

THE AI FACTORY DOES NOT ELIMINATE THE FUNDAMENTALS

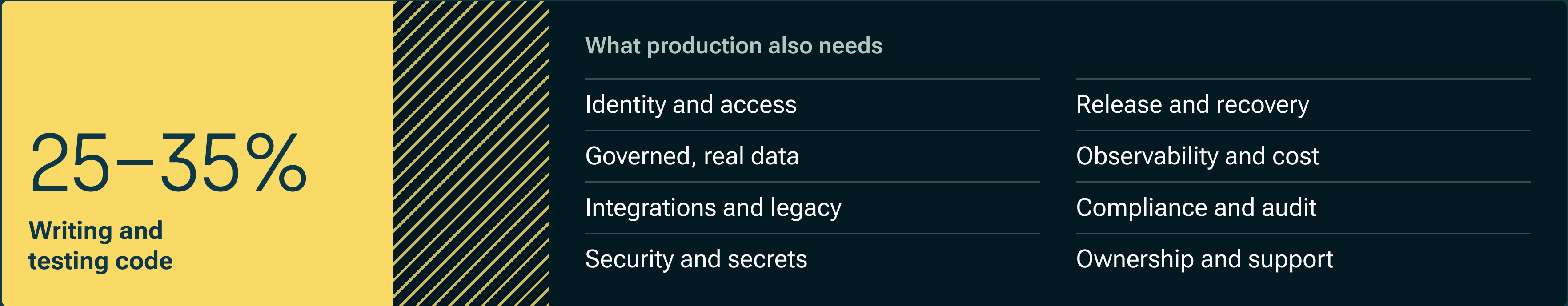
Sini Tistelgrén

Co-founder, COO and Tech Lead, Ambition

THE VIBE-CODED DEMO TOOK AN AFTERNOON.

WHY DOES PRODUCTION TAKE A QUARTER?

TIME FROM IDEA TO LAUNCH

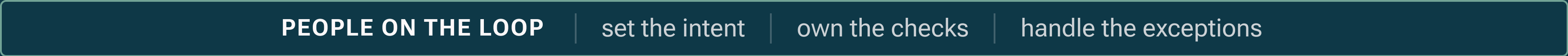


Time split: Bain & Company, Technology Report 2025. The list of what production also needs is my own.

“WE USE AI” CAN MEAN DIFFERENT THINGS

	THE VIBE-CODED DEMO	A TEAM WITH AI TOOLS	THE AI FACTORY
WHO WORKS	One person and a tool	Developers with assistants	Agents at every station
WHO CHECKS	The author, by eye	A colleague reads every change	Automated tests and rules the coding agent cannot change
THE RULES	None written down	In people’s heads	In the pipeline
SUCCESS	It worked once	More output per developer	Accepted changes in production

AN AI FACTORY IS A DELIVERY SYSTEM



SIX STATIONS. AGENTS CAN WORK AT EVERY ONE.



SIX FUNDAMENTALS, BUILT ONCE AND USED BY EVERY CHANGE



THE TEST

Can a routine change reach production without a person reading every line, and can you show afterward why that was safe?

The definition, the diagram and the test are my own. “In the loop” and “on the loop” are established terms.

ZERO LINES BY HAND.

THE EFFORT WENT INTO TESTS AND RULES.

OPENAI

1M lines of code in five months,
none of them written by hand

“...building software still demands discipline,
but the discipline shows up more
in the scaffolding rather than the code.”

ANTHROPIC

100K lines: a C compiler written
in about 2,000 agent sessions

“...it’s important that the task verifier
is nearly perfect, otherwise Claude
will solve the wrong problem.”

Lopopolo, Harness engineering: leveraging Codex in an agent-first world, OpenAI, February 2026 (“on the order of a million lines”). Carlini, Building a C compiler with a team of parallel Clauses, Anthropic, February 2026.

A SANDBOX PROVES THE AGENT.

YOUR ARCHITECTURE PROVES THE FACTORY.

	A NEW CODEBASE, BUILT FOR AGENTS	YOUR ENTERPRISE
CODE	One repository, designed for agents	Decades of systems, most designed long before agents
CONTEXT	Everything the agent needs is in the repository	Integrations, master data and business rules live in the ERP and in people’s heads
COUPLING	Nothing else depends on it	One change can touch the ERP and three other teams
DATA	Synthetic or public	Classified, and personal data stays inside your perimeter
ACCESS	One key, any model	An identity per agent, least privilege, approved models in your own tenancy
CHANGE	Push and see what happens	Every change recorded, tested, approved and verified

AN AGENT SEES ONLY WHAT IS WRITTEN DOWN

WHAT YOUR ARCHITECTURE HAS TO GIVE AN AGENT

INTERFACES	Contracts and contract tests for every integration, the ERP first	DECISIONS	Architecture rules the pipeline checks, with the reasons next to them
DOMAIN	Which system owns which data, and the business rules in plain words	SEAMS	Clean interfaces around the tightly coupled core, so agents work in bounded areas
ENVIRONMENTS	A test environment and anonymized data for each change, on demand	PERIMETER	Agent identities in your access management, approved models, logs in your monitoring

FROM OUR ASSESSMENTS A system deleted personal data on two different schedules. Nobody had written down why.

NONE OF THIS IS NEW ARCHITECTURE WORK. AI JUST MADE IT URGENT.

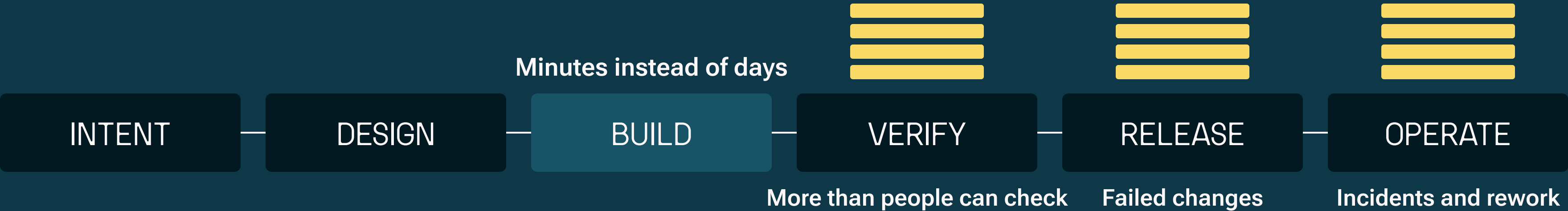
THE BLOCKERS WERE NOT THE MODEL

FROM OUR 2026 ASSESSMENTS One problem we found at each station.

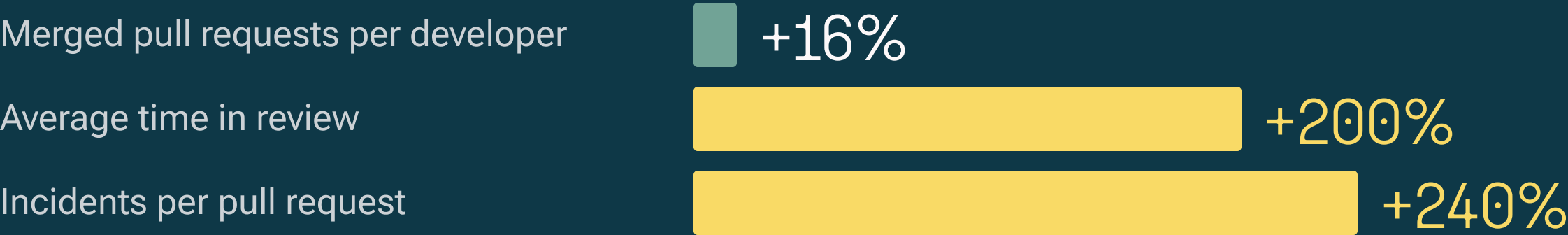
INTENT	DESIGN	BUILD	VERIFY	RELEASE	OPERATE
Requirements lived in meetings Requests passed from person to person in meetings, with no structured record.	A half-finished modernization Old and new systems ran side by side, with two versions of the rules.	Shadow AI filled the gap AI tools the company did not provide, and documents uploaded by hand to cloud AI services.	Tests came last, if at all Code was merged before its tests were designed. Breaks surfaced weeks later.	It was unclear who decides Most people surveyed did not know how to request a change, or where a request stood.	Errors never became checks Thin visibility into production, and no baseline to show what AI changed.
Missing Clear intent	Missing Sound architecture	Missing Security	Missing Independent verification	Missing Ownership	Missing Operational feedback

Aimbiton assessments, 2026. Anonymized, with details generalized. Each kind of problem shown was found in more than one organization.

COST AND RISK MOVE DOWNSTREAM



THE SAME ORGANIZATIONS, HIGH AI ADOPTION AGAINST LOW



“...our bottleneck became **human QA capacity.**”
OpenAI, Harness engineering, 2026

AI adoption goes with higher throughput **and higher instability.**
DORA, Google Cloud, 2025

Faros AI, The Acceleration Whiplash, April 2026: 22,000 developers, each organization compared with itself. Vendor telemetry.

1 INTENT

CLEAR INTENT

The model fills in whatever the request leaves out.

On tasks with key details removed, agents almost never asked a question unless they were told to.

Vijayvargiya et al., Ambig-SWE, ICLR 2026; six models

BUILD ONCE

Acceptance criteria before every build

ASK

Could two people read this requirement and build the same thing?

2 DESIGN

SOUND ARCHITECTURE

Agents repeat the patterns they find, good or bad.

AI gains in a pilot at Adidas

Loosely coupled, fast feedback



Tightly coupled to the ERP

Little or none

Gene Kim’s foreword to the DORA 2025 report

BUILD ONCE

Architecture rules that the pipeline enforces

ASK

Which architecture rules does a machine enforce, and which exist only in people’s heads?

3 BUILD

SECURITY

An agent reads untrusted content and acts with real credentials.

OpenAI on its own browser agent: prompt injection “is unlikely to ever be fully ‘solved.’”

OpenAI, Continuously hardening ChatGPT Atlas against prompt injection attacks, December 2025

BUILD ONCE

An identity and limits for every agent

ASK

What can the agent reach, and what is the worst it could do if it were tricked?

4 VERIFY

INDEPENDENT VERIFICATION

A change can pass every test and still be wrong.

AI changes that passed the tests, judged by maintainers



Would be merged

Would not: roughly half

METR, March 2026; 296 changes, adjusted to the human baseline

BUILD ONCE

Tests and rules the coding agent cannot change

ASK

What evidence do we have that does not come from whoever or whatever wrote the code?

5 RELEASE

OWNERSHIP

A tool cannot be accountable. A person who understands the system has to be.

Quiz after learning a new library

Coded by hand



With AI assistance



Shen and Tamkin, Anthropic, 2026; 52 engineers. Largest gap: debugging.

BUILD ONCE

Named owners and a record of every change

ASK

Who understands this system well enough to tell when the tool is wrong?

6 OPERATE

OPERATIONAL FEEDBACK

Production is where we find out what was true.

Small batches, frequent commits and frequent rollback strengthen the positive effect of AI.

Storer and DeBellis, Introducing the DORA AI Capabilities Model, Google Cloud, 2025

BUILD ONCE

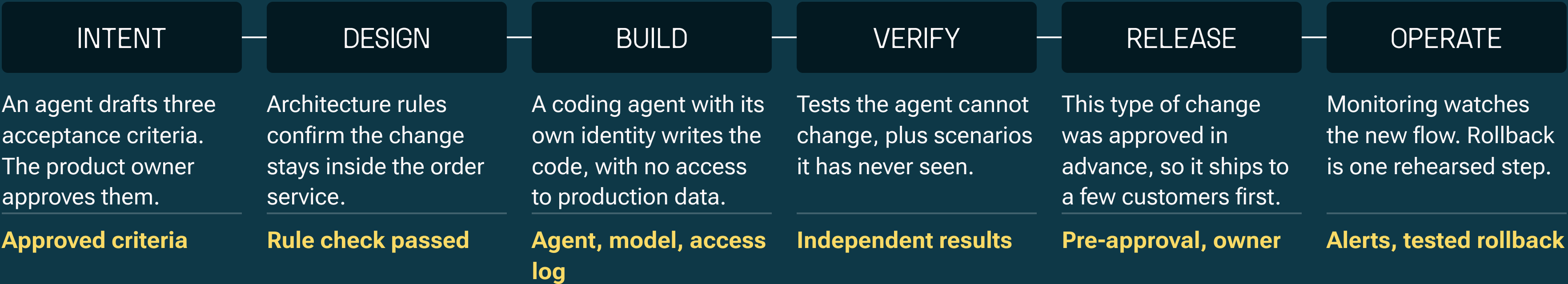
Monitoring, a rehearsed rollback, and someone who can stop the line

ASK

How quickly do we learn that a change was wrong, and how quickly can we undo it?

ONE CHANGE, START TO FINISH

THE REQUEST Customers can change the delivery address until the order ships.



PEOPLE DECIDED TWICE. EVERY STATION LEFT EVIDENCE.

A REGULATED ENTERPRISE ALREADY ASKS THESE QUESTIONS. IT CALLS THEM CONTROLS.

INTENT

Who asked for it?

DESIGN

Does it fit
the architecture?

BUILD

What made it,
with what access?

VERIFY

Who checked it?

RELEASE

Who accepted it?

OPERATE

Can we undo it?

MAKE THEM EXECUTABLE

Pre-approve a type of routine change

Each one ships on its evidence.
People decide the risky ones.

Whatever writes it never approves it

And no agent decides how risky
its own change is.

Every change carries its own record

So audit can test the pipeline itself
as a control.

Regulation (EU) 2022/2554 (DORA), Article 9(4)(e): change management based on a risk assessment approach. DORA research, Streamlining change approval.

AUTONOMY GOES ONLY AS FAR AS THE PROOF

LEVEL	SYSTEM	PROOF REQUIRED
5	Regulated systems	Evidence for every change
4	Connected systems	Interface contracts and digital twins
3	Product features	Acceptance scenarios the agent never sees
2	Internal tools	Tests the agent cannot edit
1	Web pages and prototypes	Preview and one-click rollback

A vibe-coded demo is level 1.
Most business systems are 3 to 5.

The rule: move up only when
the level below runs on the loop.

THE DEMO’S INTENT TRAVELS. ITS CODE GETS NO SHORTCUT.

Trust thermometer: Kalle Mäkelä, Aimbition, 2026. His levels 6 and 7, safety-critical systems and physical AI, are left out here.

THE ROLES STAY. THE WORK CHANGES.

	IN THE LOOP TODAY	ON THE LOOP IN A FACTORY
Product owner	Writes user stories	Writes criteria an agent can be tested against
Architect	Reviews designs	Turns patterns into rules the pipeline enforces
Developer	Writes and reviews code	Owns a service, directs agents, handles the exceptions
Tester	Tests at the end	Owns the checks and the scenarios agents never see
Platform team	Keeps the pipeline running	Runs the line: agent identities, limits, logs, rollback
Leader	Tracks output	Sets the autonomy level and owns the outcome

IF YOU LEAD ENGINEERING

Pick one loosely coupled value stream.
Measure it before any agent touches it.

Build one bounded loop: one owner,
one agent, one independent verifier.

IF YOU LEAD A BUSINESS

Replace output targets
with lead time and change fail rate.

Give the factory one owner,
with a budget and a mandate.

IF YOU ARE REGULATED

Design the first loop
with audit and security in the room.

Pre-approve one type of
routine change, with its evidence.

FOR ALL THREE: REINVEST PART OF THE TIME THAT AI SAVES

Put it into written intent, tests, architecture rules and evidence.

DORA, 2026: “Reinvest capacity, don’t reduce headcount.”

THANK YOU

THE AI FACTORY DOES NOT
ELIMINATE THE FUNDAMENTALS.
IT RUNS ON THEM.

Sini Tistelgrén

Co-founder, COO and Tech Lead, Aimbition

sini.tistelgren@aimbition.com

1 **INTENT** Clear intent
Could two people read this requirement and build the same thing?

2 **DESIGN** Sound architecture
Which architecture rules does a machine enforce, and which exist only in people's heads?

3 **BUILD** Security
What can the agent reach, and what is the worst it could do if it were tricked?

4 **VERIFY** Independent verification
What evidence do we have that does not come from whoever or whatever wrote the code?

5 **RELEASE** Ownership
Who understands this system well enough to tell when the tool is wrong?

6 **OPERATE** Operational feedback
How quickly do we learn that a change was wrong, and how quickly can we undo it?

A good answer is a name, a number or a document.

STATION	WHAT A REGULATED ENTERPRISE ADDS	WHERE THE RULE IS (EU FINANCIAL ENTITIES)
INTENT Clear intent	Every change traces back to an approved request with a named owner. The record keeps the criteria and who approved them.	DORA, Regulation (EU) 2022/2554, Article 9(4)(e). Delegated Regulation (EU) 2024/1774, Article 17(1)(c) and (d).
DESIGN Sound architecture	Data classification decides what may reach which model. Test environments hold anonymized, pseudonymized or randomized data. Production data there is an approved, reported and time-limited exception.	Delegated Regulation (EU) 2024/1774, Article 16(5)(a) and 16(6).
BUILD Security	The model provider is an ICT third party: a contract, data terms, an entry in the register of information and an exit plan.	DORA, Articles 28(3), 28(8) and 30. The exit strategy is required where the service supports a critical or important function.
VERIFY Independent verification	Whatever writes a change never approves it. The functions that approve a change are independent of those that request and implement it.	Delegated Regulation (EU) 2024/1774, Article 17(1)(b).
RELEASE Ownership	How far agents may go is a management decision, on the record. The management body approves and oversees the ICT risk framework.	DORA, Article 5(2). Outside the financial sector: NIS 2, Directive (EU) 2022/2555, Article 20(1).
OPERATE Operational feedback	Incident clocks apply when an agent caused the incident. First notice within 4 hours of classifying it as major, and within 24 hours of becoming aware of it.	Delegated Regulation (EU) 2025/301, Article 5(1)(a). NIS 2, Article 23(4): early warning within 24 hours.

My reading of the rules. DORA has applied to financial entities since 17 January 2025, and the entities listed in its Article 16 follow a simplified framework. Check the details with your legal team.

RESEARCH

1

Becker, J., Rush, N., Barnes, E. and Rein, D. (2025). Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. METR. arxiv.org/abs/2507.09089

2

Becker, J. et al. (2026). We are Changing our Developer Productivity Experiment Design. METR, 24 February 2026. metr.org/blog/2026-02-24-uplift-update

3

Whitfill, P., Wu, C., Becker, J. and Rush, N. (2026). Many SWE-bench-Passing PRs Would Not Be Merged into Main. METR, 10 March 2026. metr.org/notes/2026-03-10-many-swe-bench-passing-prs-would-not-be-merged-into-main

4

Vijayvargiya, S., Zhou, X., Yerukola, A., Sap, M. and Neubig, G. (2026). Ambig-SWE: Interactive Agents to Overcome Underspecificity in Software Engineering. ICLR 2026. arxiv.org/abs/2502.13069

5

Shen, J. H. and Tamkin, A. (2026). How AI Impacts Skill Formation. Anthropic, arXiv:2601.20245. Quiz averages from Anthropic’s summary, 29 January 2026: anthropic.com/research/AI-assistance-coding-skills

6

He, H., Miller, C., Agarwal, S., Kästner, C. and Vasilescu, B. (2026). Speed at the Cost of Quality: How Cursor AI Increases Short-Term Velocity and Long-Term Complexity in Open-Source Projects. MSR 2026. arxiv.org/abs/2511.04427

INDUSTRY RESEARCH

7

DORA (2025). State of AI-assisted Software Development. Google Cloud. The Adidas pilot is in Gene Kim’s foreword. dora.dev/research/2025/dora-report

8

Storer, K. M. and DeBellis, D. (2025). Introducing the DORA AI Capabilities Model. Google Cloud, 23 September 2025. cloud.google.com/blog/products/ai-machine-learning/introducing-doras-inaugural-ai-capabilities-model

9

DORA (2026). ROI of AI-assisted Software Development. Summary page. Google Cloud. cloud.google.com/resources/content/dora-roi-of-ai-assisted-software-development

10

DORA. Software delivery performance metrics. dora.dev/guides/dora-metrics
Streamlining change approval: dora.dev/capabilities/streamlining-change-approval

11

Bain & Company (2025). From Pilots to Payoff: Generative AI in Software Development. Technology Report 2025. bain.com/insights/from-pilots-to-payoff-generative-ai-in-software-development-technology-report-2025

12

Faros AI. The AI Productivity Paradox, July 2025: faros.ai/research/ai-productivity-paradox. The Acceleration Whiplash, April 2026: faros.ai/research/ai-acceleration-whiplash. The Speed Trap, 18 September 2026: faros.ai/research/speed-trap

ENGINEERING REPORTS FROM AI DEVELOPERS

13 Lopopolo, R. (2026). Harness engineering: leveraging Codex in an agent-first world. OpenAI, 11 February 2026. openai.com/index/harness-engineering

14 Carlini, N. (2026). Building a C compiler with a team of parallel Claudes. Anthropic, 5 February 2026. anthropic.com/engineering/building-c-compiler

15 OpenAI (2025). Continuously hardening ChatGPT Atlas against prompt injection attacks. 22 December 2025. openai.com/index/hardening-atlas-against-prompt-injection

HISTORY, WORKING MODEL AND FIELD WORK

16 Cusumano, M. A. (1991). Shifting Economies: From Craft Production to Flexible Systems and Software Factories. MIT Sloan School of Management, working paper 3325-91. dspace.mit.edu/handle/1721.1/2367

17 Allspaw, J. and Hammond, P. (2009). 10+ Deploys Per Day: Dev and Ops Cooperation at Flickr. O'Reilly Velocity. slideshare.net/slideshow/10-deploys-per-day-dev-and-ops-cooperation-at-flickr/1628368

18 A Conversation with Werner Vogels (2006). Interview by Jim Gray. ACM Queue, 4(4). queue.acm.org/detail.cfm?id=1142065

19 Mäkelä, K. (2026). Are Software Engineering “Factories” a Reality? Expert lecture, University of Jyväskylä, 28 September 2026.

20 Aimbition (2026). Assessments of software delivery and AI use in client organizations. Confidential. The examples in this talk are anonymized, with details generalized.

EUROPEAN UNION LAW

21 Regulation (EU) 2022/2554, Digital Operational Resilience Act, Articles 1, 5, 9, 16, 28, 30 and 64. eur-lex.europa.eu/eli/reg/2022/2554/oj

22 Commission Delegated Regulation (EU) 2024/1774, Articles 16 and 17. eur-lex.europa.eu/eli/reg_del/2024/1774/oj

23 Commission Delegated Regulation (EU) 2025/301, Article 5. eur-lex.europa.eu/eli/reg_del/2025/301/oj

24 Directive (EU) 2022/2555, NIS 2 Directive, Articles 4, 20 and 23. eur-lex.europa.eu/eli/dir/2022/2555/oj

25 Regulation (EU) 2024/1689, Artificial Intelligence Act, as amended by Regulation (EU) 2026/1744 (Digital Omnibus on AI). eur-lex.europa.eu/eli/reg/2024/1689/oj

26 Regulation (EU) 2024/2847, Cyber Resilience Act. eur-lex.europa.eu/eli/reg/2024/2847/oj

27 Directive (EU) 2024/2853, Product Liability Directive. eur-lex.europa.eu/eli/dir/2024/2853/oj