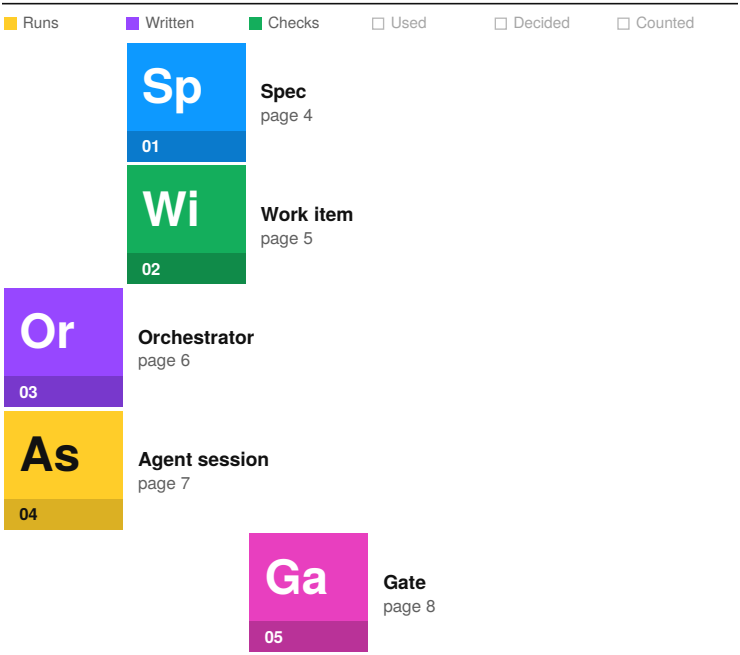
Software Factories

A pocket-sized guide to software delivery run by coding agents, as of October 2026.



My friend has a lot of questions.

What is a software factory?

When you use a coding agent, it works in turns. On each turn the harness sends the model its context and runs the tool calls the model writes. A session is all the turns from when you start the coding agent until you close it, as An Accidental Guide to: Agents, Harnesses, Skills, and Tools explains.

A coding agent can read a task, change files and run tests. A person usually starts each agent session, answers questions and checks the result. This works for one task at a time, but the person has to keep the work moving.

If you’re that person, your coffee often goes cold while you watch the agent work.

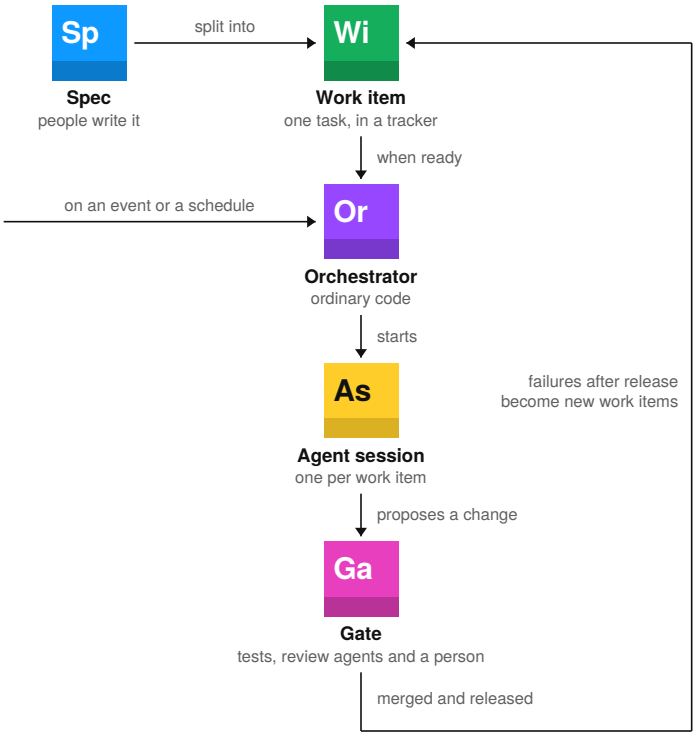
A software factory is a setup where agent sessions start without a person. The work is written down ahead of time as small tasks, and a program starts an agent session for each task once the task is ready. Each change then waits for checks and approvals before the team accepts it. People still decide what to build and which changes to accept, but they don't have to start or watch each session.

People still review every change in the factories described here, which is sometimes called "lights on" because the lights in a factory stay on so the people working there don't trip over the equipment. Factories where nobody reads the code have their own guide, An Accidental Guide to: Lights-Off Software Factories.

The setup has five parts, each with one job.

Term	Meaning
Spec	A written description of what to build, with tests that show it works.
Work item	One small task from the spec, sized for one agent session.
Orchestrator	The program that starts an agent session for each ready work item, on an event or a schedule.
Agent session	One session of a coding agent on one work item, started by a program instead of a person.
Gate	A point where work waits for required checks or approvals.

How do the parts fit together?



You start with a spec that says what the finished software must do. You split the spec into small work items and keep them in a tracker such as GitHub Issues because each agent session needs one task it can finish and test in one session.

On an event or a schedule, the orchestrator picks up a ready work item and starts an agent session for it. The session runs turn by turn, the way your own sessions do, but nobody started it or watches it. The orchestrator is ordinary code because every work item needs the same steps and the same retry limit, which says how many failed attempts are allowed before the work stops for a person.

Once the agent session proposes a change, gates hold it for tests and a person's approval. After the change ships, failures after release go back into the tracker as new work items.

Spec

Sp
01

Identify by	A reviewed requirements file or design that agent sessions read before work starts.
What	**A spec is a written description of what to build, with tests that show it works.** The spec gives every work item a shared source.
So what	**A mistake in the spec can reach every change built from it.** The product owner and engineers need to agree on the required behavior before they create work items.

Field marks

Owned by	**People.** A product owner writes the required behavior, and engineers approve the design.
Started by	**A product decision.** A new feature or a change in behavior often needs a spec before work begins.
Lasts	**As long as the decision applies.** A small spec may cover one change. A product spec may guide many changes.
Runs on	**The repo or a tracker.** A Markdown design in the repo and a requirements page linked from an issue are common forms.
Can	**Drive code and tests from one source.** The agent session and the acceptance tests use the same required behavior.
Cannot	**Cover what nobody wrote down.** Missing limits and failure cases leave the agent session to guess.

Confused with	Wi **Work item** A spec describes the intended result. A work item is one task taken from that spec for one agent session.

Work item

Wi
02

Identify by	An issue or task in a tracker, with enough detail for one agent session.
What	**A work item is one small piece of work from a spec.** The work item records the task and the result needed to finish it.
So what	**The agent session builds what the work item says.** A work item needs the relevant files, links and error details, along with a clear way to check the result.

Field marks

Owned by	**The team.** A product owner files the feature issue, and an engineer adds the technical details.
Started by	**A person or a failure.** You might file a feature issue. A failed test or monitoring alert files a repair issue.
Lasts	**Until accepted or abandoned.** A GitHub issue closes when its pull request merges, or when the team decides not to make the change.
Runs on	**A tracker.** GitHub Issues and other work trackers keep the task, links and current state together.
Can	**Be handed to one agent session.** An issue labeled ready starts one session with one clear stopping point.
Cannot	**Supply missing details.** A task that says fix login without an error or failing test leaves the session to guess.

Confused with

Sp **Spec**

A work item is one task sized for one agent session. Many work items may come from the same spec, so they all use the same description of the finished software.

Orchestrator

Or
03

Identify by	A workflow file in the repo that lists the events or schedule it responds to, and the program that runs when one arrives.
What	**An orchestrator is the program that moves work items through the factory.** The orchestrator starts agent sessions, sends proposed changes to gates and returns failed checks.
So what	**The orchestrator is ordinary code, so it follows the same steps every time.** The model works on the assigned task instead of choosing the next step, credentials or retry limit.

Field marks

Owned by	**The team.** The team sets the allowed repositories, credentials and retry limit.
Started by	**An event or the clock.** A new issue label, a failed CI run or a nightly schedule starts a run.
Lasts	**One run.** A scheduled run reads the current issues and pull requests, takes the permitted actions and exits.
Runs on	**A platform or worker.** A CI worker or scheduled service calls the tracker and the coding agent.
Can	**Move work between steps.** After a failed check, the orchestrator sends the error to a new agent session.
Cannot	**Supply a required approval.** A branch rule still requires a reviewer before the pull request merges.

Confused with	**Outer loop** An outer loop restarts one agent on the same goal, session after session, until the goal is met. An orchestrator starts a new agent session for each work item.

Agent session

As
04

Identify by	A run of a coding agent in a CI job or sandbox, for one work item, with a log and a branch.
What	**An agent session is one session of a coding agent on one work item, started by the orchestrator instead of by you.** It works the same way as a session you start, turn by turn, until it proposes a change or stops.
So what	**Each session starts fresh.** With only the work item, the repo and its instructions to go on, anything the session needs has to be written down in one of those places.

Field marks

Owned by	**The team.** The team chooses which coding agent, which instructions and what it can reach.
Started by	**The orchestrator.** A ready work item gets one session. A failed check can start another with the error attached.
Lasts	**One session.** The session ends when it proposes a change, gives up or hits its time limit.
Runs on	**A CI worker or sandbox.** The session gets a copy of the repo on a GitHub Actions runner or in a cloud sandbox.
Can	**Change files, run tests and open a pull request.** A session can fix a failing test, run the test and open a pull request with the change.
Cannot	**Remember earlier sessions or approve its own change.** The next session sees only what was written down, and a gate decides whether the change is accepted.

Confused with	**Main agent** The main agent is the loop the harness runs for your session, the one you start and watch. An agent session in a factory runs the same loop, but the orchestrator starts it and nobody watches it.

Gate

Ga
05

Identify by	A required check, approval or policy that work must satisfy before it moves on.
What	**A gate is a point where work stops until its required checks or approvals are complete.** Gates sit before work starts, before merge and before production.
So what	**A gate blocks work that lacks required tests or approvals.** You only need to step in when a gate requires your approval.

Field marks

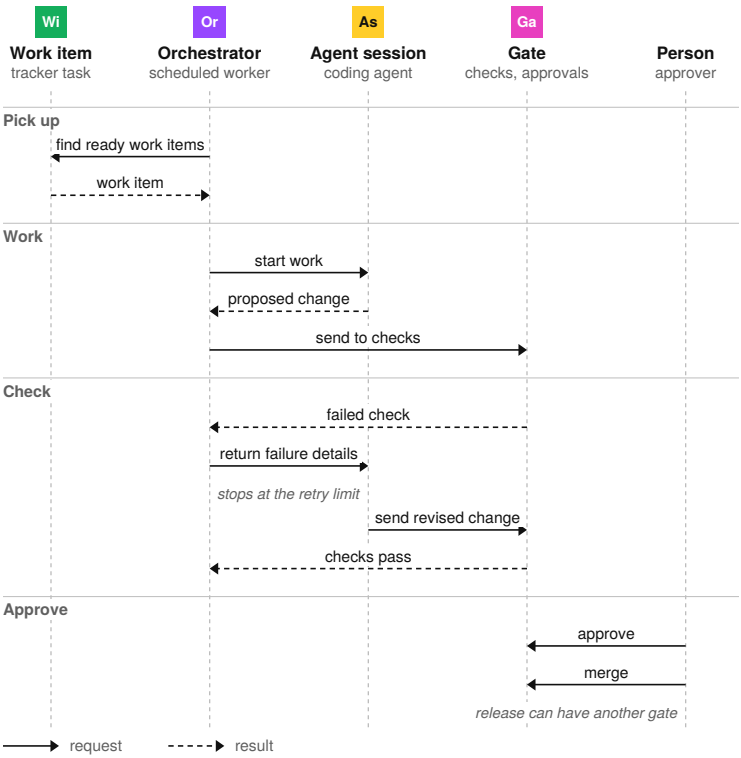
Owned by	**Repository or release owners.** A repository owner requires tests and a reviewer. A release owner requires approval before production.
Started by	**Work arriving.** A new issue enters triage, a pull request enters review and a build enters release approval.
Lasts	**Until the work passes or stops.** A failed pull request check returns the change for another attempt, up to the retry limit.
Runs on	**The tracker, pull request or release pipeline.** A branch rule blocks merge, and a deployment rule blocks production.
Can	**Stop work before the next step.** A triage gate holds an issue that has no steps for checking the result.
Cannot	**Prove the next stage will work.** A merged change may still fail during release or in production.

Confused with	**Check** A check reports pass or fail. A gate blocks the work until all required checks and approvals are complete. An Accidental Guide to: Harness Engineering covers checks.

One work item through a factory



The orchestrator keeps track of each work item until a gate accepts the change or the retry limit stops another attempt. When a check fails, the orchestrator sends the failure back with the same work item so the next agent session has both the task and the error.



Passing checks shows that a change meets the tests, but the tests may not cover every risk. A required approval gives a person one last decision before the pull request merges.

The five parts compared

	Owned by	Started by	Lasts
Sp **Spec**	People	A product decision	As long as the decision applies
Wi **Work item**	The team	A person or a failure	Until accepted or abandoned
Or **Orchestrator**	The team	An event or the clock	One run
As **Agent session**	The team	The orchestrator	One session
Ga **Gate**	Repository or release owners	Work arriving	Until the work passes or stops

	Runs on	Kept as
Sp **Spec**	The repo or a tracker	A requirements file or design
Wi **Work item**	A tracker	An issue or task
Or **Orchestrator**	A platform or worker	Code and configuration
As **Agent session**	A CI worker or sandbox	A log and a branch
Ga **Gate**	The tracker, pull request or release pipeline	Required checks and approvals

The spec and the orchestrator's code live with the software, so you can review and version them with the rest of the code.

More terms

Easy to mix up

Spec or work item?	A spec describes the intended result. A work item is one task from that spec, sized for one agent session.
Orchestrator or outer loop?	An orchestrator starts a new agent session for each work item. An outer loop restarts one agent on the same goal until the goal is met.
Orchestrator or orchestrator agent?	An orchestrator follows code. An orchestrator agent uses a model to choose the next step.
Agent session or main agent?	The main agent is the loop the harness runs for a session you start. An agent session in a factory runs the same loop, but the orchestrator starts it.
Gate or check?	A check reports pass or fail. A gate blocks work until all required checks and approvals are complete.

Related terms

Acceptance test	A test derived from the spec that checks the required behavior.
Check	A test or review that reports pass or fail. A gate blocks work until every required check passes.
Feedback path	A rule that turns a failed test or monitoring alert after release into a new work item.
Improvement loop	You change the instructions, skills or checks after an agent session makes a mistake. The change helps later sessions avoid the same mistake.
Outer loop	A loop that restarts one agent on the same goal, session after session, until the goal is met.
Review agent	An agent session that reads a change and reports findings without writing the change.
Retry limit	The number of failed attempts allowed before work stops for review.
Trigger	The event or schedule that starts an orchestrator run, such as a new issue label or a nightly time.

Sources

These public sources were checked in October 2026.

Matt Pocock, AI Coding Dictionary
September 2026 · aihero.dev/ai-coding-dictionary

Justin McCarthy, Software Factories
6 February 2026 · factory.strongdm.ai

Dex Horthy, Why Software Factories Fail
22 July 2026 · github.com/humanlayer

Birgitta Böckeler, Spec-driven development tools
2025 · martinfowler.com/articles

Kief Morris, Humans and agents
2026 · martinfowler.com/articles

Sp Wi Or As Ga

An Accidental Guide to…

Software Factories

A pocket-sized guide to software delivery run by coding agents, as of October 2026.
Written and reviewed by a person. AI does the grunt work. Set in TeX Gyre Heros and Newsreader.
Leave a star: accidental.info/star

My friend has a lot of questions.

oops@accidental.info · CC BY 4.0 · October 2026