How to Plan a Sprint and Define Its Goal



Plan around a useful goal

Sprint Planning creates a coherent reason for the work and a credible starting plan.

Use the purpose

The Scrum Team discusses why the Sprint is valuable, what can be accomplished and how the selected work can be done. Developers create and adapt the delivery plan.

Pickup-booking case

A team wants existing customers to reserve a valid pickup slot without calling staff. The Sprint should aim at that usable result rather than an unrelated list of requests.

Use the workbench

Prepare the context, clarify options, form a goal and select feasible work with quality and dependencies visible.

TAKE IT TO YOUR PROJECT

Write the valuable result the Sprint could make possible.

Connect with the Product Goal

The near-term objective should support a coherent product direction.

Use the longer view

The Product Owner explains relevant value and Product Backlog ordering. Discuss how a Sprint could move the product toward the Product Goal.

Booking case

The Product Goal concerns reliable self-service collection. A Sprint focused on valid pickup reservations supports that direction and creates a useful opportunity for feedback.

Check the boundary

A valuable idea may still be the wrong focus now if a dependency or more urgent need changes the choice. Make the reason discussable.

TAKE IT TO YOUR PROJECT

Explain how a proposed Sprint outcome advances the Product Goal.

Prepare the candidate work

Planning is easier when important needs and uncertainties are understood.

Refine usefully

Clarify value, examples, quality, dependencies and size for likely near-term items. Product Backlog refinement is ongoing and does not eliminate learning during a Sprint.

Booking case

The team discusses available slots, reservation confirmation and capacity handling. It identifies an external scheduling interface that needs early verification.

Keep options open

A candidate is not automatically a commitment. Developers and the Product Owner need enough understanding to discuss a feasible Sprint forecast.

TAKE IT TO YOUR PROJECT

Name the unanswered question most likely to affect selection.

Understand the team's capacity

Availability is an input to a forecast, not a promise of perfect productivity.

Use actual conditions

Consider skills, absences, support work, collaboration, known commitments and delivery experience. Scrum does not require one particular capacity or sizing method.

Booking example

Three Developers each have 10 working days with 6 planned focus hours, giving 180 hours. Subtract 18 known absence hours and 24 separately planned support hours to obtain 138.

Interpret carefully

This illustrative arithmetic assumes consistent definitions and no double counting. It does not guarantee that every estimated hour converts into completed value.

TAKE IT TO YOUR PROJECT

List the assumptions behind your team's available time.

Learn from previous delivery

History is useful when the work and conditions are comparable.

Inspect the evidence

Review usable results, quality, incomplete work, team changes and relevant delivery history. Use the reason behind a difference rather than mechanically copying last Sprint's amount.

Booking case

The team finished fewer items because interface access was late. This Sprint's forecast should consider whether the access problem is resolved, not simply demand a higher number.

Avoid cross-team comparisons

Team-specific sizing or velocity is not a universal productivity unit. Focus on this team's evidence and ability to meet a useful goal.

TAKE IT TO YOUR PROJECT

Name a lesson from the last Sprint that changes today's forecast.

The All-in-One PM Templates Bundle

940+ Premium Templates • One Payment • Lifetime Access

$49

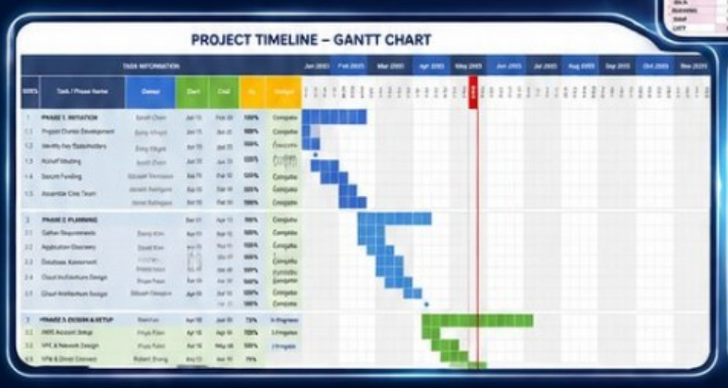
SAVE 94%

Worth $3,300+











EXCEL & GOOGLE SHEETS | PROFESSIONAL & RELIABLE | AUTOMATED DASHBOARDS | SAVE TIME & STAY ON TRACK | INSTANT ACCESS DOWNLOAD

GET LIFETIME ACCESS

Draft an outcome-focused Sprint Goal

The goal should help the team make choices as details change.

Describe the value

Use one coherent objective that explains why the Sprint is worth doing. It should provide focus while allowing flexibility in the implementation details.

Booking case

"Enable existing customers to reserve a valid pickup slot without staff help" expresses a usable outcome. The team can discuss which slice is feasible.

Test the wording

Would the goal help decide whether a new request belongs in the Sprint? A list of unrelated item names may provide little guidance.

TAKE IT TO YOUR PROJECT

Draft a goal and explain one trade-off it would help resolve.

Choose a feasible slice

A coherent goal still needs achievable work and acceptance evidence.

Select collaboratively

The Product Owner and Developers discuss suitable Product Backlog items. Developers use capacity, understanding and experience to forecast what they can accomplish.

Booking case

The team considers slot availability, creating a reservation and confirmation for existing customers. A new marketing campaign is not needed to achieve this selected goal.

Check completeness

Include the work required to make the slice usable, with relevant quality and integration. Do not count a partial happy-path screen as the whole result.

TAKE IT TO YOUR PROJECT

Choose the smallest useful result that supports your goal.

Create the initial plan

Developers decide how to turn selected work into a usable Increment.

Identify the work

Discuss the steps, dependencies, collaboration and verification needed. Break down enough work to start while retaining the ability to adapt as more is learned.

Booking case

Verify the scheduling interface, implement slot selection and reservation behaviour, test capacity changes and confirm the agreed result. The team decides who collaborates on each part.

Keep ownership with the team

The plan supports self-management. A detailed assignment imposed from outside can undermine the team's ability to coordinate its actual work.

TAKE IT TO YOUR PROJECT

Describe the first day's work and the evidence it should produce.

Make quality explicit

A forecast should include the effort needed to meet the Definition of Done.

Use the shared definition

Developers maintain quality through the Definition of Done. Story-specific criteria add the behaviour needed for the selected slice.

Booking case

The reservation must use a valid slot, update capacity correctly and satisfy the relevant shared checks. Unverified work cannot simply be counted as Done to preserve scope.

Plan the evidence

Identify representative cases, relevant failure conditions and operating implications early enough to influence the approach.

TAKE IT TO YOUR PROJECT

Choose one check the team should prepare before implementation is complete.

Check dependencies and uncertainty

A known external wait can dominate a short planning horizon.

Expose the input

Identify access, data, decisions, suppliers or other teams needed for the goal. Confirm ownership, expected timing and readiness evidence.

Booking case

The scheduling test environment is uncertain. The team verifies access early and discusses an alternative slice or response if the required interface cannot be tested.

Use the decision

A dependency that cannot be resolved within the Sprint may change selection or the goal's feasibility. Discuss it before treating the forecast as reliable.

TAKE IT TO YOUR PROJECT

Name the dependency most likely to endanger the goal.

Read the Sprint Backlog

The artifact should show the goal, selected work and a delivery plan.

Connect the elements

The Sprint Goal explains why, selected Product Backlog items describe what and the Developers' plan describes how. Keep the view understandable and current.

Booking case

Goal: valid self-service pickup reservation. Selected work: availability, reservation and confirmation slice. Plan: interface checks, implementation and verification with visible dependencies.

Inspect the result

A board of tasks with no visible goal or quality boundary may not support the decisions the team needs. Use a representation that helps the team adapt.

TAKE IT TO YOUR PROJECT

Explain the three parts of your Sprint Backlog.

Work a complete planning example

Use one coherent thread from value to the first actions.

Why and what

The Product Owner explains the self-service need. The team forms the pickup-reservation goal and selects the usable existing-customer slice based on capacity and understanding.

How

Developers plan the interface test, slot-selection behaviour, reservation change and confirmation checks. They make the external access risk and shared quality requirements visible.

Check the forecast

The illustrative 138-hour capacity informs discussion but does not replace judgement about complexity, dependencies and Done. The team agrees a credible starting plan.

TAKE IT TO YOUR PROJECT

Walk through the example and identify the first uncertainty to test.

Adapt during the Sprint

The plan should change when evidence changes the best next step.

Use inspection

Developers inspect progress toward the Sprint Goal and adapt the Sprint Backlog. Scope can be clarified and renegotiated with the Product Owner as learning occurs.

Booking case

The interface returns different availability data from the example. The team adjusts the implementation plan and discusses a narrower usable slice while protecting the goal.

Maintain boundaries

Do not lower quality or quietly accept changes that endanger the Sprint Goal. Escalate relevant wider commitments through the organisation's actual routes.

TAKE IT TO YOUR PROJECT

Describe a scope adjustment that could preserve the goal.

Handle a new urgent request

A request needs assessment against impact, goal and available options.

Understand the situation

Clarify the need and urgency with the appropriate people. Consider what happens to the current Sprint work and whether the goal remains relevant.

Booking case

A serious access failure affects existing users. The team and Product Owner discuss the response and its effect on the goal, making the change transparent.

Use the framework

The Product Owner alone has authority to cancel a Sprint if its goal becomes obsolete. A difficult task or changed detail does not automatically require cancellation.

TAKE IT TO YOUR PROJECT

Write the questions needed before changing the Sprint's focus.

Use the events for their purposes

Planning begins a learning cycle that continues through the Sprint.

Daily Scrum

Developers inspect progress toward the goal and adapt their plan. This fifteen-minute event helps the next decisions rather than serving only as a report to a manager.

Review and Retrospective

The Sprint Review inspects the product outcome with stakeholders. The Retrospective improves effectiveness and quality in how the Scrum Team works.

Booking case

User feedback on the reservation changes the next product choices, while a recurring test-data delay becomes a team improvement experiment.

TAKE IT TO YOUR PROJECT

Name one useful question for each inspection point.



Learn from 350+ top universities and companies

ILLINOIS

Duke UNIVERSITY

Google

UNIVERSITY OF MICHIGAN

IBM

VANDERBILT UNIVERSITY

JOHNS HOPKINS UNIVERSITY

Invest in your career

Explore new skills

Access 10,000+ courses in AI, business, technology, and more.

Your Sprint Planning desk check

A plan is useful when the team can explain its purpose and act on it.

Check the goal

Is there one coherent objective connected with product value? Can it guide a trade-off as the team learns?

Check the forecast

Are selected work, capacity, dependencies, uncertainty and quality understood? Do Developers own a feasible starting plan?

Use the booking case

The team can explain why valid reservations matter, what usable slice it selected and how it will inspect progress. The plan remains adaptable without losing that focus.

TAKE IT TO YOUR PROJECT

Ask the team to explain the goal, first actions and biggest uncertainty in its own words.