actions can be ambiguous; teams may revert to "waterfall-like" documentation habits when under pressure.

Agile Principles

Related terms: Agile Manifesto, Continuous Delivery, Iterative Development

These are the twelve guiding statements that expand the manifesto's values, such as delivering working software frequently, welcoming changing requirements, and maintaining a sustainable pace.

***Example*:** A product owner embraces changing priorities by re-prioritising the backlog after each sprint.

***Practical application*:** Review each principle during sprint planning to ensure the team's commitments reflect them.

***Challenges*:** Teams often struggle to balance "continuous delivery" with technical debt, leading to shortcuts that violate the principle of technical excellence.

Agile Team

Related terms: Cross-functional Team, Self-organization, Scrum Team

An Agile Team is a small, self-organising group that possesses all the skills needed to deliver a potentially shippable product increment each iteration. The team collaborates closely, shares responsibility, and adapts its workflow continuously.

***Example*:** A team of developers, testers, UX designers, and a product owner works together without hand-offs, completing a feature from concept to deployment within a sprint.

***Practical application*:** Encourage the team to collectively decide how to split work during sprint planning, fostering ownership.

***Challenges*:** Achieving true cross-functionality can be difficult in organisations where specialist silos dominate, leading to bottlenecks.

Architecture Spike

Related terms: Spike, Technical Debt, Proof of Concept

A time-boxed investigation used to explore a technical approach or architectural decision before committing to full implementation. It produces a lightweight artefact—often a prototype or a set of findings—rather than production-ready code.

***Example*:** The team allocates a half-day spike to evaluate whether a microservices architecture can meet latency requirements.

***Practical application*:** Include spikes in the sprint backlog with clear acceptance criteria to avoid scope creep.

***Challenges*:** Spikes can be over-used, draining capacity from feature work; also, the outcomes may be ambiguous if not well-defined.

Backlog Grooming

Related terms: Backlog Refinement, User Stories, Definition of Ready

The ongoing process of reviewing, clarifying, and prioritising items in the product backlog to ensure they are ready for upcoming sprints. It involves adding details, estimating effort, and removing obsolete items.

***Example*:** The product owner meets with the team weekly to split large epics into manageable user stories and assign story points.

***Practical application*:** Schedule a regular 60-minute grooming session to keep the backlog actionable and aligned with business goals.

***Challenges*:** Teams may treat grooming as a “nice-to-have” activity, leading to poorly defined stories that cause rework during sprint planning.

Burn-down Chart

Related terms: Burn-up Chart, Velocity, Iteration Tracking

A visual tool that displays the remaining work (usually in story points) against time for a sprint. The chart helps the team monitor progress and predict whether the sprint goal will be met.

***Example*:** On day three of a two-week sprint, the burn-down shows a steep decline, indicating that the team is ahead of schedule.

***Practical application*:** Update the chart daily during the stand-up to provide a shared reference point for the team’s pace.

***Challenges*:** Inaccurate estimates or scope changes can distort the chart, leading to false confidence or

unnecessary alarm.

Burn-up Chart

Related terms: Burn-down Chart, Scope Management, Release Planning

Displays the amount of work completed over time, often against the total scope. Unlike burn-down, it makes scope changes visible, helping stakeholders understand progress despite added or removed items.

***Example*:** A release-level burn-up shows a gradual rise in completed story points while the total scope line shifts upward after a new feature is added.

***Practical application*:** Use the burn-up in release reviews to discuss the impact of scope changes on delivery dates.

***Challenges*:** Frequent scope volatility can make the chart appear erratic, reducing its credibility with stakeholders.

Definition of Done (DoD)

Related terms: Definition of Ready, Acceptance Criteria, Quality Assurance

A shared checklist that specifies all conditions a product increment must satisfy to be considered complete. It typically includes coding standards, unit testing, integration testing, documentation, and deployment readiness.

***Example*:** The DoD for a web feature includes passing all automated tests, code review approval, and successful deployment to a staging environment.

***Practical application*:** Review the DoD during sprint retrospectives to refine it as the team matures.

***Challenges*:** Overly strict DoD can slow delivery; overly lax DoD can result in low-quality increments that accumulate technical debt.

Definition of Ready (DoR)

Related terms: Definition of Done, Backlog Refinement, User Story

A set of criteria that a backlog item must meet before it can be taken into a sprint. The DoR ensures that the team has enough information, size, and clarity to start work without excessive interruptions.

***Example*:** A story meets DoR when it has clear acceptance criteria, is sized using story points, and has no unresolved dependencies.

***Practical application*:** Use DoR as a gate during sprint planning; items that fail the gate are deferred to the next grooming session.

***Challenges*:** Teams may set the DoR too high, causing delays, or too low, leading to rework and blockers mid-sprint.

Daily Stand-up

Related terms: Scrum Daily, Timeboxing, Team Synchronisation

A short, time-boxed (usually 15 minutes) meeting where each team member answers three questions: what they did yesterday, what they will do today, and any impediments. The purpose is to synchronise effort and surface blockers early.

***Example*:** During the stand-up, a developer mentions a blocked API dependency, prompting the Scrum Master to raise the issue immediately.

***Practical application*:** Keep the stand-up focused on progress and impediments; avoid problem-solving discussions that belong to separate meetings.

***Challenges*:** Teams can drift into status-reporting for managers rather than collaborative coordination, reducing the meeting's effectiveness.

Decision-making Matrix

Related terms: RACI Chart, Consensus, Facilitation Tools

A visual tool that maps decisions to responsible roles, clarifying who is accountable, consulted, or informed for each decision. It helps prevent ambiguity and ensures transparent ownership.

***Example*:** The matrix shows that the product owner is accountable for feature prioritisation, while the architecture lead is consulted on technical feasibility.

***Practical application*:** Introduce the matrix during sprint planning to clarify decision rights for backlog items.

***Challenges*:** Over-engineering the matrix can create bureaucracy; under-defining roles can lead to duplicated effort or conflict.

Definition of Value (DoV)

Related terms: Business Value, Prioritisation, Stakeholder Alignment

A concise statement that articulates the expected benefit of a backlog item, often expressed in monetary terms, risk reduction, or strategic impact. It aids the product owner in ordering work based on tangible outcomes.

***Example*:** The DoV for a security feature might be "reduce breach risk by 30% and avoid potential regulatory fines of £200k."

***Practical application*:** Capture DoV alongside each user story to facilitate data-driven prioritisation.

***Challenges*:** Estimating value can be speculative; inaccurate values may misguide the team's focus.

Definition of Vision

Related terms: Product Vision, Roadmap, Stakeholder Buy-in

A high-level description of the future state the product aims to achieve, providing direction and motivation for the team. It is typically concise, inspirational, and aligned with organisational goals.

Example: "Empower small businesses to manage inventory online with a single, intuitive dashboard."

Practical application: Revisit the vision at the start of each release planning session to keep the team aligned with long-term objectives.

Challenges: Vague visions can lead to divergent interpretations; overly detailed visions can stifle creativity.

Definition of Work-in-Progress (WIP) Limit

Related terms: Kanban, Flow Efficiency, Queue Theory

A constraint that caps the number of items allowed in a given workflow stage at any time. WIP limits help reduce multitasking, expose bottlenecks, and improve flow.

Example: The team sets a WIP limit of three items for the "In Development" column on the Kanban board.

Practical application: Adjust limits based on cycle-time metrics to optimise throughput.

Challenges: Teams may resist limits, fearing reduced flexibility; improperly set limits can cause idle time or backlog pile-up.

Definition of Sprint Goal

Related terms: Sprint Planning, Product Increment, Team Alignment

A concise, measurable objective that the team commits to achieving during a sprint. The goal provides focus and a shared purpose beyond the sum of individual backlog items.

Example: "Implement the checkout flow and achieve a 95 % automated test coverage for the payment module."

Practical application: Review the sprint goal daily during stand-ups to keep the team oriented toward the intended outcome.

Challenges: Vague goals can dilute focus; overly ambitious goals may lead to unfinished work and reduced morale.

Definition of Story Points

Related terms: Relative Estimation, Velocity, Planning Poker

A unitless measure used to express the relative effort required to implement a user story. Story points capture complexity, risk, and amount of work, enabling teams to forecast capacity.

Example: A simple login page is estimated at 2 points, while a multi-currency payment integration is 13 points.

Practical application: Use Planning Poker to reach consensus on point values, fostering shared understanding.

***Challenges*:** Teams may compare points across projects, leading to inconsistent sizing; anchoring bias can skew estimates.

Definition of Technical Debt

Related terms: Refactoring, Quality Assurance, DoD

The implied cost of additional rework caused by choosing an expedient solution now instead of a more robust approach. Technical debt accumulates when shortcuts are taken, reducing future productivity.

***Example*:** Skipping unit tests to meet a deadline creates debt that must be paid later through additional testing effort.

***Practical application*:** Allocate a fixed percentage of each sprint to debt reduction, making it visible in the backlog.

***Challenges*:** Stakeholders may view debt work as “non-functional” and deprioritise it, causing long-term maintainability issues.

Definition of User Story

Related terms: Epic, Acceptance Criteria, Backlog Item

A short, simple description of a product feature from the perspective of an end-user, typically following the format: “As a , I want so that .” It captures functional requirements in a conversational style.

***Example*:** “As a shopper, I want to filter products by price so that I can find items within my budget quickly.”

***Practical application*:** Write user stories collaboratively during backlog grooming to ensure shared understanding.

***Challenges*:** Over-granular stories can lead to excessive overhead; overly large stories (epics) need to be broken down for effective sprint planning.

Definition of Velocity

Related terms: Story Points, Capacity Planning, Iteration Forecast

The average number of story points a team completes per sprint, measured over several past sprints.

Velocity provides a baseline for future sprint capacity and helps set realistic expectations.

***Example*:** Over the last five sprints, a team’s velocity averaged 42 points, guiding the next sprint’s forecast.

***Practical application*:** Use velocity to size the sprint backlog, adjusting for known holidays or team member availability.

***Challenges*:** Velocity should not be treated as a performance metric; fluctuations due to scope changes or team composition can mislead planning if not contextualised.

Definition of Work-Item Type

Related terms: Bug, Task, Feature

Categories used to classify backlog items, each with distinct workflows and priorities. Common types include "Feature," "Bug," "Technical Debt," and "Spike."

Example: In Jira, a "Bug" automatically triggers a higher-priority queue than a "Feature."

Practical application: Define clear criteria for each type to help the product owner triage and the team plan work efficiently.

Challenges: Mis-labelled items can cause confusion during sprint planning and affect metrics such as lead time.

Definition of Work-Item Prioritisation

Related terms: MoSCoW, Weighted Shortest Job First (WSJF), Business Value

The process of ordering backlog items based on factors such as value, risk, effort, and strategic fit.

Prioritisation ensures the team delivers the most important work first.

Example: Using WSJF, a team calculates the score for each feature and orders them accordingly.

Practical application: Conduct regular prioritisation workshops with stakeholders to keep the backlog aligned with evolving business needs.

Challenges: Conflicting stakeholder interests can lead to deadlock; over-reliance on a single metric may ignore qualitative factors.

Definition of WIP (Work-in-Progress) Limit

Related terms: Kanban Board, Flow Efficiency, Cycle Time

A constraint that caps the number of items allowed in a particular workflow stage at any given moment, preventing overload and encouraging smooth flow.

Example: The "Testing" column is limited to four items; once full, developers must wait before pushing new code.

Practical application: Monitor the board daily; if a column consistently hits its limit, investigate the bottleneck.

Challenges: Teams may circumvent limits by creating workarounds, undermining the intended benefits of flow control.

Definition of Epic

Related terms: User Story, Feature, Roadmap

A large body of work that can be broken down into multiple user stories. Epics represent significant business objectives that span several sprints or releases.

Example: "Create a mobile app for the e-commerce platform" is an epic that includes stories for login, browsing, cart, and checkout.

Practical application: Use epics to organise the product backlog and align them with strategic themes on the roadmap.

Challenges: Epics can become vague "catch-all" items if not regularly refined into actionable stories.

Definition of Feature

Related terms: Epic, User Story, Product Increment

A distinct piece of functionality delivering value to the end user, often representing a collection of related user stories. Features are typically tracked in the backlog and may map to release goals.

Example: The "Search" feature includes stories for keyword search, filters, and result ranking.

Practical application: Group related stories under a feature to maintain traceability from requirement to delivery.

Challenges: Over-granular features can cause unnecessary coordination; under-granular features may hide complexity, leading to surprise during implementation.

Definition of Facilitation

Related terms: Servant Leadership, Active Listening, Meeting Design

The skill of guiding a group toward a productive outcome while remaining neutral, ensuring all voices are heard, and managing group dynamics. A facilitator creates an environment where collaboration flourishes.

Example: During a retrospective, the facilitator uses a "Start-Stop-Continue" exercise to elicit feedback from all team members.

Practical application: Apply facilitation techniques such as timeboxing, parking lot, and round-robin to keep meetings focused.

Challenges: Facilitators can inadvertently dominate discussions or fail to address conflict, reducing the group's sense of ownership.

Definition of Icebreaker

Related terms: Team Building, Warm-up Activity, Facilitation Technique

A short, engaging activity used at the start of a meeting or workshop to create a relaxed atmosphere, encourage participation, and build rapport among participants.

Example: "Two Truths and a Lie" helps new team members learn about each other before a sprint planning session.

Practical application: Choose an icebreaker relevant to the group size and session purpose; keep it under five minutes.

***Challenges*:** Inappropriate or overly elaborate icebreakers can waste time or make participants uncomfortable.

Definition of Impact Mapping

Related terms: User Story Mapping, Goal-Question-Metric (GQM), Strategic Planning

A visual technique that links product features to business goals, actors, and impacts, helping teams prioritise work that delivers real value. The map consists of four layers: Goal, Actors, Impacts, and Deliverables.

***Example*:** The goal "Increase subscription conversions" leads to actor "Visitor," impact "Clicks the trial button," and deliverable "Add prominent CTA."

***Practical application*:** Conduct an impact-mapping session during release planning to ensure alignment with strategic objectives.

***Challenges*:** Teams may oversimplify the map, ignoring secondary actors or indirect impacts, resulting in sub-optimal prioritisation.

Definition of Increment

Related terms: Sprint Goal, Definition of Done, Potentially Shippable

The sum of all completed product backlog items at the end of a sprint, integrated with previous increments and meeting the DoD. It represents a usable, tested piece of the product.

***Example*:** After sprint 4, the increment includes a functional search feature, tested API endpoints, and updated documentation.

***Practical application*:** Demonstrate the increment in the sprint review to gather stakeholder feedback.

***Challenges*:** If the DoD is not consistently applied, the increment may contain hidden defects, reducing confidence in the product's stability.

Definition of Iteration

Related terms: Sprint, Timebox, Cycle Time

A fixed-length period (commonly two to four weeks) during which a cross-functional team delivers a set of backlog items, producing a potentially shippable increment. Iterations promote regular feedback and adaptation.

***Example*:** A two-week iteration starts on Monday and ends on the following Friday, concluding with a review and retrospective.

***Practical application*:** Use a consistent iteration cadence to establish rhythm and predictability for the team and stakeholders.

***Challenges*:** Changing iteration length mid-project can disrupt cadence and affect velocity calculations.

Definition of Kaizen

Related terms: Continuous Improvement, Retrospective, Lean

A Japanese term meaning “change for the better,” representing a philosophy of ongoing, incremental improvement in processes, products, and culture. In agile, Kaizen is realised through regular retrospectives and small experiments.

***Example*:** After a retrospective, the team decides to adopt pair programming for complex modules, improving code quality.

***Practical application*:** Encourage the team to propose one small improvement per sprint and track its impact.

***Challenges*:** Without visible results, teams may lose motivation to pursue Kaizen, reverting to old habits.

Definition of Kanban

Related terms: Pull System, WIP Limit, Flow Metrics

A visual workflow management method that uses a board with columns representing process stages, allowing teams to limit WIP, visualise bottlenecks, and improve flow. Kanban is often used in conjunction with Scrum to enhance delivery.

***Example*:** A Kanban board shows columns “To Do,” “In Progress,” “Code Review,” and “Done,” each with a WIP limit of three.

***Practical application*:** Implement cumulative flow diagrams to monitor lead time and identify process improvements.

***Challenges*:** Teams may misuse the board as a status report for managers, neglecting its purpose as a flow-optimisation tool.

Definition of Lead Time

Related terms: Cycle Time, Throughput, Value Stream Mapping

The elapsed time from the moment a request is made (e.g., a user story added to the backlog) until the product increment is delivered to the customer. Lead time measures overall responsiveness.

***Example*:** A feature’s lead time is 15 days from backlog entry to production release.

***Practical application*:** Track lead time to identify delays in early stages such as analysis or design.

***Challenges*:** High variability in lead time can obscure root causes, requiring deeper analysis of each workflow stage.

Definition of Lean

Related terms: Value Stream, Waste Elimination, Kaizen

A set of principles focused on delivering maximum customer value while minimising waste. Lean encourages continuous flow, pull-based scheduling, and respect for people. In software, lean aligns with agile by emphasizing rapid feedback and iterative delivery.

***Example*:** A team adopts lean by eliminating unnecessary documentation that does not add customer value.

***Practical application*:** Conduct value-stream mapping workshops to identify and remove non-value-adding activities.

***Challenges*:** Misinterpreting “waste” can lead to cutting essential activities such as testing, compromising quality.

Definition of Pair Programming

Related terms: Mob Programming, Code Review, Collaboration

An agile development technique where two developers work together at one workstation: the “driver” writes code while the “observer” reviews each line in real time. Roles switch frequently to share knowledge and improve code quality.

***Example*:** During a complex algorithm implementation, two developers rotate driver/observer every 20 minutes.

***Practical application*:** Use pair programming for onboarding new team members to accelerate knowledge transfer.

***Challenges*:** Some developers may feel uncomfortable sharing a keyboard, and perceived productivity loss can cause resistance.

Definition of Parking Lot

Related terms: Facilitation Technique, Meeting Management, Scope Control

A visual list of topics that arise during a meeting but are not immediately addressed, allowing the facilitator to defer them without losing the idea. The parking lot ensures focus while preserving valuable input.

***Example*:** During sprint planning, a discussion about a future integration is added to the parking lot for later analysis.

***Practical application*:** Review the parking lot at the end of the meeting or schedule a follow-up session to address deferred items.

***Challenges*:** If the parking lot is neglected, important topics may be forgotten, leading to incomplete decisions.

Definition of Peer Review

Related terms: Code Review, Quality Assurance, Collaborative Editing

A systematic examination of work products by colleagues to detect defects, improve quality, and share knowledge. Peer reviews can be formal (checklists) or informal (pair programming).

***Example*:** A developer submits a pull request, and three peers review the changes, providing feedback on naming conventions and test coverage.

***Practical application*:** Integrate peer review into the CI pipeline to enforce mandatory approvals before merge.

***Challenges*:** Overly critical reviews can discourage contributors; insufficient time allocated for reviews can reduce effectiveness.

Definition of Product Backlog

Related terms: Backlog Grooming, User Story, Prioritisation

An ordered list of all desired work for the product, maintained by the product owner. Items range from features and enhancements to bugs and technical debt, each with a description, priority, and estimate.

***Example*:** The backlog contains 120 items, with the top 15 ready for the next sprint.

***Practical application*:** Keep the backlog visible to the whole team using a shared tool, updating it continuously based on feedback.

***Challenges*:** Large, unmanaged backlogs become unwieldy, making it difficult to locate high-priority items.

Definition of Product Owner

Related terms: Stakeholder, Scrum Master, Backlog Management

The role responsible for maximising product value by managing the product backlog, prioritising items, and clarifying requirements. The product owner acts as the voice of the customer and ensures the team works on the most valuable features.

***Example*:** The product owner collaborates with marketing to align upcoming releases with campaign timelines.

***Practical application*:** Conduct regular stakeholder interviews to keep the backlog aligned with market needs.

***Challenges*:** Balancing competing stakeholder demands and resisting scope creep while maintaining a clear vision.

Definition of Pull System

Related terms: Kanban, WIP Limit, Just-in-Time

A workflow approach where work is only started when there is capacity downstream, preventing overload and ensuring smooth flow. Items are “pulled” from the backlog into the next stage as space becomes available.

Example: Developers pull the next story from "Ready" only after completing a current task, respecting the WIP limit.

Practical application: Implement a "ready-to-start" column with explicit entry criteria to support the pull mechanism.

Challenges: In environments with strict deadlines, teams may feel pressured to "push" work, undermining the pull principle.

Definition of Retrospective

Related terms: Continuous Improvement, Kaizen, Facilitation

A regular meeting held at the end of a sprint where the team reflects on what went well, what didn't, and how to improve. The retrospective aims to generate actionable improvement items for the next sprint.

Example: The team identifies "improve test automation coverage" as a focus for the upcoming iteration.

Practical application: Use varied formats (e.g., "4-Ls", "Mad-Sad-Glad") to keep retrospectives engaging.

Challenges: If the same issues recur without resolution, the team may become cynical, reducing the ceremony's effectiveness.

Definition of Scrum

Related terms: Sprint, Scrum Master, Product Owner

A lightweight framework for managing complex product development, consisting of roles, events, artefacts, and rules. Scrum promotes iterative progress through time-boxed sprints, daily stand-ups, and regular inspection and adaptation.

Example: A Scrum team delivers a potentially shippable increment every two weeks, incorporating stakeholder feedback after each sprint review.

Practical application: Adopt the three-pillar model (Transparency, Inspection, Adaptation) to guide daily practices.

Challenges: Misinterpreting Scrum as a methodology rather than a framework can lead to "Scrum-but" antipatterns, where essential events are skipped or altered.

Definition of Scrum Master

Related terms: Servant Leader, Facilitator, Impediment Removal

The servant-leader responsible for ensuring the Scrum process is understood and enacted. The Scrum Master coaches the team, removes impediments, and shields the team from external disruptions.

Example: The Scrum Master negotiates with a downstream team to resolve a blocking API dependency.

Practical application: Conduct regular coaching sessions to strengthen self-organisation and adherence to the DoD.

Challenges: Balancing the role of coach versus manager can be difficult, especially in organisations with hierarchical cultures.

Definition of Self-organization

Related terms: Agile Team, Empowerment, Team Autonomy

The ability of a team to determine the best way to accomplish its work without external direction.

Self-organising teams decide how to allocate tasks, manage workflow, and improve processes.

Example: The team collectively decides to adopt a new testing framework after evaluating its benefits.

Practical application: Encourage the team to set its own sprint goals and define its own Definition of Done.

Challenges: Lack of clear boundaries can lead to decision-fatigue; teams may need guidance to avoid paralysis.

Definition of Servant Leadership

Related terms: Scrum Master, Empowerment, Facilitation

A leadership philosophy where the leader's primary purpose is to serve the team, removing obstacles, fostering growth, and enabling high performance. Servant leaders listen actively, share power, and support development.

Example: A manager allocates budget for team training after hearing a request during a retrospective.

Practical application: Practice "listening first" in meetings, summarising concerns before offering solutions.

Challenges: In traditional command-and-control cultures, servant leadership may be misunderstood as a lack of authority.

Definition of Sprint

Related terms: Iteration, Sprint Planning, Timebox

A fixed-duration development cycle, typically 1-4 weeks, during which a Scrum team delivers a potentially shippable product increment. The sprint begins with planning and ends with a review and retrospective.

Example: A two-week sprint starts on Monday, with a sprint goal to implement the user profile feature.

Practical application: Maintain a consistent sprint length to stabilise velocity and facilitate forecasting.

Challenges: Mid-sprint scope changes can disrupt flow and reduce the team's ability to meet the sprint goal.

Definition of Sprint Backlog

Related terms: Sprint Planning, Task Board, Commitment

The set of product backlog items selected for the sprint, plus a plan for delivering the increment and achieving the sprint goal. It is owned by the development team and evolves daily.

Example: The sprint backlog contains five user stories and three technical tasks, each broken into subtasks on the task board.

Practical application: Update the sprint backlog during the daily stand-up to reflect completed work and emerging impediments.

Challenges: Over-commitment can lead to unfinished work, while under-commitment may waste capacity.

Definition of Sprint Goal

Related terms: Sprint Planning, Product Increment, Focus

A concise, shared objective for the sprint that provides guidance to the development team on why they are building the increment. The goal is set during sprint planning and remains the focal point throughout the sprint.

Example: "Enable users to reset their password securely."

Practical application: Refer to the sprint goal during daily stand-ups to keep the team aligned on priorities.

Challenges: Vague or overly broad goals can dilute focus, leading to scattered effort across unrelated stories.

Definition of Sprint Planning

Related terms: Sprint Goal, Backlog Grooming, Capacity Planning

A collaborative meeting at the start of each sprint where the product owner presents the top backlog items, the team forecasts its capacity, and together they select work for the sprint. The outcome is a sprint backlog and a sprint goal.

Example: The team uses historical velocity and team availability to decide it can commit to eight story points.

Practical application: Use a "capacity calculator"