



Scrum

Version Nov 2020 · verified 2026-08-06
Accountabilities, events, artifacts and
commitments from the 2020 Scrum
Guide — plus what Scrum never actually
defined.

Contents

Mental model	2
Empiricism: the three pillars	2
The Scrum Team	3
Artifacts and commitments	5
Definition of Done	6
The Sprint	8
Sprint Planning	9
Daily Scrum	11
Sprint Review and Retrospective	12
What Scrum does not define	13
Further reading	15

Mental model

Scrum is not a way of building software — it is a way of finding out, every few weeks, whether what you are building is the right thing. The framework is “purposefully incomplete”: it defines a team, a cadence, and three artifacts, and says nothing about how you estimate, design, or ship. Everything in it serves empiricism — make the work visible, inspect it against a goal, adapt. A ceremony that changes no decision is not Scrum, whatever the calendar invite says.

Empiricism: the three pillars

Pillar	Demands	Without it
Transparency	Work visible to all	Inspection misleads
Inspection	Progress checked often	Nothing to adapt to
Adaptation	Adjust as soon as you learn	Inspection is pointless

Scrum is founded on empiricism and lean thinking: knowledge comes from experience, decisions from what is observed. The pillars are a chain, not a menu — the guide is blunt that “inspection without transparency is misleading and wasteful” and that inspection without adaptation “is considered pointless”. The events exist to force that chain to complete on a fixed cadence, and the five values — Commitment, Focus, Openness, Respect, Courage — are what make people willing to be transparent at all.

Transparency -> makes inspection possible

Inspection -> makes adaptation possible

Adaptation -> changes the next Sprint

The Scrum Team

Accountability	Accountable for
Developers	The Increment and its quality
Product Owner	Product value; the Product Backlog

Accountability	Accountable for
Scrum Master	Scrum itself; team effectiveness

One team, typically 10 or fewer people, with no sub-teams and no hierarchy inside it. It is cross-functional — it holds every skill needed to create value each Sprint — and self-managing, deciding internally who does what, when, and how. The Product Owner is one person, not a committee, and for the accountability to mean anything “the entire organization must respect their decisions”.

1 Product Owner	one person, not a committee
1 Scrum Master	serves; does not assign work
Developers	anyone doing the work

Gotcha: The Scrum Master is not a project manager. Nothing in Scrum gives anyone authority to assign work to a Developer — the Developers select and plan it themselves, and “no one else tells them how to turn Product Backlog items into Increments of value”.

Artifacts and commitments

Artifact	Commitment	Answers
Product Backlog	Product Goal	Where the product goes
Sprint Backlog	Sprint Goal	Why this Sprint matters
Increment	Definition of Done	Whether it is real

Every artifact carries a commitment, so that inspecting it means something. The Product Backlog is an emergent, ordered list and “the single source of work

undertaken by the Scrum Team”. The Sprint Backlog is the Sprint Goal (why), the selected items (what), and the plan (how) — “a plan by and for the Developers”, updated throughout the Sprint as more is learned. Refinement is an ongoing activity rather than an event, and the Developers who will do the work do the sizing.

Product Goal

Product Backlog ordered, emergent

Sprint Backlog goal + items + plan

Increment meets Definition of Done

Definition of Done

Situation	Consequence
Item meets the DoD	An Increment is born
Item does not	Back to the Product Backlog
Organization sets one	Every team follows it as a minimum
It does not	The team must create one

Situation	Consequence
Several teams, one product	They share a single DoD

The Definition of Done is “a formal description of the state of the Increment when it meets the quality measures required for the product” – your quality bar, written down, applied to every item. It is what makes a Sprint’s output honestly inspectable: work that misses it “cannot be released or even presented at the Sprint Review”. Developers are required to conform to it.

Done - Checkout service (example)

Reviewed by another Developer

Unit + integration tests green in CI

Logs, metrics and an alert in place

On staging, no open P1 defects

Warning: “Done except testing” is not Done. It does not enter the Increment and it is not shown at the Sprint Review — it returns to the Product Backlog. A team that ships around its own Definition of Done has deleted the only quality check the framework has.

The Sprint

Aspect	Rule
Length	Fixed, one month or less
Next Sprint	Starts immediately after the last
Sprint Goal	No change that would endanger it
Quality	Does not decrease
Scope	Renegotiable with the Product Owner
Cancellation	Only the Product Owner may cancel

The Sprint is the container for every other event and the unit of inspection — “Sprints are the heartbeat of Scrum, where ideas are turned into value”. Fixed length is what

turns it into a measurement; a Sprint extended by three days to finish the work measures nothing. Shorter Sprints generate more learning cycles and cap the cost of being wrong, so one month is a ceiling, not a target. A Sprint may be cancelled only when its Sprint Goal becomes obsolete.

| Sprint 14 | Sprint 15 | Sprint 16 |
no gap, no hardening week, no Sprint 0

Gotcha: “Nothing changes during a Sprint” is a myth. Scope *may* be clarified and renegotiated with the Product Owner as more is learned. What is protected is the Sprint Goal, not the ticket list.

Sprint Planning

Topic	Question	Who decides
Why	Why is this Sprint valuable?	Whole Scrum Team

Topic	Question	Who decides
What	What can be Done this Sprint?	Developers, with the PO
How	How will the work get done?	Developers alone

Timeboxed to a maximum of eight hours for a one-month Sprint, and usually shorter for shorter Sprints. The Sprint Goal must be finalized before Planning ends — without one there is no objective to protect and nothing to renegotiate scope against, so the Sprint decays into a list of tickets. Decomposing selected items into work of a day or less is a common tactic the guide mentions, not a rule.

Weak: "Deliver PROJ-412, 417 and 419."

Good: "A returning customer can check out without re-entering card details."

Daily Scrum

Fact	Value
Timebox	15 minutes
For	The Developers
When	Same time and place, every working day
Structure	Whatever the Developers choose
Purpose	Progress toward the Sprint Goal

The event exists to produce “an actionable plan for the next day of work”. The Product Owner and Scrum Master take part only as Developers — that is, when they are actively working on Sprint Backlog items. It is also not the only time Developers replan; they meet throughout the day for more detailed discussions as needed.

Not: round-robin status for a manager

Yes: will we hit the Sprint Goal, and
what changes today if we won't?

Gotcha: The three questions (“what did I do yesterday...”) were removed in the 2020 Guide. A Daily Scrum run as a status report to the Scrum Master inverts the event: it is the Developers’ own replanning meeting, and nobody in it is reporting to anybody.

Sprint Review and Retrospective

Event	Timebox	Purpose
Sprint Review	4 hours	Inspect outcome, decide next
Sprint Retrospective	3 hours	Raise quality, effectiveness

Both timeboxes are maximums for a one-month Sprint and are usually shorter for shorter Sprints. The Review puts the Scrum Team and its stakeholders over the Increment together, and the Product Backlog may be adjusted on the spot — it “is a working session and the

Scrum Team should avoid limiting it to a presentation”. The Retrospective inspects individuals, interactions, processes, tools and the Definition of Done, and concludes the Sprint; the most impactful improvements may go straight into the next Sprint Backlog.

Review: team + stakeholders -> what next

Retro: the Scrum Team -> how we work

Gotcha: The Sprint Review is not a release gate. An Increment may be delivered to stakeholders before the Sprint ends — holding finished work back for a demo is a delay you chose, not one Scrum imposed.

What Scrum does not define

Not in the guide	What it actually is
Story points, velocity	An estimation practice
Backlog grooming	Older name for refinement

Not in the guide	What it actually is
Sprint zero, hardening	Every Sprint must be Done
The three daily questions	Dropped in 2020
Burn-down charts	One forecasting option
A specific tool or board	Scrum requires no tool

Of the terms above only burn-downs appears in the guide at all, named once beside burn-ups and cumulative flows with the caveat that such practices “do not replace the importance of empiricism”. Treat the rest as local choices you must justify on their own merits; they inherit no authority from Scrum. The framework is small on purpose — it “wraps around existing practices or renders them unnecessary”.

Warning: The reverse holds too. Dropping a part is not a lighter Scrum: “While implementing only parts of Scrum is possible, the result is not Scrum.” Name what you actually run, and own the trade-off instead of borrowing the word.

Further reading

- [The 2020 Scrum Guide](#)
- [Changes between the 2017 and 2020 guides](#)
- [Scrum Guide downloads and translations](#)