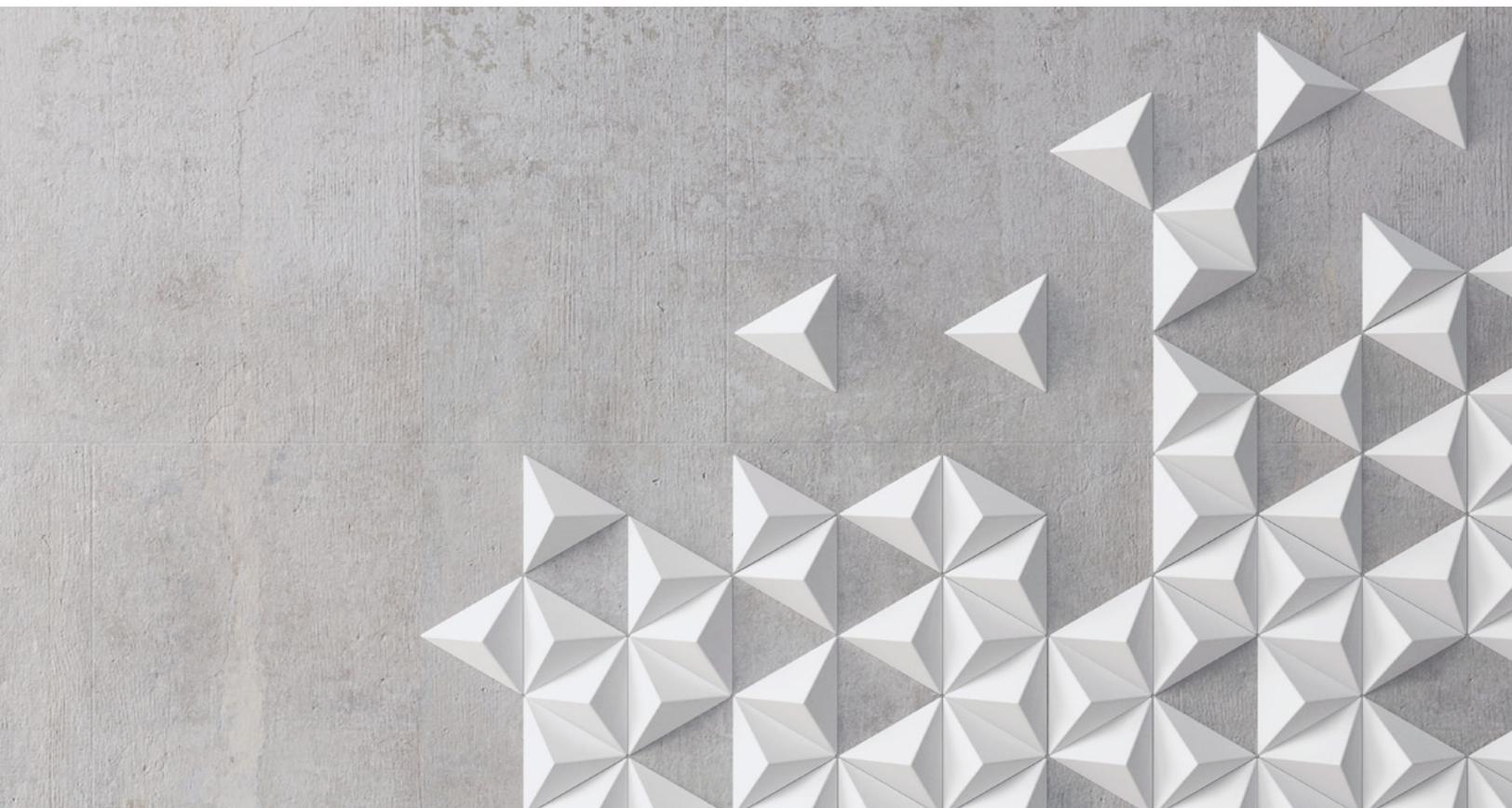


Ten ‘antipatterns’ that are derailing technology transformations

Shortsighted solutions to recurring problems—antipatterns—often sabotage a company’s transformation.

By Sven Blumberg, Thomas Delaet, and Kartikeya Swami



Most major organizations today have embarked on transformation programs in response to changes in customer, competitive, and regulatory landscapes. Whether the transformations are labeled agile, digital, or DevOps, their fundamental premise is to build value by establishing short, iterative, and continuous feedback loops between product and customers that dramatically improve both the product and its time to market.

Technology has a crucial role in enabling this faster and more flexible approach. In our experience, however, technology does not get sufficient attention on the executive agenda. This is a serious flaw given the importance of technology in driving successful digital transformations.

There can be many reasons for this, but one of the main culprits is that technology is often viewed as a “specialist thing,” and IT leaders often have a hard time communicating about technology in a way that engages non-technologists. This reality often leads to “antipatterns”: an ineffective solution for a problem. Antipatterns have serious and sometimes fatal ramifications for technology transformations.

In this article, we synthesize ten of the most frequent technology antipatterns that we have observed in transformations across more than 50 major organizations. How many do you recognize in your organization?

Technology choices

1. Force-fitting technology solutions: Are you choosing technology out of context? Watch out when technology decisions do not attract business scrutiny beyond cost and a cursory discussion of “scalability/strategic alignment.” For instance, we hear in many organizations about a “microservice-first” approach. While microservices are a critical component of many IT modernization journeys, they don’t fit the bill in all circumstances.

At one major corporation, the architects of the transformation suggested an approach that built microservices for a *client-side* application. But microservices is fundamentally a *server-side*

architecture. The architects were simply responding to an organizational push to become technologically modern, but since installation and management of a client-side application was much more cumbersome due to the multiple independent components that would need to work together seamlessly, the approach would have resulted in significant cost and complexity with no additional benefit.

Recommendation

Leaders need to raise their hands to ask “silly” questions to fully understand the rationale and purported benefit of the recommended technology choices. In the preceding example, an executive’s simple series of questions, starting with, “Explain microservices to me as if I were an eight-year old,” could have saved the company millions of dollars!

2. Adopting cutting-edge tech that’s not fully mature: Are you adopting new technology that seems promising but doesn’t have a proven track record?

With stability and scalability as two core elements of any IT organization’s focus, very careful due diligence and decision making are needed to avoid adopting technologies that haven’t fully matured. Leaders should be cautious about any technology recommendation that is pitched primarily because it is in vogue or promises to attract new talent.

A leading bank launched a major redesign for its customer-facing application, using the latest web front-end framework as the software solution stack. It was touted as “future-proof” technology that would attract new talent. The project suffered serious setbacks and cost overruns because staff didn’t have the right capabilities to support it, resulting in time and money to upskill them. It was finally delivered after two years—18 months behind schedule.

Another major bank decided to rewrite its core accounting systems, which were more than 20 years old. While the systems were clean and extensible, the bank wanted to use the latest data-store technology. The project was shelved after an investment of more than €100 million

because the new technology was not stable and the architecture created to support it had several fatal flaws.

Recommendation

Choose simple, proven technologies with which your people are familiar.

3. Building out your own cloud infrastructure without sufficient capabilities: *Have you let security and regulation block your adoption of public cloud?*

Companies are looking to take advantage of new infrastructure platforms and technologies such as container orchestrators, serverless platforms, and analytics solutions. These are complex pieces of technology that major cloud providers are constantly evolving, both in terms of capabilities and reduction in price point, to win the market.

If providing cloud-based infrastructure is not your core business, it will be impossible for you to match the cloud providers on the talent needed to build and run these platforms in a scalable, efficient, and secure way. Moreover, your (digital) competitors will be using these providers to enable them to operate at a completely different price point. At one major financial-services company, we identified more than four different private-cloud initiatives—converged infrastructure, OpenShift, Mesos/Marathon, and OpenStack—each struggling to achieve scale and competing for talent. After many months, the company halted those programs and rightly focused on public-cloud solutions. This is just one of more than 50 examples of private-cloud initiatives that have failed to deliver, often after the investment of millions of euros.

Recommendation

Focus your IT-for-IT investments in public cloud. Start by adopting only *one* of the major public-cloud providers for running your application workloads. Do not pursue a multicloud approach at first, as all major platforms differ significantly in setup and usage and require inordinate effort and investment for a similarly customized setup. For IT tooling, leverage SaaS solutions, such as workflow systems, source-code management, continuous

integration, and collaboration platforms, as much as possible. For the workloads that you need to keep on the premises, use infrastructure patterns that you know and can operate safely and securely at scale.

Technology road map

4. Initiating big-system-replacement programs: *Are you focusing on system replacement rather than improving existing systems in a way that is faster and more cost-effective?*

System-replacement projects are fundamentally complex, cost intensive, and inherently risky. They also distract the organization from building customer-centric capabilities and features in the short-to-medium term. Consequently, big-system-replacement projects should be avoided unless all other paths have proven not to be viable.

A major bank considered a core-banking-system replacement primarily because it was running an old Unix-based system on legacy hardware. But very quickly into the replacement project, the bank realized that its existing system was readily portable to new platforms, including public cloud. Additionally, updating the original monolithic code would entail substantially less cost, effort, and risk than replacing the entire system. Hence, after initial exploration, the bank chose to achieve the target business outcome by gradually modernizing the existing system at a fraction of the estimated big-system-replacement cost.

Recommendation

Before embarking on big-system replacement, ask the following:

- Can you incrementally improve the old system instead of replacing it?
- If you need to build a new system, will it deliver incremental value to your customers as it scales up over time?
- Can you gradually phase out the old system?

5. Focusing on architecture and tooling improvements without enhancing process and delivery discipline: *Did you re-architect and implement new tooling but forget to adapt the delivery processes?* One of the biggest sources of impact in technology transformations comes from simplifying the path to production, the steps involved from defining requirements to releasing software and using it with disciplined repetition across teams. This requires a lot of organizational and executive patience, as the impacted teams—app development, operations, security, support—can take weeks and months to perfect this coordinated dance. Tools and architecture changes can help, but to be effective, they need to be paired with changes to engineering practices, processes, and behaviors. Launching programs for large architecture and tooling changes often requires minimal effort, catches the executive and board's fancy, and represents that things are moving. However, in our experience, without changes to engineering practices, processes, and behaviors, such programs have minimal or no impact.

A major bank realized, after several years of significant investment in development, release, and collaboration tools, that it had no improvement in time to market and a low adoption rate. After months of futile top-down incentives and nudges for tools adoption, the bank refocused on how the tools enabled a new set of engineering practices and collaboration between teams. It showed how the new tools could simplify the path to production. At last check, more than 40 percent of the teams were onboarded on the new way, and there was dramatic improvement in both time to market and the tactical adoption of tools. In a similar example, a major European bank dramatically increased delivery speed by focusing on a common and clear understanding of all components of the delivery process, establishing a strict cadence on how it would be executed, and simplifying some of the documents and approvals required before releasing software to production.

Recommendation

To improve speed of delivery, start with baselining the path to production to identify strengths and

gaps. Follow this with simplification of process and delivery artifacts and addressing relevant gaps through tools and architectural changes, as required. Once the new process has been instituted, ensure that the teams adopt the cadence of disciplined repetition of this process.

Technology management

6. Focusing on outputs rather than business outcomes: *Are your technologists focused on output instead of business/technology outcome?* Are your technology targets too . . . tech-y? Does the technology organization clearly articulate and track customer-focused targets? If the answer to any of these questions is even a mild pause to reflect, you need to dig deeper.

Well-meaning technologists, even in customer-centric organizations, often default to focusing on tech output. It is easily measurable, all-consuming, and very “in control.” Examples include amount of screens delivered, functionalities deployed, and defects tackled. The tech-output metrics might be great, but unless you're measuring the direct impact of technology on customers, they are not particularly relevant. At one of the major financial organizations, the app-development group focused on 100 percent test automation as a key result and celebrated success—and closure—when they achieved it. However, testing took as many days as it had before, and there was no improvement in the product's time to customer.

Recommendation

- Business and technology leaders should define joint accountability for desired outcomes (aka “two in a box”):
 - » **Business outcomes:** usage of your products (number of customers, daily usage) and customer satisfaction (net promoter scores, number of support requests)
 - » **Technology outcomes:** functional availability/security of your product

and efficiency of development (release frequency, toil)

new features, automation, or infrastructure optimization.

- Have the technologists and business articulate specific and measurable business and technical *outcomes* instead of technology *outputs*. The magic of product ownership and cross-functional teams lies in understanding the trade-offs between these different business and technology outcomes and making conscious choices about what to prioritize to balance short-term objectives with long-term product health.

7. Managing IT purely for cost: Are you sacrificing significant value by overindexing on price and cost? Managing IT purely as a cost center is an outdated mindset that can have drastic repercussions, ranging from the inability to attract the right talent to discouraging use of critical and expensive technologies and platforms. For example, hiring or sourcing primarily on price typically results in subpar talent.

One financial-services company paid its vendor cheap day rates, and the vendor “recouped” the discounts by staffing novices and inflating estimates on what it would take to deliver certain features. The result was very low-quality code and long cycle times.

Recommendation

Instead of managing internal IT as a cost center, consider the following to think through how to set and align efficiency incentives:

- Bring competence into the mix when talking about the cost of talent. Some companies have defined a unified competence model ranging from “novice” to “expert” for both internal engineers and vendors. Vendor pricing is discussed in terms of “novice equivalents,” recognizing that productivity differences between novice and expert engineers are a factor of eight to ten.
- Create a true total-cost-of-ownership (TCO) view of your products to make sensible trade-off decisions on whether to invest in

8. Investing in developing new platforms without involving the business: Is your primary focus platform development instead of platform adoption by the business? IT organizations rightly invest significant effort and resources in building and deploying robust platforms. However, quite often business is not involved in platform design or development, leading to new platforms with minimal relevance for the business side and, hence, poor adoption.

A major US bank made a multimillion-dollar investment in a data lake, ostensibly to pivot to a data-first culture. As a data-organization project within IT, it was conceived, designed, and developed primarily without business engagement. The data lake was delivered slightly behind schedule. More than a year later, the bank was still trying to make progress capturing use cases the data lake supported. In addition to the unused platform, the data organization suffered significant staff turnover due to poor morale. The bank has now launched multiple programs to update the data lake to meet business needs.

Recommendation

Start and end the conversation on all technology platforms with the business problem that they will address. Focus on that relentlessly, and ensure that all the right stakeholders across business and technology have joint accountability for the platform’s delivery of customer value. When building your platform, focus on building use cases, and instead of spending time up front on putting enablers in place, accept that you might want to refactor pieces of the platform when you onboard more use cases.

Managing technologists

9. Outsourcing your core value streams: Are vendors doing the work that creates the most value for your business? If your core technology knowledge or intellectual property (IP) is outsourced (either through offshoring agreements

or vendor-support contracts), you risk limiting the impact of any transformation and depending to an unhealthy degree on a third party. Because outsourcing has proven to be an effective tool to reduce cost for commoditized activities, some leaders have expanded its scope in response to cost pressures and outsourced entire subgroups or critical platforms. Such dependence has severely restricted organizations in making bold strategy and partner choices, mostly because they have little or no control over their own IP.

For instance, at a major insurer, 90 percent of the technology organization was outsourced across different vendors. During its transformation, it realized that a major reason for underperformance was that the knowledge of its three core systems was held by three individuals at different vendors. Not only that, but they had never spoken with each other.

Recommendation

Clearly demarcate the boundaries for outsourcing at business-critical technologies or activities. If you are currently heavily outsourced for critical activities, align with your peer group and the board on a plan to progressively bring them in-house.

10. Building up an army of managers rather than developing an engineering culture: Do you value your managers more than your engineers? Career growth in most organizations usually entails people management. Gradually, talented employees who once showed great technology promise spend more time on managing people and administrative activities than on practicing the craft of engineering. They become full-time managers. Over time, they lose the ability to engage in deep technical

conversations with their teams, to role-model technical problem solving and innovation, and—most damaging—to effectively manage team performance based on detailed technical merit. Consequently, organizations have a large IT group that consistently underperforms and has little technical guidance or accountability.

One major financial organization launched a program to overhaul its performance-management process and discovered that, on average, managers spent less than an hour a week in technical discussions with their teams. Does that sound familiar?

Recommendation

Give technology managers specific responsibilities for tech delivery. Encourage a culture of tech expertise for technology managers through monetary and nonmonetary incentives. To build an engineering culture, define granular performance-review criteria for technologists that are focused on both delivery and expertise.

Identifying and addressing these challenges require a concerted effort, with focus and ownership from both business and technology leadership. As more and more organizations launch and mature their digital transformations, executives must constantly probe for any evidence of the above antipatterns and urgently move to address them. Only then can the technology transformations evolve sufficiently to support the most vaunted three-part outcome of the digital transformation: faster customer-centric delivery, business growth, and happier employees.

Sven Blumberg is a senior partner in McKinsey's Istanbul office, **Thomas Delaet** is a partner in McKinsey's Brussels office, and **Kartikeya Swami** is an alumnus of the New York office.

Copyright © 2020 McKinsey & Company. All rights reserved.