12 bad habits that slow IT to a crawl

Bottlenecks always seem born of the best intentions, but they must be rooted out — and broken. From governance to manual provisioning, here's what's holding up your IT organization's ability to deliver results.

Author: Bob Lewis

It's time to face a cold, hard fact: Your IT department is too slow. It's the result of good intentions gone bad, but in business, intentions don't matter.

When is IT too slow? Whenever any part of the business has to wait for IT to deliver the goods, that's when. The magic buzz phrase these days may be "time to value," but the true guiding principle is "ahead of your competitors." If IT keeps that from happening, you can bet your organization's business executives have lost patience with you.

Want to speed up your IT department? Start by getting rid of what slows it down -- in a word, its bottlenecks. Here are a dozen places to start your search. Ignore at your peril.

IT bottleneck No. 1: Governance

Committees are the old governance. As governance sets the pace for everything IT does, and committees slow down everything they touch, making committees the centerpiece of governance dooms IT to a snail's pace in spite of anything else you try.

Start with size. Every additional committee member slows down decisions; more than five members and progress slows to a crawl as the possibility of consensus disappears entirely.

It's worse if committee members consider themselves not IT leaders but representatives of a constituency that otherwise might not get its fair share. This sort of committee will argue forever instead of solving shared problems.

Then there's the meeting schedule. It's the metronome that sets the pace for every project the committee governs. If a committee meets monthly, then anything waiting for a decision waits a month. How many projects do you have under way that could thrive with a bottleneck like that?

How to solve this? Make culture the new governance. Think of it as the lane markers on the road, relegating steering committees to the role of guardrails — when all else fails they're there to keep IT from plummeting into the chasm, but only when all else fails.

Let culture do the heavy lifting instead. If everyone understands what's most important and knows to focus on it, most governance becomes superfluous.

IT bottleneck No. 2: Multitasking

There's a simple rule for employee multitasking: Don't do it!

This is especially true for software development projects, where studies suggest every interruption costs 15 minutes of lost developer productivity. But what's true for app dev is equally true for every other situation where IT staff have to concentrate.

The challenge isn't knowing that you have to avoid multitasking; it's how to avoid it. The answer lies in the enterprise program management office. It should have one inviolable rule: All projects have to be fully staffed or they can't launch.

What's a fully staffed project? One that never waits for a team member to become available. Limit the number of concurrent projects to the number that can be fully staffed and team members won't have to multitask.

The outcome: Not only will each project complete faster, but the whole project portfolio will complete faster.

IT bottleneck No. 3: Projects

Traditionally, organizations make new technology happen through projects. The more complex the design, the bigger the project. But as projects get bigger the risk of failure increases geometrically while the likelihood of substantial delay approaches certainty.

Consider organizing your efforts into releases instead. Releases bundle discrete enhancements for regression, stress testing, and deployment. Relying on releases instead of projects lets you take advantage of one of the most reliable heuristics of IT management: Enhancements succeed, projects fail.

By the way, if you go this route you don't have to describe it as anything radical. You can call it scrum and enjoy all the company.

Don't stop there. Organizing development work into releases still creates delay. Typical scrum sprints are a month in duration, which establishes a monthly pace for business change, not including having to wait for a Change Advisory Board (CAB) meeting (another governance committee).

So go all the way to continuous integration and deployment -- in a word, devops. Automate testing, subject each software change to continuous integration, and put each small change into production immediately. With changes this small, the CAB is superfluous.

We are, after all, long past the time when a misplaced comma will cause something to explode.

IT bottleneck No. 4: Manual provisioning

Development teams can provision a complete environment in the cloud in a matter of minutes -- or they can ask IT operations for the same environment and get it in a matter of months.

This isn't an intrinsic limitation. It's a choice. As minutes are shorter than months, and as public cloud providers have already perfected the technology, the better choice is clear.

Start automating, and empower your developers to handle provisioning for themselves.

IT bottleneck No. 5: Favoring interfaces over integration

New functionality creates new value. But in most IT shops, new functionality takes a backseat to making sure a software change doesn't break the company's spider web (or hairball) of custom-programmed point-to-point batch interfaces.

Clean up the interface tangle with a well-engineered integration system, and project teams speed up, testing takes less time, and deployments go more smoothly.

Then take it a step further: Turn "Information Technology" into "Integration Systems," whose job is to deliver reliable access to the company's core applications portfolio through standard APIs that expose data and functionality as secure, well-defined services.

IT bottleneck No. 6: Suppressing shadow IT

Shadow IT — business-department-deployed systems — causes problems. In particular, with shadow IT, islands of automation built on shabby foundations are the rule, not the exception.

But shadow IT unfailingly does what the business needs. And it doesn't have to wait for the IT governance process to give a green light, so it delivers what the business needs now.

Think of it as a free source of outsourced application teams that have brilliant business analysts but terrible architects.

Shine a light on shadow IT. Give it a bit of support instead of trying to stamp it out. In exchange you'll multiply your bandwidth, even after counting the effort required to refactor its designs with metaphorical plumbing and wiring that meets IT's metaphorical building codes.

If you've turned IT into Integration Systems you can make the APIs available to the company's shadow IT teams, too, solving the islands of automation problem at the same time.

IT bottleneck No. 7: Insisting on 100 percent solutions

IT's instinct is to bulletproof everything. But coding for a case that hits the system once every thousand transactions takes as long as coding for a case that occurs hundreds of times a day.

Take a lesson from the ancient days of IT: Program the main cases and kick out the rest as exceptions for manual processing.

Computers are good at main cases. Humans are good at exceptions.

IT bottleneck No. 8: Creating data warehouses

Pick one word to describe the typical data warehouse project and it would likely be "behind schedule."

OK, that's two words. Sue me. But data warehouses are still chronically late due to the difficulty of designing OLAP data structures optimized to answer questions nobody knows they're going to want to ask right now.

Enter NoSQL. What makes NoSQL interesting isn't only its ability to handle large data volumes. Even more valuable is its ability to accept data now and let analysts figure out its organization later, when the time comes to query it.

It's this "schema on demand" quality that lets Hadoop implement at hyperspeed in comparison.

IT bottleneck No. 9: Emphasizing TCO

Do your decision-makers forget that costs generate benefits?

Look, any schmuck can cut costs. The trick is cutting costs without impairing delivery. This is where total cost of ownership (TCO) comes into play, and not in a good way.

TCO doesn't care about functionality, except to discourage it. After all, the easiest way to cut costs is by providing and supporting stuff that's less functional and capable. Less use = lower cost.

This is what will inevitably happen with an emphasis on TCO because of one of the inviolable rules of metrics: Anything you don't measure you don't get. This includes the value of what IT does.

Besides, if everyone's focus is on cutting costs, nobody is focused on speeding things up. If speed isn't a priority — let alone *the* priority, as it should be — speed won't happen.

IT bottleneck No. 10: Forcing innovators into the 'high fidelity' IT architecture

Businesses used to insist on being the same tomorrow as they were yesterday. They demanded bulletproof, "high fidelity" systems that never lost data and always delivered the right answer.

Now, innovation matters as much as bulletproofing. It’s the future; it’s where competitive advantage happens.

Don’t force innovators — for example, shadow IT — into the “high fidelity” IT architecture. Give innovators a walled-off space they can call their own while they figure things out and make the future happen. There will be plenty of time to make their innovations hi-fi if and when they succeed.

IT bottleneck No. 11: Allowing a culture of complacency

Even well-run IT shops can get stale, especially in companies that “hold people accountable” when someone tries something innovative and it doesn’t pan out, instead of congratulating them for taking the risk.

A culture of complacency slows down operations because nobody sees any need to speed it up. After all, that’s how we’ve always done things around here.

If that’s the culture you have, go out of your way to shake things up. The alternative is death by boredom.

IT bottleneck No. 12: Establishing an arm’s-length business/IT relationship

Which do you think will deliver results faster: Arm’s-length formality, where IT “negotiates” service-level agreements while requiring its business “customers” to “sign off on requirements and specifications,” or informal conversations that start, “What are you trying to accomplish and how can we help?”

Many pundits call the formal approach “best practice.” Ignore them. Adopt better-than-best practice. Foster strong informal relationships, and never negotiate.

Why? Because negotiation is for people sitting on opposite sides of the table.

In theory, IT and the rest of the business are on the same side — aren’t they?

Source: https://www.cio.com/article/3219845/it-strategy/12-bad-habits-that-slow-it-to-a-crawl.html