

PROFESSIONAL COURSE EDITION

SCRUM PLANNING MASTERY

From Product Vision to Predictable Sprints

Backlogs · Refinement · Estimation · Reviews · Metrics · Scaling

A COMPLETE COURSE BOOK & FIELD WORKBOOK

LONDON WULFE

2026

Scrum Planning Mastery

From Product Vision to Predictable Sprints

London Wulfe

Professional Course Edition ♦♦ 2026

About This Book

Course Outcomes

Contents

Preface: Planning Without Pretending

Part I — Foundations

Chapter 1 — Scrum as a Planning System

1.1 The Problem Scrum Solves

Field Exercise: Choose the Right Work System

Knowledge Check

Chapter 2 — Empiricism, Values, and Complexity

2.1 The Three Pillars

Field Exercise: Transparency Audit

Knowledge Check

Chapter 3 — Accountabilities and Team Design

3.1 One Scrum Team

Field Exercise: Team Design Map

Knowledge Check

Chapter 4 — Events, Artifacts, and Commitments

4.1 The Sprint and Its Events

Field Exercise: Event Health Check

Knowledge Check

Part II — Product Direction

Chapter 5 — Product Vision and Product Goals

5.1 Vision Before Backlog

Field Exercise: Write the Product Goal

Knowledge Check

Chapter 6 — Stakeholders, Users, and Value

6.1 Stakeholders Are Not All Equal Requests

Field Exercise: Stakeholder Evidence Map

Knowledge Check

Chapter 7 — Building and Ordering the Product Backlog

7.1 Product Backlog Item Types

Field Exercise: Backlog Cleanup

Knowledge Check

Chapter 8 — Backlog Refinement and Readiness

8.1 Refinement Is Ongoing

Field Exercise: Refine One Real Item

Knowledge Check

Part III — Sprint Planning

Chapter 9 — Estimation, Forecasting, and Capacity

9.1 What Estimation Can and Cannot Do

Field Exercise: Build a Capacity Forecast

Knowledge Check

Chapter 10 — Sprint Planning: Why, What, and How

10.1 Preparation

Field Exercise: Run a 60-Minute Planning Simulation

Knowledge Check

Chapter 11 — Writing Strong Sprint Goals

11.1 Characteristics of a Strong Goal

Field Exercise: Goal Clinic

Knowledge Check

Chapter 12 — Building and Managing the Sprint Backlog

12.1 What Belongs in the Sprint Backlog

Field Exercise: Sprint Backlog Redesign

Knowledge Check

Part IV — Execution and Learning

Chapter 13 — Daily Scrum and Flow

13.1 Purpose and Structure

Field Exercise: Redesign the Daily Scrum

Knowledge Check

Chapter 14 — Quality, Definition of Done, and Technical Debt

14.1 Definition of Done

Field Exercise: Definition of Done Workshop

Knowledge Check

Chapter 15 — Sprint Review and Product Adaptation

15.1 Beyond the Demo

Field Exercise: Design the Next Review

Knowledge Check

Chapter 16 — Sprint Retrospective and Continuous Improvement

16.1 What the Team Inspects

Field Exercise: Run an Improvement Experiment

Knowledge Check

Part V — Leadership and Application

Chapter 17 — Metrics, Evidence, and Forecasting

17.1 A Balanced Measurement System

Field Exercise: Build a One-Page Product Dashboard

Knowledge Check

Chapter 18 — Multi-Team Planning, App Delivery, and Construction Adaptation

18.1 Multiple Teams, One Product

Field Exercise: Select a 90-Day Implementation Target

Knowledge Check

Part VI — Workbook and Implementation Tools

Template 1 — Product Vision Canvas

Template 2 — Product Goal Canvas

Template 3 — Product Backlog Item

Template 4 — Refinement Checklist

Template 5 — Sprint Planning Canvas

Template 6 — Sprint Goal Test

Template 7 — Definition of Done Workshop

Template 8 — Daily Scrum Flow Board

Template 9 — Sprint Review Plan

Template 10 — Retrospective Experiment

Template 11 — Product Metrics Dashboard

Template 12 — Impediment Escalation Log

90-Day Scrum Deployment Plan

Deployment Checklist

Capstone Project — Plan and Run a Four-Sprint Product Cycle

Capstone Scoring Rubric

Mock Final Examination

Answer Key and Rationales

Glossary

References and Further Study

About This Book

This independent course book teaches Scrum planning and practical agile product delivery. It is designed for product teams, app builders, project leaders, operations teams, and construction professionals who want to use short planning cycles to reduce uncertainty and improve delivery.

IMPORTANT NOTICE

Scrum is defined in *The Scrum Guide* by Ken Schwaber and Jeff Sutherland. This book is not affiliated with, endorsed by, or a substitute for official training from Scrum.org, Scrum Alliance, or any certification body. Scrum and related marks belong to their respective owners.

How to use it. Read one chapter, complete the knowledge check, then apply the field exercise to a real product or workstream. The templates in Part VI can be copied directly into a notebook, spreadsheet, project platform, or app.

Standards and safety. When Scrum is adapted to construction, field execution remains governed by contract documents, applicable building codes, OSHA requirements, manufacturer instructions, and relevant ACI/ASTM standards. Agile planning never overrides safety, quality, inspection, or engineering requirements.

Course Outcomes

- Define Scrum accurately and distinguish it from generic task management.
- Create Product Goals, Sprint Goals, ordered Product Backlogs, and usable Sprint Backlogs.
- Facilitate refinement, estimation, Sprint Planning, Daily Scrum, Sprint Review, and Sprint Retrospective.
- Measure outcomes, flow, quality, predictability, and learning without weaponizing metrics.
- Apply Scrum to app development, internal operations, and selected construction-planning use cases.
- Lead a complete Scrum implementation through a 90-day rollout and capstone project.

Contents

Part I — Foundations

1. Scrum as a Planning System
2. Empiricism, Values, and Complexity
3. Accountabilities and Team Design
4. Events, Artifacts, and Commitments

Part II — Product Direction

5. Product Vision and Product Goals
6. Stakeholders, Users, and Value
7. Building and Ordering the Product Backlog
8. Backlog Refinement and Readiness

Part III — Sprint Planning

9. Estimation, Forecasting, and Capacity
10. Sprint Planning: Why, What, and How
11. Writing Strong Sprint Goals
12. Building and Managing the Sprint Backlog

Part IV — Execution and Learning

13. Daily Scrum and Flow
14. Quality, Definition of Done, and Technical Debt
15. Sprint Review and Product Adaptation
16. Sprint Retrospective and Continuous Improvement

Part V — Leadership and Application

17. Metrics, Evidence, and Forecasting
18. Multi-Team Planning, App Delivery, and Construction Adaptation

Part VI — Workbook

Templates · 90-Day Deployment Plan · Capstone · Mock Exam · Answer Key · Glossary · References

Preface: Planning Without Pretending

Traditional plans often fail because they treat uncertainty like a clerical problem. A date is entered, a task is assigned, and everyone acts as though reality has agreed. Scrum takes a different position: complex work cannot be predicted in full at the beginning, but it can be steered through short cycles of transparency, inspection, and adaptation.

Scrum planning is not less planning. It is planning at multiple horizons. Product direction answers where the team is going. Product Backlog ordering answers what matters next. Sprint Planning creates a near-term forecast. The Daily Scrum adjusts the plan as facts emerge. The Sprint Review updates product direction, and the Retrospective improves the system that produced the work.

This book is written for people who need planning to survive contact with reality. It uses examples from app development, operations, and construction because the core challenge is the same: coordinate skilled people around valuable outcomes while conditions keep changing.

The goal is not to make the team look busy. The goal is to produce usable value, learn quickly, and become more reliable without sacrificing quality or people.

Part I — Foundations

The framework, its logic, its people, and its events.

Chapter 1 — Scrum as a Planning System

Scrum is a lightweight framework for generating value through adaptive solutions to complex problems.

LEARNING OBJECTIVES

Explain what Scrum is and is not. Describe planning horizons in Scrum. Identify when Scrum is appropriate.

1.1 The Problem Scrum Solves

Complex work contains unknowns that cannot be eliminated by making a larger initial plan. Customer needs shift, technology behaves unexpectedly, vendors miss dates, design assumptions fail, and teams learn as they build. Scrum creates a disciplined way to convert those unknowns into frequent decisions.

A Scrum Team works in Sprints. Each Sprint is a fixed-length container — one month or less — during which the team pursues a Sprint Goal and creates a usable Increment. The Sprint is not merely a reporting period. It is a learning cycle that forces the team to confront reality often enough to adapt before the cost of being wrong becomes enormous.

PLANNING HORIZONS

Product planning establishes direction through a Product Goal. Release or outcome planning explores likely paths without turning them into promises. Sprint Planning creates a forecast for the current Sprint. Daily planning adjusts the work toward the Sprint Goal. These horizons are connected, but each carries a different level of detail and certainty.

Horizon	Primary question	Typical artifact	Certainty
Product	What valuable future state are we pursuing?	Product Goal and ordered Product Backlog	Directional
Sprint	What valuable objective can we achieve next?	Sprint Goal and Sprint Backlog	Forecast
Daily	What adjustment improves progress today?	Updated Sprint Backlog	Immediate
Release / outcome	When might a useful capability be available?	Forecast or roadmap	Probabilistic

WHEN SCRUM FITS

Scrum is most useful when the work is complex, the correct solution is not fully known, feedback can change the plan, and a cross-functional team can create a usable result in short cycles. It is less useful for highly repetitive production work with stable inputs and a proven process; flow systems such as Kanban or standard work may fit better there.

Construction field production is not automatically a Scrum problem. A structural concrete placement, for example, is constrained by drawings, inspections, sequence, curing, safety, and irreversible physical work. Scrum can still help teams develop software, coordinate design decisions, solve recurring process problems, or manage innovation — but it should complement, not replace, pull planning, look-ahead planning, and production controls.

CORE DISTINCTION

A Sprint is a fixed learning cycle. The Sprint Backlog is a forecast. The Sprint Goal is the commitment that gives the forecast purpose.

Field Exercise: Choose the Right Work System

1. List three workstreams you currently manage.
2. Classify each as complex, complicated, or repetitive.
3. Choose one complex workstream that could produce a usable result within two weeks.
4. Write what evidence would show that Scrum is helping.

Knowledge Check

1. Why is a Sprint more than a reporting period?

2. What is the difference between a forecast and a guarantee?

3. Name one situation where Scrum may not be the best primary system.

Chapter 2 — Empiricism, Values, and Complexity

Scrum replaces false certainty with evidence, disciplined inspection, and timely adaptation.

LEARNING OBJECTIVES
Describe transparency, inspection, and adaptation. Connect the five Scrum values to planning behavior.
Recognize fake agility.

2.1 The Three Pillars

Transparency means the important facts are visible and understood. A backlog full of vague requests is not transparent. A board that hides blocked work is not transparent. A status report that turns uncertainty into green check marks is not transparent.

Inspection means deliberately comparing results with goals and standards. Teams inspect progress toward the Sprint Goal, the quality of the Increment, stakeholder reactions, and the effectiveness of their working system. Inspection without adaptation becomes theater.

Adaptation means changing direction, scope, sequence, behavior, or method quickly enough to matter. The purpose is not constant turbulence. The purpose is to respond when evidence shows the current approach is inadequate.

THE FIVE VALUES IN PLANNING

Commitment is commitment to goals and quality, not obedience to an overloaded task list. Focus means concentrating on the most valuable objective. Openness means exposing risk and unfinished work. Respect means treating professionals as capable decision-makers. Courage means raising inconvenient facts early.

Value	Healthy planning behavior	Antipattern
Commitment	Protect the Sprint Goal and Definition of Done	Promise every requested item
Focus	Limit competing priorities	Start everything
Openness	Show uncertainty, blockers, and trade-offs	Green-status theater
Respect	Developers own the plan for the work	Managers assign every task
Courage	Stop, renegotiate, or escalate when facts change	Hide trouble until review day

FAKE AGILITY

Common warning signs include renaming project managers as Scrum Masters without changing authority, converting requirements into hundreds of tickets, measuring individual utilization, forcing velocity targets, and treating Sprint Planning as a commitment ceremony. These practices preserve command-and-control behavior while borrowing Scrum vocabulary.

Field Exercise: Transparency Audit

1. Choose one current team board or plan.
2. Mark every item whose status, owner, acceptance expectation, or dependency is unclear.

3. Identify one fact people are reluctant to expose.
4. Define one visible signal that would make the system more honest.

Knowledge Check

1. Why is inspection useless without adaptation?

2. How does focus affect work-in-progress?

3. Give one example of fake agility.

Chapter 3 — Accountabilities and Team Design

Scrum works through a small, cross-functional, self-managing team with three accountabilities.

LEARNING OBJECTIVES

Explain Product Owner, Scrum Master, and Developer accountabilities. Design a cross-functional Scrum Team. Avoid role confusion and hidden hierarchies.

3.1 One Scrum Team

The Scrum Team consists of one Product Owner, one Scrum Master, and Developers. These are accountabilities, not job titles or ranks. The team is collectively responsible for creating a valuable, useful Increment every Sprint. A healthy team normally has ten or fewer people because communication paths and coordination overhead grow quickly as the group expands.

Cross-functional means the team collectively possesses the skills needed to create value. Self-managing means the team decides internally who does what, when, and how. It does not mean the team ignores budgets, laws, architecture, contracts, or organizational constraints.

PRODUCT OWNER

The Product Owner is accountable for maximizing product value and effective Product Backlog management. This includes communicating the Product Goal, creating and clearly communicating Product Backlog items, ordering them, and ensuring the backlog is transparent and understood. Others may help, but accountability remains with the Product Owner.

A Product Owner is not a committee secretary. When several executives can independently reorder the backlog, the team has multiple competing Product Owners even if only one person holds the title.

SCRUM MASTER

The Scrum Master is accountable for establishing Scrum as defined and improving the Scrum Team’s effectiveness. The Scrum Master teaches, coaches, facilitates when helpful, and works to remove organizational impediments. The role is not meeting clerk, team boss, or ticket administrator.

DEVELOPERS

Developers create the plan for the Sprint, build quality into the Increment, adapt their plan daily toward the Sprint Goal, and hold each other accountable as professionals. Whoever performs the work needed to create the Increment is a Developer in Scrum, regardless of specialty.

Decision	Primary accountability	Collaboration required
Product Goal and backlog order	Product Owner	Stakeholders and Developers
How work is performed	Developers	Product Owner for clarity; Scrum Master for coaching
Scrum effectiveness	Scrum Master	Entire team and organization
Definition of Done	Scrum Team within organizational standards	Quality, security, legal, operations

Field Exercise: Team Design Map

1. Name the product — not the project.

2. List every skill required to create a usable Increment.
3. Identify skills outside the team that create handoffs.
4. Propose the smallest stable team that can reduce those handoffs.

Knowledge Check

1. Why are the Scrum accountabilities not job titles?

2. Who orders the Product Backlog?

3. Who creates the detailed Sprint plan?

Chapter 4 — Events, Artifacts, and Commitments

Every Scrum element exists to create focus and enable inspection and adaptation.

LEARNING OBJECTIVES

Map Scrum events to inspection and adaptation. Explain each artifact and commitment. Use timeboxes correctly.

4.1 The Sprint and Its Events

The Sprint contains Sprint Planning, Daily Scrums, development work, Sprint Review, and Sprint Retrospective. A new Sprint starts immediately after the previous one ends. Shorter Sprints create more learning opportunities and limit risk, but increase event frequency. The team should choose a length that supports meaningful Done Increments and keep it stable long enough to establish rhythm.

Sprint Planning initiates the Sprint by creating a Sprint Goal and initial Sprint Backlog. The Daily Scrum inspects progress toward the Sprint Goal and adapts the plan. The Sprint Review inspects the product outcome with stakeholders and adapts the Product Backlog. The Retrospective inspects how the team worked and plans improvements.

Event	Purpose	Max timebox (one-month Sprint)	Primary output
Sprint Planning	Set Sprint direction and initial plan	8 hours	Sprint Goal + Sprint Backlog
Daily Scrum	Inspect progress and adapt plan	15 minutes	Updated plan
Sprint Review	Inspect outcome and adapt future direction	4 hours	Product Backlog adaptations
Sprint Retrospective	Improve quality and effectiveness	3 hours	Actionable improvement

ARTIFACTS AND COMMITMENTS

The Product Backlog is an emergent, ordered list of what is needed to improve the product; its commitment is the Product Goal. The Sprint Backlog contains the Sprint Goal, selected Product Backlog items, and the Developers' plan; its commitment is the Sprint Goal. The Increment is the integrated, usable result; its commitment is the Definition of Done.

Artifact	Commitment	Planning function
Product Backlog	Product Goal	Maintains strategic direction and options
Sprint Backlog	Sprint Goal	Provides near-term focus and an adaptable plan
Increment	Definition of Done	Makes quality and completeness transparent

TIMEBOX RULE

A timebox is a maximum duration, not a requirement to consume every minute. End when the purpose is achieved.

Field Exercise: Event Health Check

1. Rate each Scrum event from 1 to 5 for purpose, participation, evidence, and adaptation.
2. Identify the event that behaves most like status theater.
3. Rewrite its agenda around the artifact and commitment it should inspect.

Knowledge Check

1. What three elements make up the Sprint Backlog?

2. Which event adapts the Product Backlog with stakeholders?

3. What commitment belongs to the Increment?

Part II — Product Direction

Deciding what is worth building, and in what order.

Chapter 5 — Product Vision and Product Goals

Strong Sprint Planning begins with a product direction worth pursuing.

LEARNING OBJECTIVES

Distinguish vision, outcomes, and outputs. Write a measurable Product Goal. Create a lightweight product roadmap.

5.1 Vision Before Backlog

A Product Backlog without a product direction becomes a queue of demands. Vision describes the change the product seeks to create and the people it serves. The Product Goal translates that direction into a concrete future state for the product. Only one Product Goal should be active at a time, though the backlog may contain ideas for later goals.

Outputs are things produced: features, reports, screens, forms, or integrations. Outcomes are changes in behavior, capability, risk, cost, revenue, safety, speed, or satisfaction. Scrum planning becomes powerful when outputs are treated as hypotheses for creating outcomes.

A PRACTICAL PRODUCT GOAL FORMAT

For *[target users]*, improve *[valuable outcome]* from *[baseline]* to *[target]* by *[time horizon or decision point]*, while maintaining *[critical guardrail]*. The format is optional; the discipline is not. The goal should be valuable, understandable, measurable enough to inspect, and broad enough to require multiple Sprints.

Weak statement	Stronger Product Goal
Build a project dashboard	Enable project leaders to identify schedule threats at least seven days earlier, reducing preventable missed commitments by 25% while maintaining data accuracy.
Launch mobile app	Allow field supervisors to complete daily reports in under five minutes with 90% same-day submission.
Improve onboarding	Increase the percentage of new users reaching their first successful outcome within one day from 35% to 65%.

ROADMAPS AS CHOICES, NOT PROMISES

A useful roadmap communicates outcomes, sequencing assumptions, and decision points. It does not pretend that every feature and date is known. Use Now / Next / Later, outcome horizons, or Product Goal sequences. Add confidence ranges and explicit assumptions where timing matters.

Field Exercise: Write the Product Goal

1. Name the user or stakeholder whose condition should improve.
2. State the current baseline.
3. Choose a target outcome and guardrail.
4. List three assumptions that could make the goal wrong.
5. Define how often the goal will be inspected.

Knowledge Check

1. What is the difference between an output and an outcome?

2. How many active Product Goals should a Product Backlog have?

3. Why can a detailed feature roadmap create false certainty?

Chapter 6 — Stakeholders, Users, and Value

Backlog order should reflect evidence about value, risk, learning, and cost of delay.

LEARNING OBJECTIVES

Identify stakeholders and users. Use discovery to reduce planning uncertainty. Prioritize by value and risk.

6.1 Stakeholders Are Not All Equal Requests

Stakeholders include customers, users, sponsors, operations, sales, security, legal, finance, support, and others affected by the product. The Product Owner listens broadly but orders one Product Backlog. The loudest request is not automatically the most valuable next move.

User research, analytics, operational data, interviews, support tickets, field observation, and prototypes turn assumptions into evidence. Discovery and delivery should interact continuously: discover enough to make the next responsible decision, deliver a usable slice, then learn from reality.

VALUE LENSES

Value may appear as revenue, cost reduction, risk reduction, safety, compliance, user satisfaction, learning, time saved, reliability, strategic positioning, or mission impact. A useful prioritization discussion makes the value hypothesis explicit rather than hiding it behind labels like “high priority.”

Factor	Question
User value	Whose problem becomes meaningfully better?
Business value	What measurable result matters to the organization?
Risk reduction	What uncertainty, safety exposure, or failure mode is reduced?
Learning value	What important assumption will this test?
Cost of delay	What is lost by waiting?
Effort and dependency	What capability and coordination are required?

SIMPLE PRIORITIZATION METHODS

Use methods as conversation aids, not mathematical truth. Cost of Delay divided by duration can help sequence time-sensitive work. RICE considers reach, impact, confidence, and effort. Value-versus-effort matrices are fast but coarse. Whatever method is used, the Product Owner remains accountable for order and should revisit assumptions as evidence changes.

Field Exercise: Stakeholder Evidence Map

1. List the five most influential stakeholders.
2. For each, record the outcome they care about and the evidence they use.
3. Identify one user group whose voice is missing.
4. Schedule one discovery activity before the next Sprint Planning.

Knowledge Check

1. Why should the loudest request not automatically come first?

2. Name three forms of product value.

3. What is the purpose of a prioritization method?

Chapter 7 — Building and Ordering the Product Backlog

The Product Backlog is a living decision system, not a warehouse of tickets.

LEARNING OBJECTIVES
Create clear Product Backlog items. Slice work vertically. Order the backlog around goals and learning.

7.1 Product Backlog Item Types

Product Backlog items may represent features, defects, experiments, technical improvements, risk controls, research, compliance work, or other changes needed to improve the product. Scrum does not require user stories. Use the format that creates sufficient shared understanding with the least waste.

A useful item has a clear purpose, enough detail for current decisions, an understood relationship to the Product Goal, and a size small enough to complete within one Sprint before selection. Items near the top need more clarity than distant options.

VERTICAL SLICING

Horizontal slices divide work by technical layer — database, API, interface, testing — and delay usable value until all layers are complete. Vertical slices create a thin end-to-end capability that can be integrated, tested, and inspected. A narrow but usable workflow is generally better than a large amount of partially complete infrastructure.

Large item	Possible vertical slices
Build project reporting	Submit one report type; attach one photo; supervisor approval; export one summary
Create marketplace	Search one category; view one listing; request contact; seller response
Automate invoicing	Generate draft from one contract type; validate rates; manager review; send approved invoice

ORDERING RULES

Order by the best current combination of value, urgency, risk, learning, dependencies, and effort. Do not create separate “technical” and “business” backlogs that compete for capacity; technical health is part of product value. Keep obsolete items out. The backlog should be long enough to preserve options, not so large that refinement becomes inventory management.

BACKLOG QUALITY TEST
Can the team explain why the top items matter now, how they support the Product Goal, and what evidence will be produced by completing them?

Field Exercise: Backlog Cleanup

1. Export or review the top 30 items.
2. Delete or archive items no longer connected to a plausible goal.
3. Split the three largest near-term items vertically.

4. Write the value hypothesis for the top five items.
5. Reorder the top ten using explicit trade-offs.

Knowledge Check

1. Does Scrum require user stories?

2. What is vertical slicing?

3. Why are separate technical and business backlogs risky?

Chapter 8 — Backlog Refinement and Readiness

Refinement prepares choices without turning uncertainty into excessive documentation.

LEARNING OBJECTIVES

Facilitate effective refinement. Create appropriately sized items. Use readiness as a guide rather than a gate.

8.1 Refinement Is Ongoing

Product Backlog refinement is the ongoing activity of breaking down and further defining items. It is not a prescribed Scrum event. Teams may hold scheduled sessions, refine collaboratively during the Sprint, use short discovery workshops, or combine approaches.

Refinement should improve shared understanding, expose dependencies, reduce item size, and support future selection. It should not become a pre-Sprint waterfall in which analysts fully specify work and hand it to Developers.

A PRACTICAL REFINEMENT CONVERSATION

Start with why the item matters and how success will be observed. Explore users, rules, examples, risks, quality implications, data, design, dependencies, and testability. Split the item until it can plausibly meet the Definition of Done within one Sprint. Size only after the team understands enough to compare it with other work.

Question	Purpose
What outcome does this support?	Connect work to value
What examples and edge cases matter?	Create shared understanding
What must be true before we start?	Expose dependencies and constraints
How will we know it works?	Clarify acceptance and evidence
Can it be Done within one Sprint?	Protect flow and feedback
What is still uncertain?	Preserve honesty

DEFINITION OF READY CAUTION

Scrum does not prescribe a Definition of Ready. A lightweight readiness checklist can help a team remember useful conditions, but a rigid gate can create silos, delay learning, and imply certainty that does not exist. The official standard is that items selected for Sprint Planning should be clear and small enough to be Done within the Sprint.

Field Exercise: Refine One Real Item

1. Choose one high-value item likely to be considered within two Sprints.
2. Write its purpose and success evidence.
3. Generate at least five concrete examples or acceptance scenarios.
4. Identify dependencies and risks.
5. Split it into the smallest valuable end-to-end slice.

Knowledge Check

1. Is refinement a formal Scrum event?

2. What is the danger of a rigid Definition of Ready?

3. When should an item be sized?

Part III — Sprint Planning

Turning direction into a realistic near-term forecast.

Chapter 9 — Estimation, Forecasting, and Capacity

Estimates support decisions; they are not promises or performance scores.

LEARNING OBJECTIVES

Compare relative and flow-based forecasting. Plan realistic capacity. Avoid velocity misuse.

9.1 What Estimation Can and Cannot Do

Estimation is useful when it improves a decision: whether an item fits, which option is smaller, what risk needs discussion, or what forecast range is plausible. Precision beyond available knowledge creates false confidence. Estimates should become less detailed as the planning horizon extends.

Relative estimation compares items using story points, sizes, or reference examples. Time estimation expresses likely effort or duration. Flow-based forecasting uses historical throughput and cycle time. None of these methods removes uncertainty; each creates a model that should be checked against actual results.

STORY POINTS AND VELOCITY

Story points may combine effort, complexity, and uncertainty. Velocity is the amount of points completed in a Sprint. It can help one stable team forecast its own work when the scale and Definition of Done remain consistent. It should not be used to compare teams, set quotas, reward individuals, or prove productivity. Once velocity becomes a target, point inflation becomes rational behavior.

CAPACITY PLANNING

Capacity is not headcount multiplied by available hours. Account for vacations, support duty, incidents, training, known meetings, onboarding, and specialist bottlenecks. Then protect slack for coordination and uncertainty. A team planned to 100% utilization has no ability to adapt and will carry work over at the first disruption.

Planning input	Example
Historical throughput	8-11 items completed per two-week Sprint
Known absence	One Developer unavailable for three days
Support load	Approximately 15% of team attention
Quality work	Two recurring regression activities
Uncertainty buffer	Reserve space rather than hide it in estimates

PROBABILISTIC FORECASTING

When enough historical data exists, Monte Carlo simulation can answer questions such as “How many items are likely to finish by this date?” or “By what date is this amount of work likely to finish?” Forecasts should be expressed as ranges with confidence levels, not single dates disguised as facts.

Field Exercise: Build a Capacity Forecast

1. Collect the last six Sprints of completed item counts or points.
2. Remove work that did not meet the Definition of Done.

3. Record known availability changes for the next Sprint.
4. Create a conservative, likely, and optimistic range.
5. State what could invalidate the forecast.

Knowledge Check

1. Why should velocity not be compared across teams?

2. What does 100% utilization do to adaptability?

3. What kind of answer does probabilistic forecasting produce?

Chapter 10 — Sprint Planning: Why, What, and How

Sprint Planning creates a coherent goal and an initial plan — not a pile of assigned tasks.

LEARNING OBJECTIVES
Prepare and facilitate Sprint Planning. Answer Why, What, and How. Create a realistic Sprint forecast.

10.1 Preparation

Good Sprint Planning begins before the meeting. The Product Owner ensures the Product Goal and top backlog items are transparent. Developers understand their likely capacity, technical condition, and Definition of Done. Stakeholder learning from the previous Sprint has already influenced the backlog. The Scrum Master helps the team improve the process and removes organizational obstacles.

Planning should include the entire Scrum Team. Other people may attend to provide advice, but they do not take over the team’s decisions. The Product Owner proposes how the product could increase value. The team then collaborates on the Sprint Goal.

TOPIC ONE: WHY IS THIS SPRINT VALUABLE?

The Sprint Goal gives the Sprint a single objective and creates flexibility in the exact work needed to achieve it. A goal such as “Complete tickets 101-115” is merely a list. A stronger goal states the valuable change or capability the Sprint should create.

TOPIC TWO: WHAT CAN BE DONE?

Developers select Product Backlog items through discussion with the Product Owner. Selection considers value, capacity, past performance, quality obligations, risk, and the Definition of Done. The result is a forecast, not a scope contract. The team should be able to explain how the selected items support the Sprint Goal.

TOPIC THREE: HOW WILL THE WORK GET DONE?

Developers plan the work necessary to create an Increment that meets the Definition of Done. The plan may include tasks, experiments, pairing, testing, integration, deployment, design, reviews, or coordination. Detail should be sufficient to begin; the plan will evolve throughout the Sprint.

Phase	Facilitating questions	Output
Orient	What changed? What is our capacity? What quality concerns exist?	Shared context
Why	What valuable objective is achievable now?	Draft Sprint Goal
What	Which items best support that objective?	Selected items
How	What initial approach and coordination are needed?	Actionable plan
Stress-test	What assumptions, dependencies, and failure modes threaten the goal?	Adapted forecast

PLANNING FINISH LINE

The Sprint Goal is clear, selected items plausibly support it, the team has enough plan to start, and everyone understands the Definition of Done.

Field Exercise: Run a 60-Minute Planning Simulation

1. Select a real Product Goal and top backlog items.
2. State capacity and known interruptions.
3. Draft three alternative Sprint Goals.
4. Choose one goal and select the smallest coherent set of supporting items.
5. Decompose only enough to reveal the first days of work and critical coordination.
6. Run a confidence vote from 1 to 5 and address low scores.

Knowledge Check

1. What are the three topics of Sprint Planning?

2. Who selects the Product Backlog items for the Sprint?

3. Why is the Sprint Backlog a forecast?

Chapter 11 — Writing Strong Sprint Goals

A Sprint Goal turns selected work into a focused mission.

LEARNING OBJECTIVES

Write outcome-oriented Sprint Goals. Use goals to make trade-offs. Recognize weak goals.

11.1 Characteristics of a Strong Goal

A strong Sprint Goal is valuable, focused, understandable, achievable within the Sprint, and flexible about the exact solution. It provides a reason for the selected work to belong together and a basis for daily decisions. It should survive modest scope changes without becoming meaningless.

The goal is created collaboratively by the Scrum Team and finalized before Sprint Planning ends. The Product Owner brings value context; Developers bring feasibility and delivery knowledge; the Scrum Master helps the conversation remain empirical and focused.

Weak goal	Why weak	Improved goal
Finish 25 points	Measures estimate consumption, not value	Enable supervisors to submit and approve a complete daily report from a phone
Complete login, settings, and profile tickets	Feature list without purpose	Allow new customers to securely create and recover an account without support
Work on performance	Too vague to guide trade-offs	Reduce checkout response time enough to support the seasonal traffic target

USING THE GOAL DURING THE SPRINT

At the Daily Scrum, ask what has been learned about progress toward the goal and how the plan should change. When unexpected work appears, evaluate whether it threatens or supports the goal. If a selected item becomes unnecessary but the goal remains achievable, the scope can change through collaboration between Developers and Product Owner.

Only the Product Owner can cancel a Sprint, and only when the Sprint Goal becomes obsolete. Cancellation should be rare because shorter Sprints naturally limit exposure.

Field Exercise: Goal Clinic

1. Collect the last five Sprint Goals used by the team.
2. Score each for value, focus, clarity, achievability, and flexibility.
3. Rewrite the weakest goal.
4. List three decisions the rewritten goal would help the team make.

Knowledge Check

1. Who creates the Sprint Goal?

2. Can scope change during the Sprint?

3. When may a Sprint be canceled?

Chapter 12 — Building and Managing the Sprint Backlog

The Sprint Backlog is owned and continuously adapted by Developers.

LEARNING OBJECTIVES
Build an actionable Sprint Backlog. Visualize work and risk. Manage scope while protecting the Sprint Goal.

12.1 What Belongs in the Sprint Backlog

The Sprint Backlog contains the Sprint Goal, selected Product Backlog items, and the actionable plan for delivering them. It should be visible enough to support real-time decisions. No single task format is required. Teams may use a board, checklist, workflow, dependency map, or combination.

The Sprint Backlog belongs to the Developers. Managers and stakeholders may view it, but they should not silently add tasks, reassign work, or convert it into an individual utilization tracker.

DESIGN THE WORKFLOW AROUND DONE

A workflow should expose states that matter: ready to begin, active, review, test, blocked, integrated, and Done. Too many columns create administrative noise; too few hide queues. Add work-in-progress limits when starting exceeds finishing. Make blocked work visibly painful rather than parking it in a quiet column.

Signal	Healthy response
Many items started	Stop starting; swarm to finish
Review/testing queue growing	Shift skills and pair across the bottleneck
Blocked item threatens Sprint Goal	Escalate immediately and replan
New urgent request	Product Owner and Developers evaluate trade-off against Sprint Goal
Item larger than expected	Split vertically or renegotiate scope while preserving value

TASKS ARE TEMPORARY HYPOTHESES

Tasks represent the current understanding of how to deliver. They may be added, removed, or reorganized as the team learns. Treating the original task breakdown as fixed defeats adaptation. What must remain transparent is progress toward the Sprint Goal and whether the Increment can meet the Definition of Done.

Field Exercise: Sprint Backlog Redesign

1. Sketch the current workflow from selection to Done.
2. Mark every queue and handoff.
3. Choose one work-in-progress limit.
4. Define how blocked work will be signaled and escalated.
5. Identify one specialist bottleneck and a cross-skilling action.

Knowledge Check

1. Who owns the Sprint Backlog?

2. Why can too much work-in-progress reduce delivery?

3. Are Sprint tasks fixed for the entire Sprint?

Part IV — Execution and Learning

Delivering a usable Increment and learning from it.

Chapter 13 — Daily Scrum and Flow

The Daily Scrum is a planning event for Developers, not a status meeting for management.

LEARNING OBJECTIVES

Facilitate a useful Daily Scrum. Use flow signals to adapt the plan. Resolve impediments outside the timebox.

13.1 Purpose and Structure

The Daily Scrum is a 15-minute event for Developers to inspect progress toward the Sprint Goal and adapt the Sprint Backlog. Developers choose any structure that achieves this purpose. The familiar three questions are optional and often become individual status reports.

A useful Daily Scrum looks at the goal and the work, not at each person in sequence. The board may be reviewed from right to left — starting with work closest to Done — to encourage finishing. The team identifies coordination needed today, then continues detailed problem-solving immediately afterward with the relevant people.

FLOW SIGNALS

Age of active work, blocked time, queue size, cycle time, work-in-progress, and progress toward the Sprint Goal reveal more than percentage-complete statements. When one item has been active far longer than normal, it deserves attention even if its owner says it is “almost done.”

Daily question	Why it matters
What changed in our understanding of the Sprint Goal?	Keeps planning empirical
Which item should we finish next?	Reduces work-in-progress
Where is flow blocked or aging?	Exposes risk early
Who needs to collaborate today?	Creates a concrete coordination plan
Does the Sprint forecast need adaptation?	Prevents late surprise

IMPEDIMENTS

An impediment is a condition that reduces the team’s effectiveness and cannot be resolved through ordinary self-management. The Scrum Master helps cause its removal, but the team should not wait passively. Track systemic impediments, their age, responsible escalation path, and recurrence. Repeated impediments are organizational design signals.

Field Exercise: Redesign the Daily Scrum

1. Observe one Daily Scrum without changing it.
2. Record how much time is spent reporting versus replanning.
3. Run the next Daily Scrum from the Sprint Goal and board, right to left.
4. Move detailed discussions to an immediate follow-up.
5. Compare whether risks surfaced earlier.

Knowledge Check

1. Who is the Daily Scrum primarily for?

2. Are the three traditional questions required?

3. Why examine aging work?

Chapter 14 — Quality, Definition of Done, and Technical Debt

Predictable planning is impossible when “Done” does not mean usable.

LEARNING OBJECTIVES

Create and evolve a Definition of Done. Integrate quality into planning. Manage technical debt transparently.

14.1 Definition of Done

The Definition of Done is the formal description of the state of the Increment when it meets the quality measures required for the product. It creates transparency about completeness. If a Product Backlog item does not meet the Definition of Done, it cannot be considered part of the Increment and returns to the Product Backlog for future consideration.

If the organization has mandatory quality standards, all Scrum Teams must follow them as a minimum. The team may strengthen the Definition of Done. Multiple teams working on one product must share a Definition of Done sufficient to produce one integrated Increment.

Quality area	Example criteria
Functional	Acceptance examples pass; no known critical defect
Engineering	Reviewed, integrated, automated checks pass
Security / privacy	Required scanning and data controls complete
Operations	Monitoring, rollback, support notes, and deployment readiness complete
User experience	Accessibility and supported-device checks complete
Documentation / compliance	Required records and approvals complete

DEFINITION OF DONE VS. ACCEPTANCE CRITERIA

Acceptance criteria describe conditions specific to one item. The Definition of Done describes quality conditions generally required for every Increment. A team should not copy the entire Definition of Done into every backlog item.

TECHNICAL DEBT

Technical debt is the future cost created by shortcuts, aging design, missing automation, weak architecture, or deferred quality. Some debt is deliberate and visible; hidden debt destroys predictability because future work becomes slower and riskier. Treat debt as product work, connect it to outcomes such as reliability or lead time, and order it transparently with other value.

CONSTRUCTION PARALLEL

A punch-list item is not “Done concrete” merely because the placement occurred. Completion depends on specified finish, curing, testing, documentation, protection, acceptance, and correction of nonconformance. The same principle — quality defines completion — applies even though the governing standards differ.

Field Exercise: Definition of Done Workshop

1. List every activity required before a product increment is genuinely usable and supportable.
2. Group items by functional, quality, security, operations, and compliance.
3. Remove item-specific acceptance criteria.
4. Choose the minimum standard that must apply now.
5. Identify one improvement to strengthen next Sprint.

Knowledge Check

1. What happens to work that does not meet the Definition of Done?

2. How do acceptance criteria differ from the Definition of Done?

3. Why does hidden technical debt damage forecasting?

Chapter 15 — Sprint Review and Product Adaptation

The Sprint Review is a working session for inspecting outcomes and deciding what to do next.

LEARNING OBJECTIVES
Prepare a collaborative Sprint Review. Gather actionable evidence. Adapt the Product Backlog and roadmap.

15.1 Beyond the Demo

A demonstration may be part of the Sprint Review, but the event is not a presentation ceremony. The Scrum Team and stakeholders inspect what was accomplished, discuss changes in the environment, examine progress toward the Product Goal, and collaborate on next steps. The Product Backlog may be adapted during the conversation.

Only Done work is part of the Increment. Unfinished work can be discussed honestly, but it should not be presented as completed value. The review should center on outcomes, evidence, and decisions rather than a tour of every ticket.

A STRONG REVIEW AGENDA

Begin with the Sprint Goal and what was learned. Inspect the working Increment in realistic scenarios. Compare expected and actual outcomes. Invite stakeholders to use, question, and challenge the product. Discuss market, operational, regulatory, schedule, and technology changes. Then update priorities and next experiments.

Review segment	Evidence
Goal outcome	Was the Sprint Goal achieved and what changed?
Product use	Working Increment in realistic scenarios
User / business result	Analytics, feedback, cost, risk, or behavior data
Environment	New opportunities, constraints, competitors, regulations, incidents
Adaptation	Updated Product Backlog order, experiments, or Product Goal thinking

STAKEHOLDER DESIGN

Invite people who can provide relevant evidence or make decisions. Prepare them to collaborate rather than merely watch. A giant audience can turn the review into performance theater; use focused product reviews, user sessions, or executive updates where necessary while preserving a real inspection-and-adaptation event.

Field Exercise: Design the Next Review

1. Write the three decisions the team needs help making.
2. Invite stakeholders connected to those decisions.
3. Choose realistic scenarios for inspecting the Increment.
4. Prepare outcome evidence, not just output lists.

5. Reserve at least one-third of the event for discussion and adaptation.

Knowledge Check

1. Is the Sprint Review only a demo?

2. Can unfinished work be counted in the Increment?

3. What should happen to the Product Backlog during or after the review?

Chapter 16 — Sprint Retrospective and Continuous Improvement

The Retrospective converts experience into deliberate improvement of quality and effectiveness.

LEARNING OBJECTIVES

Facilitate safe, evidence-based retrospectives. Select high-leverage improvements. Track whether changes work.

16.1 What the Team Inspects

The Scrum Team examines individuals, interactions, processes, tools, and the Definition of Done. It considers what went well, what problems occurred, and how those problems were or were not resolved. The team identifies the most helpful changes and may add them to the next Sprint Backlog.

A Retrospective should address the system, not assign blame. Psychological safety does not mean avoiding accountability; it means people can expose errors and risks without fear of humiliation or retaliation, allowing the team to solve the real cause.

FROM SYMPTOMS TO CAUSES

Use techniques such as Five Whys, causal mapping, timeline analysis, or a lightweight A3. Avoid stopping at “communication problem.” Ask what conditions made the communication failure likely: unclear authority, overloaded channels, delayed decisions, hidden queues, missing standards, or incentives that rewarded local optimization.

Retrospective step	Question
Set the stage	What conditions support honest participation?
Gather data	What actually happened?
Generate insight	What patterns and causes explain it?
Decide	What small change has the best leverage?
Close	Who will act, when will we inspect, and how will we know?

IMPROVEMENT WORK

Choose one or two improvements the team can truly implement. Give them owners, observable results, and an inspection date. Endless lists of good intentions create improvement inventory. Treat changes as experiments: state the problem, predicted effect, action, and evidence.

Field Exercise: Run an Improvement Experiment

1. Choose one recurring delivery problem.
2. Write a neutral problem statement using evidence.
3. Map at least three contributing conditions.
4. Select one change within the team’s influence.
5. Predict the measurable effect.
6. Inspect the result after one or two Sprints.

Knowledge Check

1. What does the Retrospective inspect?

2. Why is “communicate better” usually a weak action?

3. How many improvements should a team normally attempt at once?

Part V — Leadership and Application

Measuring what matters and scaling responsibly.

Chapter 17 — Metrics, Evidence, and Forecasting

Measure value, flow, quality, and capability — not activity for its own sake.

LEARNING OBJECTIVES
Choose balanced metrics. Use Evidence-Based Management concepts. Create honest forecasts and dashboards.

17.1 A Balanced Measurement System

No single metric represents product success. Combine outcome measures with delivery, quality, and capability measures. Metrics should trigger useful questions and decisions. When tied mechanically to rewards or punishment, they are likely to be manipulated.

Useful outcome measures include adoption, task success, revenue, cost avoided, safety exposure reduced, satisfaction, time saved, and mission impact. Delivery measures include cycle time, throughput, work item age, and release frequency. Quality measures include escaped defects, change failure rate, reliability, and rework.

Dimension	Example measures	Danger
Value	Outcome attainment, satisfaction, revenue, cost avoided	Vanity metrics without behavior change
Flow	Cycle time, throughput, work age, WIP	Using speed without quality
Quality	Defects, reliability, rework, recovery time	Hiding defects to protect score
Predictability	Forecast accuracy ranges, goal success rate	Treating forecast as contract
Capability	Automation, learning time, dependency reduction	Measuring activity instead of ability

EVIDENCE-BASED MANAGEMENT

Evidence-Based Management complements Scrum by focusing on value delivery under uncertainty. It uses goals, experiments, and measures across four Key Value Areas: Current Value, Unrealized Value, Ability to Innovate, and Time to Market. The point is not to build a giant scorecard; it is to make investment and improvement decisions using evidence.

BURNDOWN AND BURNUP

Burndown charts show remaining work; burnup charts show completed work and total scope. Either can be useful, but neither proves value. Scope changes, inconsistent sizing, and late quality work can distort interpretation. Pair progress charts with Sprint Goal status and flow data.

FORECAST COMMUNICATION

Communicate ranges and assumptions. For example: “Based on the last 20 Sprints, there is an 85% likelihood of completing 34 or more similarly sized items by October 30, provided team composition and Definition of Done remain stable.” This is more honest and actionable than a single unsupported date.

Field Exercise: Build a One-Page Product Dashboard

1. Select one Product Goal outcome measure.
2. Add two flow measures and two quality measures.
3. Add one capability measure.
4. For each metric, write the decision it should inform.
5. Remove any metric that exists only to make people look busy.

Knowledge Check

1. Why should metrics not be tied mechanically to rewards?

2. Name the four Evidence-Based Management Key Value Areas.

3. What is the difference between burnup and outcome measurement?

Chapter 18 — Multi-Team Planning, App Delivery, and Construction Adaptation

Scrum scales through one product direction, integrated quality, and reduced dependency — not through more meetings.

LEARNING OBJECTIVES
Coordinate multiple teams around one product. Apply Scrum to app development. Use Scrum responsibly in construction-related work.

18.1 Multiple Teams, One Product

When multiple Scrum Teams work on one product, they should share a Product Goal, Product Backlog, Product Owner, and integrated Increment. Each team may have its own Sprint Goal, but all work must integrate to a common Definition of Done. Separate component backlogs create local optimization and delayed integration.

Scaling problems are often dependency problems. Before adding coordination layers, simplify architecture, reduce handoffs, create cross-functional teams, standardize integration, and clarify product ownership. A meeting cannot compensate for organizational design that prevents teams from delivering independently.

APP DELIVERY PLAYBOOK

For a new app, begin with a Product Goal tied to a user outcome, not “build the whole app.” Create an ordered backlog of thin, testable workflows. Deliver a walking skeleton early: the smallest end-to-end path through interface, logic, data, deployment, and feedback. Use prototypes and experiments to test risky assumptions before scaling the build.

A strong early sequence might be: validate the problem, establish one secure user flow, instrument analytics, test the outcome with real users, then expand. Do not spend months building architecture for hypothetical scale before the product has demonstrated value.

Sprint	Possible objective for a field-planning app
1	A supervisor can create one project and submit one basic daily report
2	A manager can review, comment, and approve the report
3	The team can attach progress photos and identify missing reports
4	The product can surface one schedule-risk signal from submitted data
5	Pilot users can complete the workflow with measured time and error rates

CONSTRUCTION ADAPTATION

Scrum can support construction design coordination, prefabrication development, digital-tool creation, process improvement, closeout workflow redesign, safety innovation, estimating tools, and other complex knowledge work. It should not replace the project CPM schedule, contract milestones, pull plans, three-to-six-week look-aheads, daily huddles, inspection plans, or detailed work packages used to

control physical production.

For a concrete operation, Scrum may manage the improvement product — for example, developing a digital pre-pour readiness system — while Last Planner practices manage field commitments. The Product Backlog could contain capabilities such as inspection status, mix-design approval, embed tracking, pump confirmation, weather alerts, curing records, and test-cylinder documentation. The Definition of Done must respect ACI, ASTM, OSHA, contract, and engineer requirements applicable to the actual project.

Construction planning need	Best-fit primary approach	Possible Scrum use
Master milestones and critical path	CPM scheduling	Develop or improve supporting tools
Trade coordination and reliable weekly commitments	Pull planning / Last Planner	Experiment with coordination workflows
Daily production control	Daily huddles, production tracking, standard work	Build dashboards or solve systemic impediments
Complex design or digital product	Scrum	Primary delivery framework
Repetitive stable production	Lean flow / takt / standard work	Occasional improvement Sprint

LEADERSHIP RESPONSIBILITIES

Leaders create conditions for Scrum by clarifying product boundaries, empowering one Product Owner, funding stable teams, reducing dependencies, rewarding outcomes, and tolerating honest evidence. They should ask what was learned, which assumption changed, and what system impediment needs removal — not demand that every Sprint deliver a predetermined feature list.

Field Exercise: Select a 90-Day Implementation Target

1. Choose one product or complex workstream.
2. Define its Product Goal and accountable Product Owner.
3. Design a stable cross-functional team.
4. Choose a Sprint length and initial Definition of Done.
5. Identify three dependencies leadership must remove.
6. Select outcome, flow, and quality measures for the first 90 days.

Knowledge Check

1. What should multiple Scrum Teams working on one product share?

2. Why is a walking skeleton useful?

3. Name two construction-planning systems Scrum should not replace.

Part VI — Workbook and Implementation Tools

Copy these templates into your preferred tool. Keep them lightweight. A template is useful only when it improves a decision, conversation, or outcome.

Template 1 — Product Vision Canvas

Field	Working notes
Product / service name	
Primary users	
Current problem or unmet need	
Desired future condition	
Why our organization is positioned to help	
Critical guardrails	
Evidence that the vision matters	

Template 2 — Product Goal Canvas

Field	Working notes
Target user or stakeholder	
Baseline condition	
Target outcome	
Decision horizon	
Guardrail measures	
Key assumptions	
Evidence sources	
Product Goal statement	

Template 3 — Product Backlog Item

Field	Working notes
Title	
Purpose / value hypothesis	
Description or user scenario	
Acceptance examples	
Risks and dependencies	
Size / forecast input	
Evidence after completion	
Connection to Product Goal	

Template 4 — Refinement Checklist

Field	Working notes
Why does this matter now?	
Who benefits?	
Examples and edge cases	
Data / design / security considerations	
Dependencies and constraints	
How can this be sliced smaller?	
Can it meet the Definition of Done in one Sprint?	
Remaining uncertainty	

Template 5 — Sprint Planning Canvas

Field	Working notes
Product Goal context	
Capacity and availability	
Known incidents / support / interruptions	
Proposed Sprint Goal	
Selected Product Backlog items	
Initial delivery plan	
Critical dependencies and risks	
Confidence vote and concerns	

Template 6 — Sprint Goal Test

Field	Working notes
Sprint Goal	
Valuable outcome	
Users / stakeholders affected	
How selected items support it	
What scope could change while preserving goal?	
Evidence of achievement	
Failure conditions	

Template 7 — Definition of Done Workshop

Field	Working notes
Functional quality	
Engineering / technical quality	
Security and privacy	
Operations / release readiness	
Accessibility / user experience	
Documentation / compliance	
Organizational minimum standards	
Next planned improvement	

Template 8 — Daily Scrum Flow Board

Field	Working notes
Sprint Goal status	
Work closest to Done	
Aging work	
Blocked work and escalation	
Today's collaboration plan	
Forecast change	

Template 9 — Sprint Review Plan

Field	Working notes
Decisions needed	
Stakeholders to involve	
Sprint Goal and outcome evidence	
Realistic product scenarios	
Environmental changes	
Questions for stakeholders	
Product Backlog adaptations	

Template 10 — Retrospective Experiment

Field	Working notes
Observed problem	
Evidence	
Contributing conditions	
Proposed change	
Predicted result	
Owner and start date	
Inspection date	
Actual result and next decision	

Template 11 — Product Metrics Dashboard

Field	Working notes
Product Goal outcome measure	
Current Value measure	
Unrealized Value measure	
Time to Market / flow measures	
Ability to Innovate / capability measures	
Quality measures	
Forecast assumptions	
Decisions this dashboard supports	

Template 12 — Impediment Escalation Log

Field	Working notes
Impediment	
First observed	
Impact on goal / flow / quality	
Owner	
Immediate containment	
Root condition	
Escalation path	
Resolution and recurrence check	

90-Day Scrum Deployment Plan

Period	Primary objective	Key actions	Exit evidence
Days 1-15	Establish product and team	Define product boundary, Product Owner, Product Goal, team membership, Sprint length, baseline measures, initial Definition of Done	One accountable Product Owner; stable team; visible direction
Days 16-30	Launch first Sprints	Create ordered backlog, refine thin slices, conduct Sprint Planning, Daily Scrum, Review, Retrospective	At least one usable Done Increment and documented learning
Days 31-60	Stabilize flow and quality	Reduce WIP, expose aging work, strengthen Definition of Done, improve refinement, remove top dependencies	Improving cycle time, fewer carryovers, stronger stakeholder feedback
Days 61-90	Improve outcomes and system	Connect metrics to Product Goal, introduce probabilistic forecasting, run product experiments, address organizational impediments	Evidence-based roadmap decisions and repeatable Sprint rhythm

Deployment Checklist

- ☐ Name the product and user outcome clearly.
- ☐ Empower one Product Owner to order one Product Backlog.
- ☐ Create one stable, cross-functional Scrum Team.
- ☐ Select a Sprint length of one month or less and keep it consistent.
- ☐ Publish a practical Definition of Done before selecting work.
- ☐ Create thin Product Backlog items that can be Done within a Sprint.
- ☐ Use Sprint Goals rather than ticket quotas.
- ☐ Inspect product outcomes with real stakeholders every Sprint.
- ☐ Run one measurable improvement experiment at a time.
- ☐ Remove systemic dependencies rather than adding coordination bureaucracy.

Capstone Project — Plan and Run a Four-Sprint Product Cycle

Select a real product, app, service, or complex internal workflow. Produce the following portfolio of evidence.

Deliverable	Minimum standard
Product framing	Vision, users, baseline, Product Goal, guardrails
Product Backlog	At least 20 ordered items with clear value hypotheses; top items vertically sliced
Quality	Definition of Done and item-specific acceptance examples
Sprint Planning	Four Sprint Goals, capacity assumptions, selected items, initial plans
Execution evidence	Daily flow snapshots, impediment log, forecast adaptations
Sprint Reviews	Stakeholder evidence, product outcomes, backlog changes
Retrospectives	At least three improvement experiments and results
Metrics	Outcome, flow, quality, and capability dashboard
Final report	What changed, what was learned, and next Product Goal decision

Capstone Scoring Rubric

Dimension	Weight	Excellent performance
Product value and goal clarity	20%	Clear users, baseline, measurable outcome, and coherent Product Goal
Backlog quality and slicing	20%	Ordered around value and learning; thin end-to-end items
Sprint planning quality	20%	Strong goals, realistic forecasts, adaptable plans
Quality and integration	15%	Definition of Done consistently produces usable Increments
Evidence and adaptation	15%	Decisions visibly change based on stakeholder and flow evidence
Team-system improvement	10%	Retrospective experiments address causes and show measured results

Mock Final Examination

Choose the best answer. Recommended passing score: 80%.

1. The primary purpose of Sprint Planning is to:
 - A. Assign every task to an individual
 - B. Create a Sprint Goal and initial forecast for achieving it
 - C. Approve the Product Owner's fixed scope
 - D. Estimate the entire Product Backlog
2. Who is accountable for ordering the Product Backlog?
 - A. Scrum Master
 - B. Developers
 - C. Product Owner
 - D. Stakeholder committee
3. Which statement about the Sprint Backlog is correct?
 - A. It is fixed after Sprint Planning
 - B. It is a forecast owned and adapted by Developers
 - C. It contains only technical tasks
 - D. It is managed by the Scrum Master
4. Product Backlog refinement is:
 - A. A required two-hour weekly event
 - B. An ongoing activity, not a prescribed Scrum event
 - C. Performed only by the Product Owner
 - D. The same as Sprint Planning
5. A strong Sprint Goal primarily provides:
 - A. Individual utilization targets
 - B. A reason for the selected work to belong together
 - C. A guarantee that all selected items will finish
 - D. A substitute for the Product Goal
6. Work that does not meet the Definition of Done:
 - A. Can be counted if 90% complete
 - B. Can be shown as Done with a disclaimer
 - C. Is not part of the Increment
 - D. Must automatically enter the next Sprint
7. The Daily Scrum is primarily for:
 - A. Management status collection
 - B. Developers to inspect progress and adapt their plan
 - C. The Product Owner to reprioritize the Product Backlog
 - D. Stakeholder demonstrations
8. Velocity is most responsibly used to:
 - A. Compare teams
 - B. Set annual productivity targets
 - C. Support one stable team's internal forecasting

D. Calculate individual bonuses

9. Which is an outcome?

- A. Twelve features released
- B. A new dashboard designed
- C. Customer onboarding time reduced by 40%
- D. Thirty story points completed

10. Who can cancel a Sprint?

- A. Scrum Master
- B. Developers by majority vote
- C. Product Owner when the Sprint Goal becomes obsolete
- D. Any executive sponsor

11. The Sprint Review should primarily:

- A. Approve completed tickets
- B. Inspect outcomes with stakeholders and adapt future direction
- C. Evaluate individual performance
- D. Finalize the Retrospective action

12. A rigid Definition of Ready can be harmful because it may:

- A. Make work too small
- B. Create silos and false certainty
- C. Replace the Definition of Done
- D. Eliminate all dependencies

13. Cross-functional means:

- A. Everyone has identical skills
- B. The team collectively has the skills to create a usable Increment
- C. Departments attend the same meeting
- D. Developers report to multiple managers

14. The commitment associated with the Product Backlog is:

- A. Sprint Goal
- B. Definition of Done
- C. Product Goal
- D. Release date

15. Which behavior best reduces cycle time?

- A. Start more items
- B. Maximize specialist utilization
- C. Limit WIP and swarm to finish
- D. Increase point estimates

16. Technical debt should be:

- A. Hidden from the Product Owner
- B. Managed as transparent product work connected to outcomes
- C. Assigned only to a separate architecture team
- D. Ignored until release

17. Multiple Scrum Teams on one product should share:

- A. Separate Product Owners and backlogs
 - B. One Product Goal, Product Backlog, Product Owner, and integrated Definition of Done
 - C. Identical Sprint Goals
 - D. A single Daily Scrum
8. A probabilistic forecast should communicate:
- A. One guaranteed completion date
 - B. Ranges, confidence, and assumptions
 - C. Individual productivity
 - D. Only average velocity
9. The Retrospective should produce:
- A. A long list of complaints
 - B. One or two actionable improvement experiments
 - C. A revised Product Goal
 - D. Stakeholder approval
10. In construction operations, Scrum should generally:
- A. Replace CPM, pull planning, and safety planning
 - B. Ignore contract requirements to preserve adaptation
 - C. Complement established production controls for suitable complex work
 - D. Be used only for software

Answer Key and Rationales

#	Answer		#	Answer
1	B — Create a Sprint Goal and initial forecast for achieving it		11	B — Inspect outcomes with stakeholders and adapt future direction
2	C — Product Owner		12	B — Create silos and false certainty
3	B — A forecast owned and adapted by Developers		13	B — The team collectively has the skills to create a usable Increment
4	B — An ongoing activity, not a prescribed Scrum event		14	C — Product Goal
5	B — A reason for the selected work to belong together		15	C — Limit WIP and swarm to finish
6	C — Is not part of the Increment		16	B — Managed as transparent product work connected to outcomes
7	B — Developers to inspect progress and adapt their plan		17	B — One Product Goal, Backlog, Owner, and integrated Definition of Done
8	C — Support one stable team's internal forecasting		18	B — Ranges, confidence, and assumptions
9	C — Customer onboarding time reduced by 40%		19	B — One or two actionable improvement experiments
10	C — Product Owner when the Sprint Goal becomes obsolete		20	C — Complement established production controls for suitable complex work

Glossary

Term	Meaning
Acceptance criteria	Item-specific conditions used to clarify expected behavior or results.
Artifact	A Scrum object that represents work or value: Product Backlog, Sprint Backlog, or Increment.
Backlog refinement	Ongoing activity of breaking down and further defining Product Backlog items.
Capacity	The team's realistic ability to perform work after considering availability and obligations.
Cycle time	Elapsed time from starting an item until it is Done.
Definition of Done	Formal description of the quality state required for an Increment to be considered Done.
Developers	Scrum Team members accountable for creating any aspect of a usable Increment.
Empiricism	Making decisions based on observation and experience through transparency, inspection, and adaptation.
Forecast	A best current prediction that is expected to change with evidence.
Increment	A concrete, usable step toward the Product Goal that meets the Definition of Done.
Product Backlog	Emergent, ordered list of what is needed to improve the product.
Product Goal	Future state of the product that provides a target for planning.
Product Owner	Accountable for maximizing product value and effective Product Backlog management.
Scrum Master	Accountable for establishing Scrum and improving Scrum Team effectiveness.
Sprint	Fixed-length event of one month or less containing all Scrum events and product work.
Sprint Backlog	Sprint Goal, selected Product Backlog items, and Developers' delivery plan.
Sprint Goal	Single objective for the Sprint and commitment for the Sprint Backlog.
Throughput	Number of work items completed in a period.
Timebox	Maximum duration for an event.
Velocity	Amount of relative estimate completed by one team in a Sprint; useful only with caution.
Work-in-progress	Work started but not yet Done.

References and Further Study

1. Schwaber, Ken, and Jeff Sutherland. *The Scrum Guide: The Definitive Guide to Scrum*, November 2020. Current official Scrum definition available through [ScrumGuides.org](https://www.scrumguides.org).
2. Scrum.org. *Introduction to Sprint Planning*. Official learning resource covering Sprint Goal, selected Product Backlog items, Developers' plan, and the Sprint Planning timebox.
3. Scrum.org. *Product Backlog and Product Backlog Refinement* learning resources.
4. Scrum.org. *Definition of Done* learning resources.
5. Scrum.org. *Evidence-Based Management Guide*, May 2024.
6. Anderson, David J. *Kanban: Successful Evolutionary Change for Your Technology Business*.
7. Forsgren, Nicole, Jez Humble, and Gene Kim. *Accelerate: The Science of Lean Software and DevOps*.
8. Project Management Institute and Agile Alliance. *Agile Practice Guide*.
9. Lean Construction Institute. *Last Planner System* resources for production planning in construction.
10. Applicable OSHA regulations, contract documents, building codes, ACI standards, and ASTM test methods for field construction work.

The current official Scrum Guide remained the November 2020 edition at the time this course book was prepared in July 2026. Readers should verify future revisions before certification study or organizational policy updates.

**PLAN WITH PURPOSE.
DELIVER SOMETHING USABLE.
LEARN BEFORE THE COST OF
BEING WRONG GETS BIG.**

SCRUM PLANNING MASTERY · LONDON WULFE