

November 25, 2026

Beyond Delivery: Measuring What Matters

BEST PRACTICES

Written by: David Stevens, Director of Business Development

Getting software into production matters. Deadlines matter. Budgets matter. Teams should be accountable for what they commit to delivering. But there's a problem with using delivery itself as the primary definition of success: software can ship exactly as planned and still fail to produce the outcome the business expected. A project can be on time, on budget, and technically sound while employees continue using spreadsheets and workarounds. A new workflow can automate a process without making it meaningfully faster. Delivery tells you whether the team completed the work. It doesn't necessarily tell you whether the work mattered.

Define Success Before Development Begins

The best time to decide how a software initiative will be measured isn't after launch. It's before development begins. Start with a small number of measurable business objectives. Depending on the initiative, those might include reducing processing time, eliminating manual work, increasing adoption, reducing errors or rework, increasing operational throughput, reducing support requests, or increasing revenue.

The key is making those objectives specific. "Improve operational efficiency" establishes a direction. "Reduce average processing time from 30 minutes to 10" establishes a measurable definition of success. Those objectives should then influence what gets built and how the work is prioritized. If the team can't explain how a major feature contributes to one of the outcomes the business is trying to achieve, it's worth asking why that feature is being prioritized in the first place.

Keep Measuring

Those objectives shouldn't disappear once development begins. As teams deliver working software, they have opportunities to test the assumptions that started the project. Did the new workflow actually reduce processing time? Are people using the new capability? Did solving one problem reveal another bottleneck? The answers should influence what happens next.

That's one of the advantages of incremental delivery. It allows a team to build, measure, learn, and adjust while there's still time to act on what it's learning. This doesn't mean every sprint needs to produce a dramatic improvement in a KPI, and it doesn't make traditional project metrics irrelevant. Schedule, budget, quality, and predictability still matter. They simply answer a different question. Delivery metrics tell you how effectively the team delivered software. Business metrics tell you whether the software delivered value.

Strong technology organizations need visibility into both. Because ultimately, "we shipped it" shouldn't be the end of the conversation. It should lead to a much more important question: Did it do what we built it to do?

CONTACT US

