



For CEOs and PE Partners

Spec-Driven Development

The velocity is real. It outran everything built
to catch it.

Prepared by **Grant Howe, Managing Partner**

August 2026

Geekbyte

PE Portfolio Technology Leadership

The Gap You Are Paying For

Your portfolio companies are paying for AI coding tools, and the tools are working. Engineers are faster, and by 2025 the organizational throughput number had turned positive too. What did not follow is stability. More is reaching production, and more of what reaches production breaks.

84%	+98%	-7.2%	Reversed
of developers use or plan to use AI tools	pull requests merged per developer	delivery stability, per 25% more AI adoption (2024; still negative in 2025)	throughput fell 1.5% in 2024, rose by 2025 — stability did not follow

Sources: 2024 and 2025 DORA State of AI-assisted Software Development reports; Faros AI study of 10,000 developers; 2025 Stack Overflow Developer Survey.

That asymmetry is the finding. Faros named the volume half of it the AI Productivity Paradox — individual engineers really are faster on the task in front of them. The half that has not recovered is everything downstream of writing the code: review, verification, the judgment about whether what was produced is correct. Speed came back. Confidence did not.

If you have asked why release cadence improved while incidents did not, and heard an answer about tooling maturity or hiring, this is the gap that answer is describing.

Why Tools and Hiring Won't Close It

When humans wrote every line, writing was the constraint, and everything downstream — review, testing, approval to ship — was sized to human speed. AI removed the constraint it was pointed at and left everything downstream at its original capacity. More tools push more volume into the

39 pts

the distance between
how fast developers felt
and how fast they were

*METR study, cited in the 2025
DORA report*

same review queue. More engineers add reviewers who need context nobody wrote down.

The scarce resource is no longer writing code. It is deciding what should be written and verifying what came back.

It is also why asking the team is not a measurement. In one study cited by the same DORA report, developers who had been slowed 19% by AI tools believed those tools had made them 20% faster. Self-report and the number you report on are two different things, and only one of them shows up in the P&L.

That is a process problem, and left alone it compounds: the instability stays priced into every release, architecture drifts because nobody records why decisions were made, and the knowledge that used to live in a few senior engineers' heads now has to cover a codebase they did not personally write.

What Changes

Spec-Driven Development puts structure where the constraint now sits. Work does not start until what is being built — and why — is written down. Work does not advance without a named owner signing off in writing. The reasoning behind every decision is captured as it happens, not reconstructed after someone leaves.

For the engineering team, that is a methodology. For the owner, it is three things that show up in the value of the asset.

What the Owner Gets	What Actually Changes
Speed stops costing stability	The review lane is designed for machine volume instead of inheriting a size set for human volume, so shipping faster stops meaning shipping more defects.
Key-person risk drops	The reasoning behind the architecture lives in the repository, not in two senior engineers. That changes what a resignation costs you mid-hold.
The process documents itself	The written record is a by-product of shipping, not a documentation project someone funds later under deal pressure.

It is not a rewrite, a re-platform, or a reorganization. It maps onto the team that exists — whether that is eight engineers or eighty — and is adopted incrementally alongside the current process.

How This Reads at Exit

I have sat on both sides of technical diligence across 27+ M&A integrations — as the CTO answering the questions and as the acquirer asking them. The requests are standard every time: process documentation, defect history, architectural decision records, evidence that changes are reviewed before they ship.

What varies is whether the company answers from artifacts that already exist, or assembles something in three weeks while the deal team watches. The second version is expensive twice: it consumes engineering leadership at the worst possible moment, and thin answers become diligence findings, which become negotiating positions.

An organization running spec-driven produces those answers as exhaust from normal operation — the specs, the signed approvals, the traced defects, the recorded rationale all exist because the process created them on the way to shipping.

Provability starts counting twelve to eighteen months before a sale process. By the time the bankers are in the room, you are not building a record — you are assembling one.

That is why this is a hold-period decision, not an engineering preference. The value is realized in a room the engineering team will not be in.

Six Questions You Should Be Able to Answer About Your Company

You do not need to evaluate a methodology. You need to know whether you can answer these about your own company. Most owners discover they are relying on one person's memory for several of them.

- When we ship something significant, where is the written record of why we built it that way?
- Who signs off before code reaches a customer, and what do they leave behind?
- The last defect that reached a customer — which review step should have caught it, and what changed afterward?
- Is our constraint writing code, or reviewing and verifying what gets written?
- If our two most senior engineers resigned this quarter, how much of what they know is written down?
- If a buyer asked for our process documentation and defect history tomorrow, what would we hand them?

What a good answer sounds like: here is the document, here is the record, here is what we changed. Answers describing intentions, tooling, or plans to improve are telling you the record does not exist yet.

How to Find Out

The methodology is open source under AGPL-3.0 — any team can read it and adopt it without us. Geekbyte's value is the operator sitting alongside the CTO while it lands, which is where these efforts usually fail.

My background is 27+ M&A integrations and four PE exits as a CTO and SVP inside portfolio companies; the hard questions — what to formalize first when half the team is contractors, how to sequence against a committed roadmap — sit in operator judgment, not in a document.

Engagement shape: a two-week diagnostic that ends in a written read you can act on or shelve. If the read says go, a ninety-day implementation — owner training, the first work run end-to-end with coaching, and measurement in place so next quarter's conversation is about numbers.

The Next Move

Pick one company. Thirty minutes. We walk through how this would land there specifically.