Speed and stability are not a trade-off: the case for small batches

21 October 2026 · 6 min read · blogs.iictp.org/speed-and-stability-engineering-governance

Many teams believe they must choose between shipping fast and shipping safely. The research behind the second TPBOK domain, Engineering Governance & Agile Delivery, points the other way: the practices that make delivery fast, mainly small changes and automated checks, are the same ones that make it stable.

TL;DR The short version

- Shipping fast and shipping safely are not a trade-off: small changes and automated checks give you both.
- The DORA research measures delivery with four numbers: deployment frequency, lead time, change failure rate and time to recover.
- Start by measuring the lead time of your last 20 changes and limiting work in progress.

Read online, listen to it, or change the text size: https://blogs.iictp.org/speed-and-stability-engineering-governance

The four measures that matter

The DORA research programme, summarised by Nicole Forsgren, Jez Humble and Gene Kim in Accelerate (2018), found that delivery performance can be described by four measures: how often you deploy, how long a change takes from commit to production (lead time), how often a change fails, and how quickly you recover when one does. The first two measure speed. The last two measure stability. In their data, the same teams tended to do well on both.

- **Deployment frequency.** How often you release to production.
- **Lead time for changes.** From code committed to code running.
- **Change failure rate.** The share of changes that cause a failure needing a fix.
- **Time to restore service.** How long recovery takes.

Why small batches work

A large change is hard to review, hard to test and hard to undo. When it fails, you do not know which part failed. A small change fails in a way you can see and fix quickly. There is also arithmetic behind this. Little's Law, proved by John Little in 1961, says that the average time work spends in a system equals the amount of work in progress divided by the rate at which it finishes.

> Example with made-up numbers: a team has 20 items in progress and finishes 5 a week. Average time in the system is 20 ÷ 5 = 4 weeks. Cut work in progress to 10 and, at the same finishing rate, it falls to 2 weeks. You did not work faster. You stopped starting so much.

Waterfall was never the plan

Winston Royce's 1970 paper is often cited as the origin of the strict sequential "waterfall". Read it and you find the opposite: he warned that a single sequential pass is risky and recommended building a pilot first so the hard problems appear early. The lesson is to find risk early, not to follow phases in order.

Define "done" so a reviewer can check it

A good quality bar is binary. Tests pass, the change is reviewed, security checks have run, the feature can be switched off. If "done" depends on someone's feeling, it will drift under pressure. Automating the bar, in a pipeline that blocks release when a check fails, replaces a meeting with a rule.

Anti-patterns

- **Water-Scrum-Fall.** Sprints in the middle, but requirements are fixed up front and release is a separate big event at the end.
- **Velocity inflation.** Points are used to compare teams, so they inflate, and the number stops meaning anything.
- **Long-lived branches.** Integration is deferred until it hurts.

What to do on Monday

1. Measure your lead time for the last 20 changes. Do not guess.
2. Limit work in progress so that finishing matters more than starting.
3. Pick one manual gate and automate it.

Engineering Governance & Agile Delivery is 28% of the ATP exam and 24% of the CTP exam. The specialist Agile Delivery Practitioner (ADP) goes deepest. See exam facts.

Sources and further reading

1. Forsgren, N., Humble, J. and Kim, G. (2018). Accelerate. IT Revolution Press.
2. Humble, J. and Farley, D. (2010). Continuous Delivery. Addison-Wesley.
3. Little, J. D. C. (1961). "A Proof for the Queuing Formula: L = λW." Operations Research 9(3).
4. Royce, W. W. (1970). "Managing the Development of Large Software Systems." Proceedings of IEEE WESCON.
5. Takeuchi, H. and Nonaka, I. (1986). "The New New Product Development Game." Harvard Business Review 64(1).

Published by IICTP, an independent technical credential body, on 21 October 2026. The current version of this post is at https://blogs.iictp.org/speed-and-stability-engineering-governance. More guides: blogs.iictp.org.