



BEYOND TWO SPEEDS

Two Speeds, One Broken Seam

Bimodal IT names a real tension and then freezes it into a caste system — a view from enterprise delivery

Two Speeds, One Broken Seam

Transformation · No. 204

Originally written December 2016 | Re-edited and re-rendered for the WhereBeyond Edition, 2025

“The organisation had built a sprinter and bolted it to a cargo ship, and then wondered why the timetable was set by the ship.”

1. The handoff that never happened on time

I watched a digital team do everything right. Two-week sprints, a working prototype of a new account-opening journey inside eight weeks, real customer feedback shaping it week by week. It was held up internally as the example — the proof that the organisation could finally move at the speed the market demanded. This was the fast half of the two-speed operating model everyone had been reading about: agile, cloud-hosted, unburdened by the old ceremonies.

Then the journey had to write a record into the core banking platform. That change joined a different queue entirely — a quarterly release window, a change-advisory board, an impact assessment that ran to dozens of pages. The eight-week build waited another four months to reach a customer. By the time it launched, the feedback that had shaped it was stale and two of the team had moved on. The organisation had built a sprinter and bolted it to a cargo ship, and then wondered why the timetable was set by the ship.

That experience is, in miniature, the whole case against bimodal IT — and I want to make that case plainly, because the idea is now everywhere, and it is the wrong answer to a question it nonetheless gets right.

2. The question is real

Give bimodal its due. The tension it names is not invented. An enterprise genuinely cannot run everything at one cadence. The thirty-year-old ledger that has to reconcile to the penny, sits inside a web of regulation, and cannot be down while the overnight batch runs is not governed the way a mobile front-end is governed, and it should not be. Anyone who has tried to impose a single delivery rhythm across both has watched it fail from one end or the other — either the front-end is throttled to the pace of the core, or the core is destabilised by the pace of the front-end. The instinct that these are different animals is sound.

So the popularity of the two-speed model is no mystery. It gave leaders a vocabulary for something they could feel but had not been able to name, and it gave them permission to let the digital teams run without first solving the far harder problem of the core. That permission felt like progress.

3. The answer builds a caste system

Here is where it goes wrong. Bimodal takes a real spectrum of cadence and freezes it into two permanent castes, and everything corrosive follows from that freezing.

- The fast side becomes where the talent, the visibility, and the interesting work live. The slow side becomes where you are sent, not where you choose to go.
- The two sides develop different tools, different language, and — fatally — different status. One is "innovation"; the other is "legacy", a word that does more quiet damage than any architecture diagram.
- And the boundary between them, the interface, becomes an orphan. It belongs to neither mode, is owned by neither team, and is exactly where every piece of real value has to cross.

Bimodal's fatal move is to treat the boundary between the two speeds as a given to be managed, when the boundary is precisely the thing that needs to be engineered away.

The model's defenders will say the two modes are meant to collaborate. In practice the structure fights the intention. You cannot tell one half of your organisation it is the future and the other half it is the past and then expect the handoff between them to be a partnership of equals. The account-opening journey did not stall for want of goodwill. It stalled because the two halves were built to run at different speeds and were never asked to close the gap — the gap was the design, not the defect.

4. What the model gets right, taken seriously

The strongest defence of bimodal is worth confronting head on, because it holds the truth any replacement has to preserve. It goes: some systems must be slow. A core that moves money cannot adopt "move fast and break things" without breaking things that must not break. Change control, predictability, and stability are not the timidity of an old guard; in a regulated, business-critical system of record they are the correct engineering response to the cost of being wrong. Force agility onto that and you get an outage, a mis-statement, or a letter from the regulator.

Every word of that is right. But notice what it actually argues for. It argues that cadence must vary with the cost of error — which is a spectrum, tuned system by system — not that the organisation must be cloven permanently in two. Bimodal takes a sound principle about risk-appropriate pace and hard-codes it into a permanent org chart. The principle is dynamic and graduated; the structure is static and binary. The failure is not in recognising that a ledger is not a landing page. It is in concluding that the people,

platforms, and careers behind them must therefore live in separate worlds.

"The right question is: at what pace should this particular thing change, given what it costs to get it wrong? — and that question has as many answers as you have systems, not two."

5. Cadence, not castes

If the two-speed model is the wrong answer, what is the better one? Not a single speed — that fails too. The alternative is to treat cadence as a graduated property of each system and to spend the effort where bimodal refuses to: on the seam.

1. Engineer the interface, not the queue. The account-opening story failed at the write to the core. The disciplined answer is to invest in stable, well-versioned interfaces — the kind the industry is now building with API layers and services in front of the core — so the front-end can move quickly against a contract without waiting on the core's release window. The core stays slow; the boundary stops being a wall.
2. Give the core its own path to safe speed. The slow-side world is not condemned to quarterly releases by nature. Automated testing, deployment pipelines, and the operational disciplines coming out of the DevOps movement let even conservative systems raise their safe cadence without lowering their standard of care. Slower than the front-end, yes; frozen, no.
3. Share ownership across the seam. The teams that build a journey should hold a stake in the systems it depends on, rather than throwing a change over a wall into someone else's backlog. "You build it, you run it" is uncomfortable across a core boundary, but even a partial version of it kills the orphaned-interface problem.
4. Refuse the word "legacy" as a caste marker. The systems of record are where the firm's money and memory live. A model that teaches your best engineers to treat them as a punishment posting is quietly hollowing out the most important capability you have.

None of this is free, and it is harder than drawing two boxes on a slide. Engineering a stable interface in front of a thirty-year-old core is real work, and raising the safe cadence of a conservative system is slower and less glamorous than standing up a greenfield digital team beside it. That difficulty is exactly why bimodal is attractive: it lets an organisation look fast without doing the unglamorous work at the seam. The bill for skipping that work simply arrives later, at the handoff, every single time.

6. The reckoning

Bimodal IT will not disappear soon, because it solves a political problem even as it worsens a technical one: it lets leadership be seen to back the future without doing the hard work on the core. That is precisely why it deserves the scepticism. It is a model that turns the interface — the one place value has to cross — into no one's job, and dresses a two-tier hierarchy up as an operating strategy.

The question bimodal asks is the right one, and I would keep it: not everything in an enterprise can or should move at the same speed. But the answer is a spectrum you tune and a seam you engineer, not a border you police. The organisations that pull ahead will not be the ones that ran their digital teams fastest. They will be the ones that made the distance between the fast thing and the slow thing short enough that it stopped deciding the timetable.