

A shared language for better teamwork

Company-Wide Agile Glossary Framework

Practical guide, starter definitions and a reusable worksheet

When someone says work is “ready”, “done” or “released”, does everyone understand the same thing? A useful glossary makes those meanings visible. It supports conversations, onboarding and decisions without pretending every organisation works in the same way.

Purpose

Use this guide to create and maintain a shared glossary for your team or organisation. Keep recognised framework meanings separate from your organisation’s working agreements.

Agile and Scrum, in plain language

Agile: an approach guided by values and principles that emphasise collaboration, useful outcomes and adapting as you learn [14].

Scrum: a framework that helps a team tackle complex work through short cycles of delivery, feedback and adaptation [1].

Three types of entry

Framework definition: a term with an authoritative meaning, such as Sprint in Scrum.

Common practice: a technique teams may choose, such as user stories.

Local agreement: a business term, board label or process that needs a definition in your context.

Use this guide: start with the relevant definitions on pages 2-6. Agree your approach on page 7, then copy the worksheet on page 8 for each new entry. Sources and revision notes are on page 9.

Begin small

Pick five terms that have caused confusion recently. Compare interpretations, agree plain-language definitions and add one useful example for each.

Edition 2.1 | Reviewed 11 October 2026

Created for Coded Female by Georgia. Examples are illustrative; they do not describe a real organisation’s policies or results. The glossary is a suggested working practice, not a required Scrum artefact.

Scrum: work, goals and quality

Framework definitions. Use these short explanations alongside the official Scrum Guide [1]. Local prompts are suggestions for your glossary, not extra Scrum rules.

Sprint [1]

A fixed period of one month or less for creating value.

Product Backlog [1]

An evolving, ordered list of product improvements.

Product Goal [1]

The future product state the team works towards.

Sprint Backlog [1]

The Sprint Goal, selected items and the Developers' delivery plan.

Sprint Goal [1]

The single objective for the Sprint.

Increment [1]

A usable product addition that meets the Definition of Done.

Definition of Done (DoD) [1]

The quality conditions an Increment must meet before it counts as complete. Teams on one product share this definition; applicable organisational standards set a minimum.

Local prompts: record your Sprint length, link to the relevant backlog and quality standard, and explain board labels. A two-week Sprint is one possible choice, not a universal requirement. Avoid using "backlog" when the reader needs to know which backlog you mean.

Scrum: people and conversations

Use the recognised names when describing Scrum. A local team nickname does not explain its accountabilities.

Product Owner (PO) [2]

Accountable for maximising product value and effective Product Backlog management. One person holds this accountability, even when work is delegated.

Scrum Master (SM) [3]

Accountable for establishing Scrum and the Scrum Team's effectiveness. Supports self-management and helps address impediments.

Developers [1]

The people creating a usable Increment, not only programmers.

Sprint Planning [1]

The Scrum Team plans the Sprint's purpose, work and approach.

Daily Scrum [1]

A 15-minute event for Developers to inspect progress and adapt their plan.

Sprint Review [4]

The Scrum Team and stakeholders inspect results and discuss what to do next. A working session, not only a demonstration.

Sprint Retrospective [1]

The Scrum Team plans improvements to quality and effectiveness.

Keep the distinction clear: refinement is ongoing work to clarify backlog items, not a prescribed Scrum event. Review is not a release gate. "Start, Stop, Continue" is one optional retrospective format [1].

Common practices, clearly labelled

These terms are useful in many teams. Their formats and local rules can vary. Do not present them as mandatory Agile or Scrum requirements.

User story [5]

A short description of a user need that supports a conversation about value. A common format is “As a [person], I want [capability], so that [benefit]”. Not every backlog item needs this format.

Epic [6]

A large user story that needs breaking into smaller stories. Tools and organisations may use “epic” differently, so record your local hierarchy.

Acceptance criteria

Observable conditions for a particular item to be accepted. Make the expected behaviour clear; choose a format that helps the conversation. This guide recommends keeping these separate from the shared quality standard.

Gherkin [7]

A language for structured examples, using keywords such as Given, When and Then. It can describe context, action and expected outcome. It is one option, not a compulsory format for every user story.

Definition of Ready (DoR) [8; 1]

An optional agreement about when an item has enough clarity to begin. It is not prescribed by the Scrum Guide. Use it to support discussion, rather than insisting every uncertainty is removed.

Velocity [9]

The total estimates attached to completed stories in an iteration, often expressed in story points. It may inform forecasting for that team. Do not use it to rank people or compare teams.

Local prompt: if you use story points, explain your estimation approach and what counts as complete. Avoid calling velocity a measure of customer value or turning a forecast into a delivery guarantee.

Flow and delivery terms

Kanban definitions below follow The Kanban Guide, May 2025 [10]. Agree the workflow boundaries before interpreting the numbers.

Work in progress (WIP) [10]

The number of items started but not finished.

Throughput [10]

The count of items finished per unit of time.

Cycle time [10]

Elapsed time from an item's defined start to its defined finish.

Work item age [10]

Elapsed time from an item's start to now, while it remains unfinished.

Service level expectation (SLE) [10]

A forecast combining elapsed time and probability. Use historical cycle times when available, or revise an initial best guess as evidence grows. It is not a guaranteed deadline.

Lead time | local measurement agreement

Record the start and end points you use, for example request received to customer delivery. State whether you count calendar or working days. Do not assume it means the same interval as cycle time.

Released | local availability agreement

Define when the intended audience can use a change. Deployment means installing it into an environment. A feature flag switches access on or off; a phased rollout makes it available to groups gradually. Record how these affect customer availability.

Illustrative SLE: "85% of items finish within eight calendar days." This shows a format, not a benchmark for your team. Pair delivery measures with evidence of whether users benefited.

Cross-team and scaling language

Include these entries only when your organisation uses them. None is a universal team structure or planning rule for Agile.

Squad, tribe and chapter

Treat these as organisation-specific labels. A squad may mean a delivery team, a tribe a group of teams, and a chapter a professional grouping across teams. Record the actual purpose, membership and authority of each locally.

Historical context [11]: Spotify's 2014 engineering culture material described an evolving approach with variation between squads. It should not be treated as a current, universal operating model for every organisation.

Planning Interval (PI) | SAFe [12]

In the Scaled Agile Framework (SAFe), a PI is a shared planning and delivery period for an Agile Release Train, a team of Agile teams. It is typically 8-12 weeks. The current term is Planning Interval; older material may say Program Increment.

PI Planning | SAFe [13]

A coordinated planning event for teams on an Agile Release Train to align their plans. It belongs to the SAFe context and is not required for all cross-team work.

Dependency and dependency tracker | local practice

A dependency is a relationship where progress relies on another item, decision or team. A tracker records those relationships. Agree who follows up, the needed date, the impact and the next action.

Business terms and abbreviations | local agreement

Add terms that affect requirements or reporting: "active customer", "completed order", project codes and tool names. Spell out abbreviations such as objectives and key results (OKRs) and key performance indicator (KPI). Define your calculations and time windows.

Local prompt: document your actual structure. Avoid assuming every squad has a Product Owner and Scrum Master, or that every epic maps to quarterly objectives.

Build it, use it and maintain it

The following is a suggested approach. Adapt it to your team's size and existing documentation practices.

1. Collect and agree

Collect ambiguous words from recent discussions. Involve the people who use them, across delivery and business roles. Compare interpretations using an example.

2. Keep one accessible reference

Choose an existing shared location, such as Confluence, SharePoint, Notion or a shared document. Check access, search and editing needs. Link entries from onboarding and relevant work.

3. Use proportionate ownership

Agree a maintainer and relevant reviewers. A small team may need one conversation; a larger organisation may need cross-team review. A central committee is optional.

4. Review when the meaning changes

Update entries when ambiguity appears or a process changes. Choose a periodic check if useful, but fix errors promptly. Record dates, sources and superseded meanings.

5. Introduce it through real work

Share the link, use it in a discussion and invite corrections. Quizzes are optional. Slash commands and automated chat searches require an integration to be set up.

A completed local entry: "Released"

Meaning: available to intended customers in the live service.

Example: customers can use the new password reset flow.

Excludes: code merged or deployed with access disabled.

Maintainer: product and delivery colleagues, as agreed.

Review trigger: the release process or intended audience changes.

Status: illustrative example, not an organisation's approved policy.

Your glossary entry

Copy this page for each term, or recreate these fields in your shared document. Keep framework definitions and local agreements visibly separate.

Term, abbreviation and full name

Category: framework definition / common practice / local agreement

Plain-language definition

Example and important exclusions

Local application and link to the actual policy or process

Source, framework edition and date checked

Maintainer and people who reviewed the definition

Status: draft / agreed / superseded | version and review date

Next action or review trigger

Before sharing: can someone new to the team understand the term? Is an example helpful? Are unresolved questions visible? Does a claimed standard or policy have a real source?

Sources and revision notes

Source pages checked on 11 October 2026. Older publication dates identify the referenced editions; they do not mean a newer edition was assumed.

[1] Scrum Guide

November 2020; framework definitions and accountabilities.

[2-4] Scrum.org explanations

Product Owner, Scrum Master and Sprint Review.

[5] Agile Alliance: User Stories

Common practice and purpose.

[6] Agile Alliance: Epic

Large user stories and decomposition.

[7] Cucumber: Gherkin Reference

Structured examples and keywords.

[8] Agile Alliance: Definition of Ready

Readiness agreements.

[9] Agile Alliance: Velocity

Calculation, forecasting and pitfalls.

[10] The Kanban Guide

May 2025; flow metrics and SLE.

[11] Spotify Engineering Culture, part 1

27 March 2014; historical context.

[12-13] Scaled Agile: Planning Interval and PI Planning

Current terminology and SAFe context.

[14] Agile Manifesto and principles

Agile values and guiding principles.

scrum.org: Scrum Master | scrum.org: Sprint Review | Scaled Agile: PI Planning

What changed from the 2025 document

Separated framework meanings from example company policies. Corrected Sprint and backlog wording. Clarified optional practices and velocity limitations. Updated PI terminology and qualified Spotify-derived labels. Removed assumed tools, compulsory approvals and unverified ISO-9001 alignment. Added flow terms, practical maintenance guidance and a worksheet. Edition 2.1 explains Agile, Scrum, quality conditions, feature flags and phased rollouts for newer readers.

Definitions are concise explanations, not quotations or replacements for the linked guides. Coded Female is not affiliated with or endorsed by the framework owners. Local recommendations and fictional examples are labelled throughout.