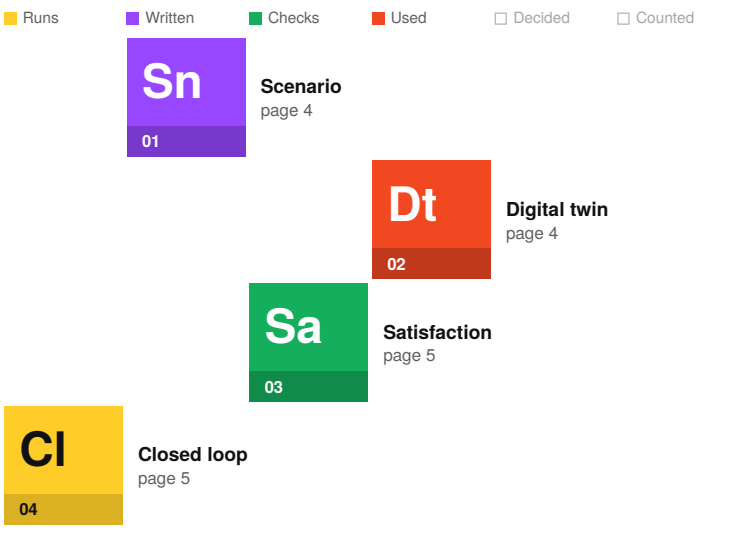
Lights-Off Software Factories

A pocket-sized guide to software factories where nobody reads the code, as of October 2026.



My friend has a lot of questions.

What is a lights-off software factory?

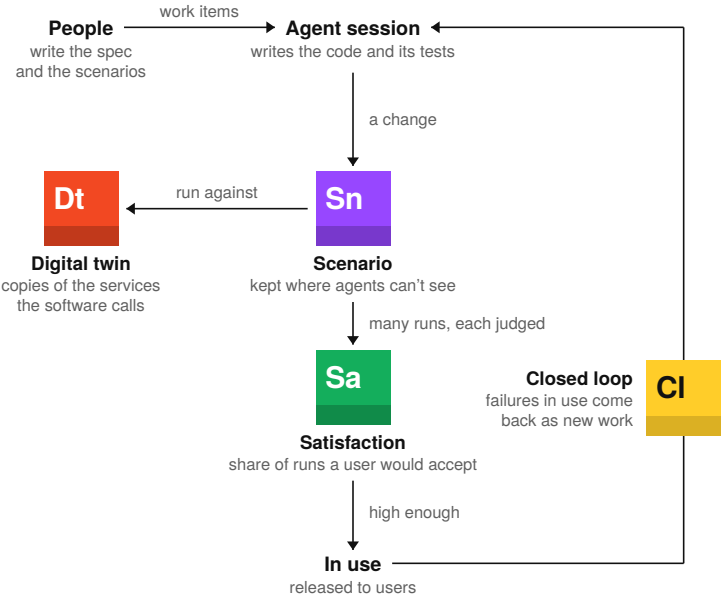
In a software factory, a program starts the agent sessions, and a person still reads each change and approves it at a gate. An Accidental Guide to: Software Factories covers that kind of factory.

A factory that runs with nobody on the floor can turn the lights off. A lights-off software factory is one where nobody writes or reads the code, sometimes called a dark factory.

At the gate, a person decides whether a change does what you wanted and whether it breaks anything. When nobody reads the code, these four parts make those decisions.

Scenario	A story of someone using the software from start to finish, kept where agent sessions can't see or change it.
Digital twin	A working copy of an outside service that scenarios can use thousands of times.
Satisfaction	The share of scenario runs that a model, asked to judge each run, says a user would accept.
Closed loop	The route from a failure in use back to a released fix with no person in it.

How do the parts fit together?



Agent sessions write the code. Each change runs against scenarios the sessions couldn't see, with digital twins in place of outside services. If the satisfaction score is high enough, the change goes to users. Nobody reads the code at any step.

Scenario

Sn
01

Identify by	A user story run from start to finish.
What	**A scenario is a story of someone using the software.** A judge model decides whether each run would satisfy that person.
So what	**Agent sessions never see the scenarios.** An agent can't write code that only passes a scenario it can't see.

Field marks

Owned by	Cannot
Product and engineering people.	**Cover a missing story.**

Confused with **Test**

An acceptance test checks one fixed result and often sits beside the code, where agent sessions can change it.

Digital twin

Dt
02

Identify by	A working copy of a service the software calls.
What	**A digital twin copies an outside service.** The copy is built from its API and checked against the real service.
So what	**Scenarios can run thousands of times.** There are no rate limits, usage costs or risks to real accounts.

Field marks

Owned by	Cannot
The factory team.	**Know behavior nobody observed.**

Confused with **Mock**

A mock returns canned answers for one test. A digital twin acts like the real service across many scenarios.

Satisfaction

Sa
03

Identify by	A percentage worked out from judged scenario runs.
What	**Satisfaction is the share of runs a judge model accepts.** The same scenario can pass on one run and fail the next.
So what	**You see how a change does across many user journeys.** One pass or fail tells you much less.

Field marks

Owned by	Cannot
Product and engineering people.	**Speak for real users.**

Confused with — **Passing tests**

Passing tests means each test got the one result it checks. Satisfaction says how often a judge model accepted results across many runs.

Closed loop

Cl
04

Identify by	Fixes that reach users with nobody approving them.
What	**A closed loop turns failures in use into new work.** Each fix then goes through the scenarios and out to users.
So what	**The factory can fix a failure after release.** Nobody has to file the fix or approve it.

Field marks

Owned by	Cannot
The team that runs the software.	**Recover from every failure.**

Confused with — **Feedback path**

A feedback path creates a work item, but a person approves the fix. A closed loop releases the fix without a person's approval.

What has been shown?

As of October 2026, three examples are public: a company that says it runs lights off, a startup that tried it and stopped, and a large build that came close.

A factory runs lights off	Since February 2026, one software company has said nobody writes or reviews its code. The company reports no defect rates, satisfaction numbers or shipped results.
Unread code stays maintainable	In July 2025, one startup stopped reading its code. By November, the code was so hard to change that a founder spent two weeks rewriting the core by hand. The startup now reads the code.
Agents can do almost all review	A team at a large AI lab spent five months building a product with no hand-written code and merged about 1,500 pull requests. Agents did most of the review. The team says it doesn't know how the design will hold up over the years.
The results are measured	No published study reports defect, incident or cost data from a factory where nobody reads the code.

Tip. When someone says their factory runs lights off, ask to see what it has shipped, the evidence that the software works, and the plan for failures the agents can't fix. An Accidental Guide to: Software Factory Claims explains how to read the answers.

Sources

These public sources were checked in October 2026.

StrongDM AI, Software factories and the agentic moment
February 2026 · factory.strongdm.ai/

StrongDM AI, Digital Twin Universe
February 2026 · factory.strongdm.ai/techniques/dtu

Dex Horthy, Why Software Factories Fail
July 2026 · github.com/humanlayer/advanced-context-engineering-for-coding-agents/blob/main/wsff.md

OpenAI, Harness engineering: leveraging Codex in an agent-first world
February 2026 · openai.com/index/harness-engineering/

Sn Dt Sa Cl

An Accidental Guide to…

Lights-Off Software Factories

A pocket-sized guide to software factories where nobody reads the code, 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