

Why late changes cost so much — and the part you control

AUTONOMOUS SYSTEMS · REQUIREMENTS & CHANGE

A mistake in what a system is supposed to do gets far more expensive the later it is found. Driverless systems cannot find all of theirs early — their safety rules require them to keep looking during testing and operation. So the part worth fixing is not when the change is found. It is how long it waits for someone to decide.

What I did: read the NASA cost study, a GAO review of 46 major programs, two autonomous-safety standards (ISO 21448, UL 4600), Waymo's two published safety-method papers, and the AI evaluation guides from OpenAI and Anthropic — and put them side by side.

21–78×

Cost to fix a requirements error first found in test, vs at the start (NASA)

29–1,615×

The same error first found in operation (NASA)

72% vs 11%

R&D cost growth, programs that changed requirements vs those that did not (GAO)

63%

Of 46 major programs changed requirements after development began (GAO)

 $L = \lambda W$

Open changes = changes arriving × time to decide (Little's law)

THE STORY, IN FOUR STEPS

1 — The curve

NASA measured it: fixing a requirements error costs a little early and a lot late. Found in test, 21 to 78 times more. Found in operation, far more again.

2 — The usual advice

So freeze requirements early and analyse them hard. For most systems, that is right. GAO's data points the same way.

3 — Why driverless is different

Safety standards for self-driving systems expect new hazards to be found in testing and on the road, and require the design to change when they are. Waymo calls this "discovery." You cannot freeze what you have not found yet.

4 — The part nobody measures

The cost curve records when a problem is found, not how long it then sits. A change found in test but decided after launch is paid for at the launch price.

THE ONE IDEA

You cannot choose when a driverless system reveals a requirement. You can choose how long it waits for a decision.

The backlog of undecided changes follows a simple law: the number open equals how many arrive times how long each waits. Halve the wait and you halve the backlog — without finding a single problem sooner.

THE SAME THING IN AI PRODUCTS

AI products also learn what "correct" means by testing. Every failure an evaluation finds is a late requirement. The teams that win are the ones that turn each one into an owner, a decision and a new test fastest.

WHAT I WOULD MEASURE

- **Time to owner** — found to a named decider.
- **Time to decision** — as a spread, not an average.
- **Age of open changes** — reviewed like schedule and risk.
- **Phase crossings** — changes found in one phase and decided in the next. The direct cost of waiting.
- **Reopen rate** — proof that speed is not buying bad decisions.

WHAT WOULD PROVE ME WRONG

If waiting adds no cost once you hold the discovery phase and complexity constant.

The late-change curve is real. For systems that must keep discovering their own requirements, the controllable part is the wait — decide fast, and know who decides.

The requirement you find in testing

ABOUT 90 SECONDS

THE HOOK

Everyone in engineering knows the chart: a mistake is cheap to fix early and expensive to fix late. NASA put numbers on it. A requirements error first found in testing cost twenty-one to seventy-eight times as much to fix as the same error caught at the start.

THE USUAL ANSWER

So the standard advice is: get the requirements right up front and freeze them. Government reviews of big programs say the same — programs that changed requirements after development started saw their development costs grow much faster.

THE TWIST

Driverless systems can't fully do that. Their own safety standards expect new hazards to be discovered in testing and on the road, and require the design to change when they are. Waymo literally calls it "discovery." Some requirements only exist once the system meets the real world.

THE INSIGHT

Here is what the cost chart doesn't show. It records when a problem was found — not how long it then sat waiting for a decision. A change found in test but decided after launch gets paid for at the launch price. And the number of open changes is simply how many arrive, times how long each one waits.

SO WHAT

You can't schedule when the system reveals a requirement. You can decide how long it waits. So I would measure time to owner, time to decision, and how many changes cross into the next phase while waiting — and drive those down. AI products have the same loop, just faster: every failure an evaluation finds is a late requirement.

CLOSE

Decide fast, and know who decides.

SOURCES

- NASA JSC, Error Cost Escalation Through the Project Life Cycle (INCOSE, 2004) · GAO-08-467SP, Assessments of Selected Weapon Programs (2008) · ISO 21448:2022 · Koopman, UL 4600 key ideas (2023) · Ratiu et al., Safety Performance Indicators (2024)
- Waymo, Safety Methodologies and Readiness Determinations (2020) · Waymo, Building a Credible Case for Safety (2023) · Little, L = λW (1961) · OpenAI, Evaluation best practices · Anthropic, Define success criteria and build evaluations