It’s become one of the more persistent and unshakeable complaints from businesses large and small: tech projects routinely suffer long development delays and expensive cost overruns, making them appear in too many cases to be more troublesome than they are worth.

Having heard these complaints too frequently to ignore, we set out to determine if the anecdotal claims are true—and, if they are, to identify why these problems keep arising and what the most effective solutions might be.
What we learned was surprising, both in terms of the breadth of project shortfalls and the way they must be addressed. We discovered that popular agile software development principles—involving, for instance, cross-disciplinary teaming and progress tracking—are pertinent but not sufficient to guarantee success. We also learned that improving the outcomes of IT projects at most companies will depend squarely on technology leadership creating the proper requirements and culture for teams to be able to deliver in agile fashion. Up to now, this has proved to be difficult to achieve.

**IT Frustrations Are Palpable**

To begin tackling this issue, BCG conducted a survey of global business and technical-side C-suite executives across 25 industries, including technology, consumer goods, retail, health care, and manufacturing, among many others.

We deliberately focused our survey on internal software development projects, primarily custom programs and applications targeting a specific business priority, such as marketing automation and streamlining tools, forecasting suites, and pricing models. And we avoided large-scale enterprise resource planning implementations or cloud migrations and other big projects potentially impacted by the contributions of external contractors and system integrators; such projects are so wide ranging and complex that it would be difficult to determine the precise locus of their shortcomings.

The survey results were unambiguous: disappointment with IT is a real thing. Nearly half of all respondents said that more than 30% of the technology development projects in their organizations suffer from delays or budget overruns, while nearly one in five respondents indicated that less than satisfactory outcomes occur more than 50% of the time. (See Exhibit 1.)

Moreover, despite the hype in recent years that agile software is an antidote to failing IT projects, our survey found that there was no correlation between the methodology used to design and deliver programs and their success. Obviously, there are exceptions to this—agile principles can underpin excellent outcomes. But not consistently. In fact, 64% of respondents said their IT teams already used some version of agile software.
development tools as the basis for their primary program design efforts. And still the troubling trend lines for IT projects have continued.

The executives who took BCG’s survey also had definite ideas about why the scheduling and budget overruns were so commonplace. Three causes stood out as the worst culprits: lack of alignment between the technology and business sides of the organization regarding the specific operational objectives for the program; unrealistic timelines; and insufficient financial and developer resources earmarked for the project. (See Exhibit 2.)

Exhibit 2 - Tech Projects Are Most Often Delayed by Unclear Business Outcomes, Unrealistic Timelines, or Resource Shortages

Taken together, these three explanations for why IT projects languish point to organizational leadership problems on both the technology and business sides of a company. CIOs, CTOs, and the heads of IT departments (or whatever their title is at a given company) are all too often either left out of the overall decision-making process when a technology implementation is initiated, or they fail to plainly set expectations for what an investment in a new application will specifically deliver both in terms of a timeline and performance. Ultimately, overcoming this leadership challenge is the biggest driver of success.

It is worth noting, however, that technology leaders at companies have significant challenges to navigate to make the types of changes required to get IT projects back on track. CIOs and CTOs are compelled to oversee work on disciplines much broader than their skillsets as organizational managers, for instance; engineering decisions are complex and usually beyond a technology leader’s knowledge set.

At the same time, business leaders do not fully understand a CIO’s portfolio, especially its technical nuances. As a result, CIOs are blamed and penalized for any technical snag and defect, but they are not generally rewarded when they deliver to plan or beyond expectations. Despite these issues, technology leaders have no choice but to adopt new approaches to improve IT outcomes—because the current situation is not sustainable.
Overcoming the Disappointments

Of course, identifying that a leadership problem exists and alleviating it are two very different things. The former is a 30,000-foot view and just reflects that something is generically wrong; the latter must occur at a more granular level, where the problems are addressed. To take that more granular approach, it is essential to explore the common elements that are present when an IT project succeeds—and to consider what new approaches are necessary to adopt these more salient practices.

Based on survey responses, we determined that generating better IT outcomes depends on the following factors.

A Strategic Partnership. This is the proverbial seat at the table. When technology leaders were directly involved from the start in the strategy development for a tech build, the success rate was 154% higher than projects without this communications channel. Technology leaders speak a very different language than business department managers, who can’t evaluate technical paradigms and archetypes or AI algorithms and machine learning dictionaries. But these managers should be able to adequately describe which performance improvements they are seeking from a new application or program and what constitutes a positive return on investment.

In turn, CIOs must take the business wish list and explain what is possible—and what resources are required to achieve the possible—in clear language that demystifies the technology. In other words, the strategic dialogue between the users and the builders should be the first step, rather than engaging IT in a second step when communicating in the same language becomes much more difficult, if not impossible.

Incentives. At most companies, technology developers and designers are stuck in a negative feedback loop in which minimum results are acceptable outcomes, while delivering real value brings little notice and certainly minimal professional gain. Mere technical proficiency—without real measurable gains in, for instance, delivery dates or costs—is considered the standard that meets expectations. Mistakes, errors, bugs, and delays that could come from taking risks to improve the design and development process in building an application are penalized. With upside ceiling and downside risk, maintaining the status quo is a more sustainable option for developers.

A better approach is to incentivize incremental gains that can ultimately provide significant returns to the organization. For example, measure the success of an IT project in speed to outcome; the desired delivery could be that a pricing program first covers a relatively small subset of categories, then broadens to a wider group in the next iteration and then adds markdown decisions in yet another version, and so on as it evolves.

The power of incentives is supported strongly by the BCG survey. Three critical software development improvement metrics stood out as having the potential to motivate the best results:

- Delivering or accelerating business improvements
- Meeting or accelerating deadlines and timelines
When meaningful rewards were offered for improvements in all these dimensions, the success rate for IT efforts were nearly 25% higher than for projects without such incentives.

**Performance Tracking.** Simply put, technology teams are frequently not focused on business outcomes. Engaged—and, in our view, distracted—by concentrating keenly on how a program is built, developers view success in terms of achieving steps or technology benchmarks, not in company results. The problem lies in not establishing benchmarks in the form of KPIs, which typically gauge business in numerical scores that can be tangibly and objectively measured and analyzed.

For instance, commonly program developers will set as their goal to build a data layer, or install a CRM program, or design an API layer. But for successful technology projects, these goals should be reframed as creating a common data layer around transactions for the past five years that can serve as the foundation of business intelligence and analytics efforts, directly informing strategic aims.

Similarly, deploying a CRM is a less useful target than implementing a new marketing campaign in half the time it took before. And designing an API layer could be better translated as creating a tool for refreshing personalized product images on an essential app without latency.

Too often, developers view success in terms of achieving steps or technology benchmarks, not in company results.

Once these goals are carefully chosen and cloaked in the language of business outcomes, it's critical to track them properly. As with any KPIs, if they are not measured in a disciplined and carefully calibrated way, they won't serve a useful purpose.

In our survey, we found that if reasonably valid tracking mechanisms are in place for an IT project, success rates improve. Moreover, while achieving business outcomes remains an essential goal for a technology implementation, chances for satisfactory outcomes are improved when scheduling and budgeting progress are tracked and, if needed, set back on course as well. Specifically, we found that when effective early warning systems are in place—those that can alert management of problems before they become too cumbersome—success rates increase by up to 16 percentage points.

Even when organizations have tracking mechanisms in place, however, the real financial impact of overruns is often understated. In relative terms, if the gold standard is that a tech program costs $10 to deliver $100 in twelve months, then a six-month delay typically means more cost and postponed value realization—and the projected 10x return is quickly reduced to 4x or 5x in actualized value.
Talent. Having the appropriate skills and capabilities in a technology organization to drive business performance improvements is one of the more difficult challenges today. Variability and hidden surprises in technical implementations are the norm, not the exception. The layered, iterative, and scattered nature of IT solutions means that pivots and quirky dependencies during any project are a feature, not a bug. Overcompensating for unpredictability in an IT team can produce diminishing returns quickly.

Consequently, the most impactful members in an IT organization are those with prior experience—the more experience, the better.

For CIOs and other technology leaders, there are serious obstacles to managing the talent pools required to drive successful IT implementations. In most cases, technology managers are removed by many years from actual design, development, and programming. Consequently, on-the-ground, day-to-day direction of a new project using the latest implementation tools and platforms is not necessarily a CIOs sweet spot. Instead, they have to rely heavily on their staffs to be creative and innovative, using their most recent experiences as their guides.

In other words, leadership in this instance is more about strategic management and keeping the project focused on the business outcomes than actual coding and cadences. Execution excellence is in the hands of developers.

The positive delta of well-travelled and well-rounded technical staffers manifested clearly in the survey when we looked closely at the connection between the type of talent companies had and IT project success rates. First of all, broadly, fully staffed teams with cross-disciplinary capabilities had a 76% increase in project performance versus organization that had multiple and key positions vacant.

But on a more granular level, having product managers with wide ranging skills increased success rates by 30%. And the presence of specialized engineers who are experts in large-scale design and machine learning methods—among the most current sets of knowledge needed in business-based advanced IT projects—improved outcomes by 14%. (See Exhibit 3.)
When considering the common reasons for software development success today, it is hard to ignore a very large recent presence in the IT project environment: namely, the emergence of generative AI and the role it will increasingly play in future technology design efforts and in software applications.

We believe that GenAI will only amplify the critical principles for success that we have identified. This is primarily because choosing to adopt a GenAI platform is a step closely linked to a company’s overall strategic agenda. Implementing GenAI will require deep coordination between tech and business counterparts, an unwavering orientation towards measurable business value, and a set of thoughtful incentives to attract and reward specialized talent that can straddle the nuances of AI innovation and its hands-on applications.

**Practical Next Steps**

Each of the factors described above are essential for improving internal software development programs and overcoming the notion that IT projects fail too often. Taken alone, however, these factors are only a band aid, not a cure; taken together, real results are possible. Most importantly, the common elements of success in IT implementations are directly the result of actions that CIOs and tech leaders are in the best position to steward in their organizations.

First, **frame the parameters** of the initial discovery-stage exercises—the phase where the project’s goals, requirements, and scope are thoroughly examined—so that they are reasonably (and rigorously) time-boxed. This may involve consultations with business units that will use the program, along with market research and

---

**Exhibit 3 - Successful Software Projects Require the Right Talent**

| Average tech project success rate, by position and experience level (%) |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| Project manager             | Solution architect          | Data architect              | Data engineer               | Data scientist              | DevOps engineer             |
| 52                          | 62                          | 65                          | 62                          | 63                          | 62                          |
| 68                          | 57                          | 67                          | 61                          | 60                          | 59                          |
| 67                          | 65                          | 67                          | 66                          | 67                          | 68                          |
| 62                          | 61                          | 67                          | 68                          | 66                          | 68                          |
| 68                          | 69                          | 60                          | 68                          | 68                          | 68                          |

Source: BCG survey of 403 global business and technical-side C-suite executives across 25 industries.

1 Basic = able to perform routine tasks and follow instructions but needs more depth and breadth of knowledge and experience; intermediate = able to apply skills and knowledge to various situations and problems but may require guidance or supervision in complex or unfamiliar scenarios.

2 Advanced = able to demonstrate mastery of skills and expertise in various contexts and challenges and can work independently or lead others; expert = able to innovate and create new solutions based on extensive skills and knowledge, and can mentor, coach, or teach others; leader = able to shape and influence the direction and strategy of the field or domain, and can inspire, empower, and mobilize others.
the creation of prototypes. But it should be focused on short-term specific outcomes tied to iterative long-term adjustments, which in turn avoids lengthy exercises and overanalysis.

The fastest path to placing the software in users’ hands is the quickest route to real-world value creation, rather than theoretical problem resolution. Importantly, discovery-stage efficiency is in many ways a function of how much experience the development team has—if they have seen it before, they are more likely to know how to address a challenge in scoping without delay. Indeed, this is where access to the best talent matters most.

Second, **play an active role** in identifying the core business outcome of the IT project—one tied to actual hands-on usage. This should be a goal defined as a capability, not a tool or a technical specification. For instance, the potential outcome may be described as pricing decisions in near-real time for a specific region.

Our empirical finding is that the initial design and production should be limited to what is possible to launch in three to four months of development work—and thus the program should be relatively circumscribed. After it proves out in its first implementation, it can be expanded to more regions and a broader set of SKUs. This way, the business could measure actionable outcomes at a fast clip and provide feedback ahead of further development.

Third, **change the culture** of IT development from a risk-averse mindset to taking chances to improve based on prior knowledge. To do this, measure and reward velocity of outcomes and extensibility of solutions, such as the modularity or the ability to reuse aspects of the program. In turn, that creates an enriching technical challenge to rapidly develop programs in a narrow timeline that are ultimately sufficiently flexible to be augmented and transferred to other parts of the organization.

Driving IT programs that support value creation without delays or budget overruns may seem like a tall task, but it is achievable. Improvement will depend on changing business as usual, and technology leaders are at the forefront of that change.

Technology leaders must enable the prerequisites for agile methodologies—implementing a culture change, hiring the appropriate talent, and creating an incentive structure that allows IT designer and developer teams to focus on salient outcomes, program expandability, and high throughput. By doing this, tech leaders can ensure that agile principles actually—finally—work.
ABOUT BOSTON CONSULTING GROUP

Boston Consulting Group partners with leaders in business and society to tackle their most important challenges and capture their greatest opportunities. BCG was the pioneer in business strategy when it was founded in 1963. Today, we work closely with clients to embrace a transformational approach aimed at benefiting all stakeholders—empowering organizations to grow, build sustainable competitive advantage, and drive positive societal impact.

Our diverse, global teams bring deep industry and functional expertise and a range of perspectives that question the status quo and spark change. BCG delivers solutions through leading-edge management consulting, technology and design, and corporate and digital ventures. We work in a uniquely collaborative model across the firm and throughout all levels of the client organization, fueled by the goal of helping our clients thrive and enabling them to make the world a better place.

© Boston Consulting Group 2024. All rights reserved.

For information or permission to reprint, please contact BCG at permissions@bcg.com. To find the latest BCG content and register to receive e-alerts on this topic or others, please visit bcg.com. Follow Boston Consulting Group on Facebook and X (formerly Twitter).