

McKinsey  
& Company

# A new management science for technology product delivery

February 2020

# A new management science for technology product delivery

Over four decades ago, Turing Award winner Fred Brooks argued in his book “The Mythical Man-Month” that when a technology project is falling behind schedule, adding programmers delays the project further. This simple but counterintuitive observation has frustrated business and technology executives for decades.

Why aren't technology projects as predictable as other capital-investment projects in responding to changes in labor, a key factor of production? And why aren't these projects measurable with early management-science concepts like manufacturing throughput, cycle times, and unit costs? Many technology companies continue to highlight the challenges they face in managing demand and supply across technology-project portfolios, setting business expectations with accurate estimates, and communicating the productivity, quality, and effectiveness of technology-delivery teams in a way that inspires confidence and attracts investment.

Although technology products and the project teams delivering them remain very hard to manage scientifically, the rise of agile methods in the last two decades has helped. Agile practitioners report the significant benefits of agile methods on quality and speed of technology development. The impact on GE, ING, Spotify, and others has inspired entirely new thinking on technology operating models and reinforces the benefits of cross-functional business and technical teams. Yet even agile experts we speak to agree that agile has not yet risen to the level of management science. Instead, agile exists somewhere between management best practices and what some experts describe as a quasi religion.

The upshot: most technology leaders continue to base management decisions on broad rules of thumb and intuition. “Small teams are better than large ones,” “quality of engineering talent trumps quantity,” “colocated teams beat distributed teams,” and “team

autonomy is better than team dependency” are just a few examples of the conventional wisdom ingrained in technology-management circles. Unfortunately, these rules of thumb offer little insight into how managers can affect project outcomes like quality, speed, and utility of technology products. Further, intuition utterly fails when justifying and planning investments in technology capabilities to meet the surging demand for digitization faced by IT organizations.

With 93 percent of companies reporting that digital is critical to achieving their goals, we expect technology leaders will come under increased pressure to plan, measure, and track team performance levers more scientifically. As a result, McKinsey decided to study this topic deeply. We found several forces converging to help bring more “science” to the management side of technology:

- *Legacy IT organizations are quickly pivoting to a product mind-set and operating model.* A simple Google search shows interest in the topic of product management has increased 300 percent in the past decade. Taking their lead from digital natives and Silicon Valley icons, many IT organizations have been pivoting away from recruiting legacy “project management” talent to instead focus on “product management” talent. The latter effort seeks expertise in understanding the end-to-end customer journey and employs product development practices guided by customer testing and data.<sup>1</sup> Further, IT shops are tearing down silos that for years isolated user-experience design, application development,

<sup>1</sup> Chandra Gnanasambandam, Martin Harrysson, Shivam Srivastava, and Yun Wu, “Product managers for the digital world,” May 2017, McKinsey.com.

and infrastructure. They are restructuring into persistent product teams with more autonomy to make full-stack technology decisions faster and in response to changing customer needs.

- *Technology functions now sit atop mountains of untapped data about their own operations.* Data previously occupying separate systems—HR records, project management, code repositories, communication tools, finance, and service ticketing—can now be engineered in ways that reveal insights into the conditions affecting technology-team performance. According to a recently published finding by Microsoft, calendaring data (among other sources of information) signaled Microsoft managers that in the company's devices unit, management practices related to meetings were reducing engineers' job satisfaction. That insight enabled managers to take action.
- *Developer operations maturity is growing, making process analysis feasible across DevOps stages.* The past decade has seen widespread adoption of tools like Jira, Git, CircleCI, and more to support the DevOps practices within large technology-delivery teams. Our analysis of two years of data from more than 300 teams within one company revealed that on average, only 20 percent of the time did engineering and infrastructure teams deliver projects according to their original project estimates. The company now uses this metric to help teams continuously improve their estimation and capacity-management practices.
- *Deep instrumentation of end-user experiences and service operations offers new objective measures of technology quality and adoption.* Beyond DevOps environments, data from systems like ServiceNow, Remedy, and Heap Analytics enable technology leaders to track objective performance signals—release velocity, new-feature adoption, late-stage defects, and outages in production environments—at a granular level. In a recent study covering hundreds of IT projects, fully colocated delivery teams working on almost 5,000 epics delivered 40 percent fewer bugs. However, distributed teams resolved issues faster. Our findings suggest that optimizing team location may help development managers achieve an optimal combination of speed and quality.

Together, these trends point to a future in which delivery teams can use data and analytics to drive continuous improvement, managers can make better-informed resourcing decisions, and executives obtain greater insight into the supply and demand factors affecting technology investments.

## **Toward a new management science for technology delivery**

Our work thus far has focused on two categories of outcome variables: schedule adherence and quality. Schedule adherence—which we have measured as estimate to actual project duration and as quantity and magnitude of project and task overruns—is a critical measure for technology teams to get right, given the cost of overruns in direct financial terms and indirectly in terms of time to market for critical capabilities. Quality, in our work, is measured as a normalized percentage of bugs produced during the development process. We define these as “early-stage” defects because teams typically detect and fix them before they affect the end user's experience.

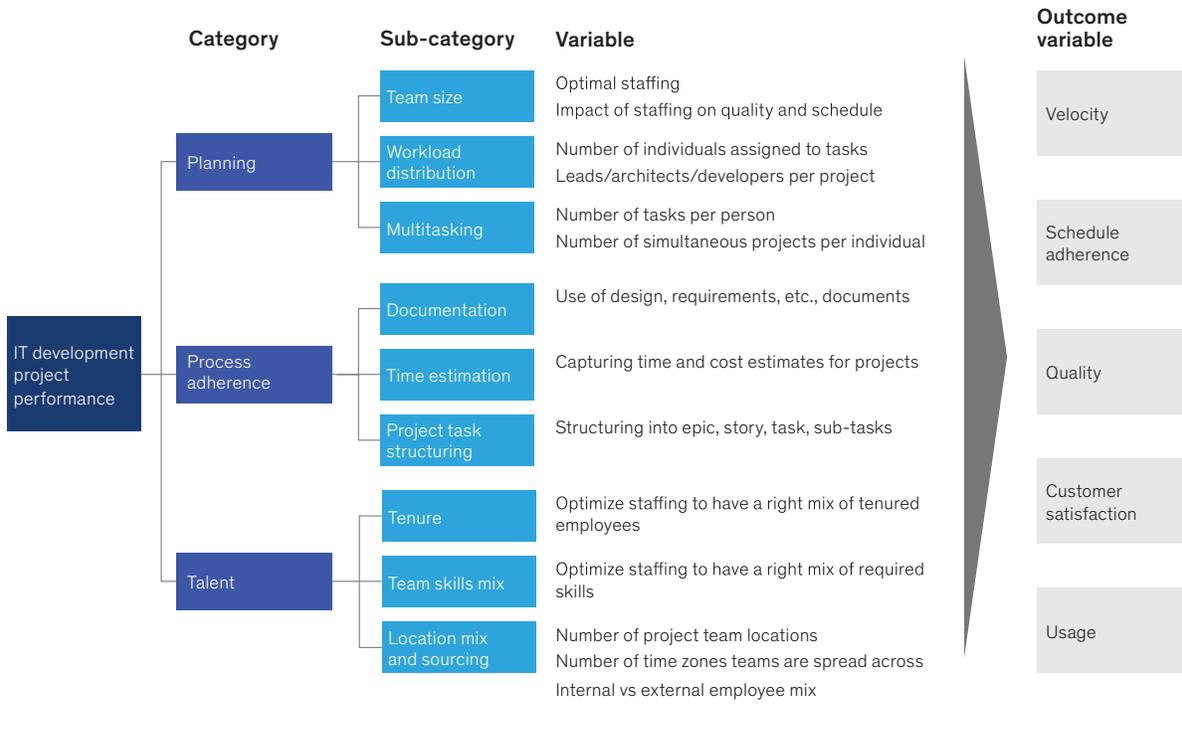
Using a hypothesis-driven approach, we worked with senior leaders and technology-delivery managers to identify favorable outcomes, as well as conditions and practices in operational systems and databases that are associated with favorable outcomes. The conditions and practices—the input factors in our model—fall into three broad categories (Exhibit 1):

1. *Planning category in the exhibit* includes variables managers can influence in the planning stage and/or continuously during sprint planning cycles. Among these are team size, optimal task workload per resource, and degree of multitasking, to name a few.
2. *Process adherence* includes variables measuring the degree to which teams are following common process best practices. These include the presence of artifacts such as architecture documents, design specs, and test scripts, as well as fill rates on task estimation and packaging of work units into epics, stories, and tasks.
3. *Talent indicators* include variables for measuring team structure, composition, and location. Among these variables are average tenure of resources, number of prior projects completed, degree of team colocation, and number of sites hosting project resources.

## **Early findings point to multiple areas for impact**

To get our arms around the right issues, we first studied medium-size technology shops within a financial institution and a professional-services firm. Collectively, the data set included two years of data, representing over 600 technology projects, more than 5,000 epics (subprojects), and over 100,000 stories, tasks, and subtasks. More than 500 full-time employees of these companies and contractor-based technology professionals contributed to these

**Technology-management analytics encompasses measures of planning, process, and talent.**



Note: Not exhaustive.  
Source: McKinsey Technology Management Analytics

projects. The findings of this initial research effort suggested several areas where management can improve project performance.

**Small and medium-size teams are most likely to optimize quality and speed**

Overall on average, quality and schedule adherence was higher with project teams of 10 to 15 people. Team size was most associated with normalized project delivery time, with teams seeing improvements of about 5 percent in elapsed time with each additional team member up to 15 (Exhibit 2). Beyond 15, average elapsed-time improvements continue to increase, but at a significantly slower rate.

The impact of team size on quality measures in our sample was far less pronounced, but quality appears to decrease with increases to normalized team size (Exhibit 3). This pattern may reflect the challenges inherent to deploying proper unit-, system-, and integration-testing discipline across large teams. While we expect early-stage defects as a percentage of overall project issues to differ widely by engineering talent levels, regardless of project size, development managers should strive for a rate of early-stage defects below the average of 10 percent.

From an overall planning perspective, the data support a simple test for rightsizing teams by comparing a

synthetic measure of normalized project size (derived from number of epics, stories, tasks, and subtasks) with normalized team size and then adjusting for project timeline. Exhibit 4 applies this method to the projects we studied. While the two variables understandably have a linear relationship, outliers are easily identified as small projects with many contributors and large projects with few contributors.

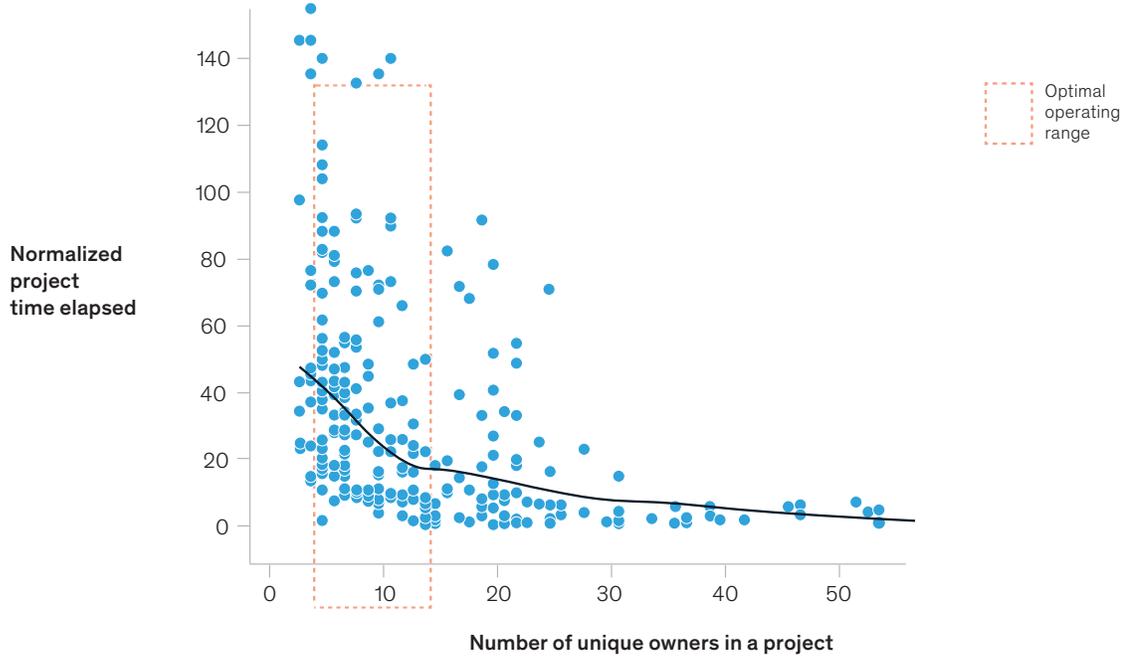
**Software bugs are costlier to fix in later stages**

Based on separate examples from a financial and professional-services firm, we found an optimal range of bugs generated by project teams: around 10 percent. Generating more than the optimal percentage (10 percent in the referred cases) of bugs interferes with project and team planning, which increases the project cost and often leads to schedule overrun. The cost of a bug may not be easily quantifiable, but depending on the phase of the software development life cycle (SDLC), the cost can blow out of proportion. An IBM study found that cost of fixing a bug can multiply 100 times between the design and downstream maintenance phases of technology projects (Exhibit 5). The further teams move into the SDLC, the more complex resolving bugs becomes.

Today we can rely on data science and piles of data that IT teams already have to make planning decisions

Exhibit 2

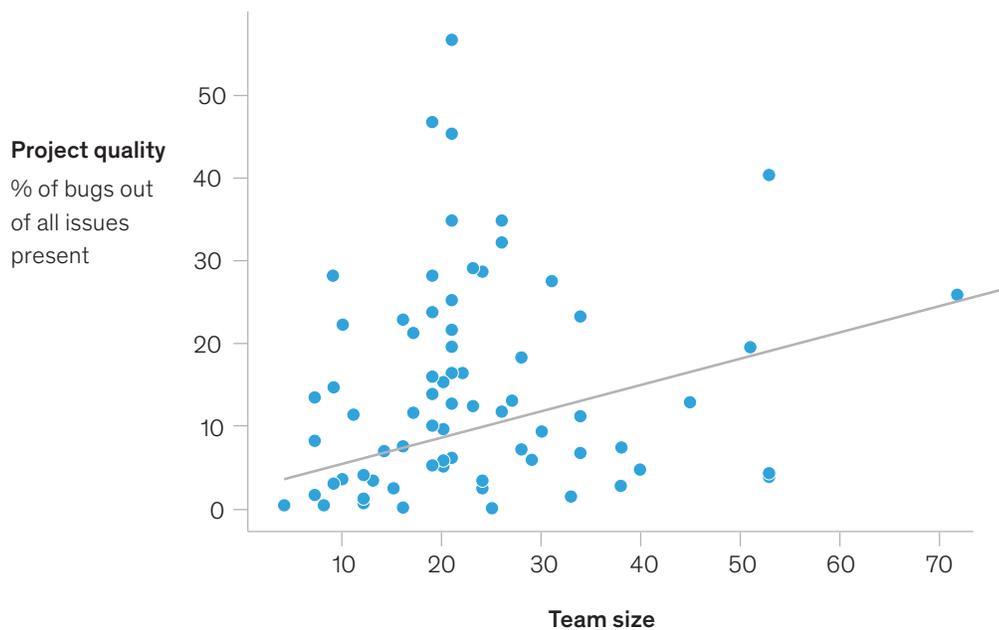
**Adding headcount improves delivery speed, but incremental benefits are reduced beyond an optimal team size.**



Note: 254 = Sample size (number of projects)  
Source: McKinsey Digital 20/20 analysis; McKinsey Technology Management Analytics

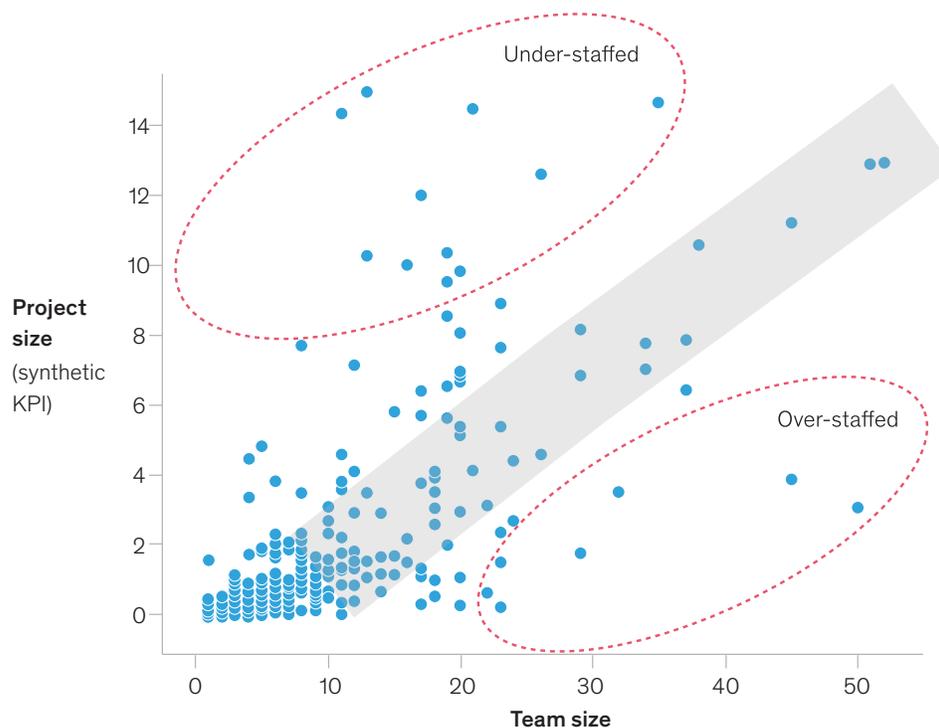
Exhibit 3

**Defect rates rise along with team size.**



Note: 72 = Sample size (number of projects)  
Source: McKinsey Digital 20/20 analysis; McKinsey Technology Management Analytics

## Comparing project size to team size highlights outliers.



Note: 296 = Sample size (number of projects)  
Source: McKinsey Digital 20/20 analysis; McKinsey Technology Management Analytics

that can help minimize such costs. Simply tracking bugs and defects into a tracking tool or tracking system does not help; rather, cost minimization requires a structured problem-solving approach to identify, analyze, and deal with root causes.

### Teams that adhere to processes excel by walking the walk of agile teamwork

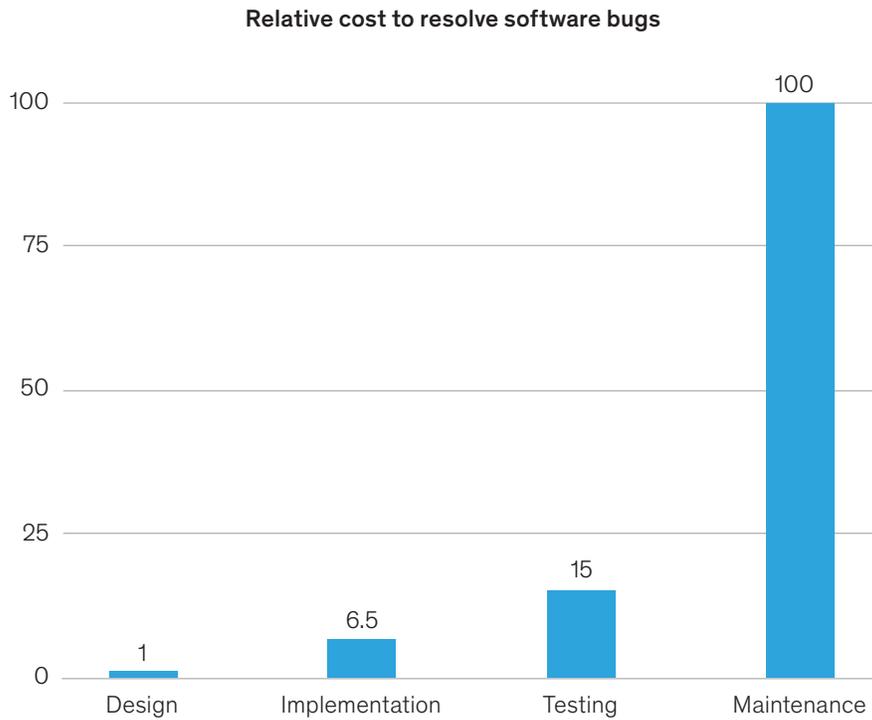
Prior studies by McKinsey using 360-degree diagnostics of agile team capabilities have shown that three capabilities are significantly correlated with predictability of technology projects (Exhibit 6):

1. *Product management.* The team has clear product vision and an understanding of the medium- and long-term expectations of what the product is aiming to deliver (e.g., specific business objectives, results, key performance indicators). The team actively drives the adoption and improvement of the product even after its release.
2. *Agile ceremonies.* The team conducts all agile activities with appropriate team members regularly participating in each, including sprint review, daily scrum, sprint planning, sprint retrospective, and backlog grooming.

3. *Team autonomy.* The team is able to deliver a product end to end with minimal dependencies on other teams. As a result, the team can deliver outcomes without significant handoffs and coordination.

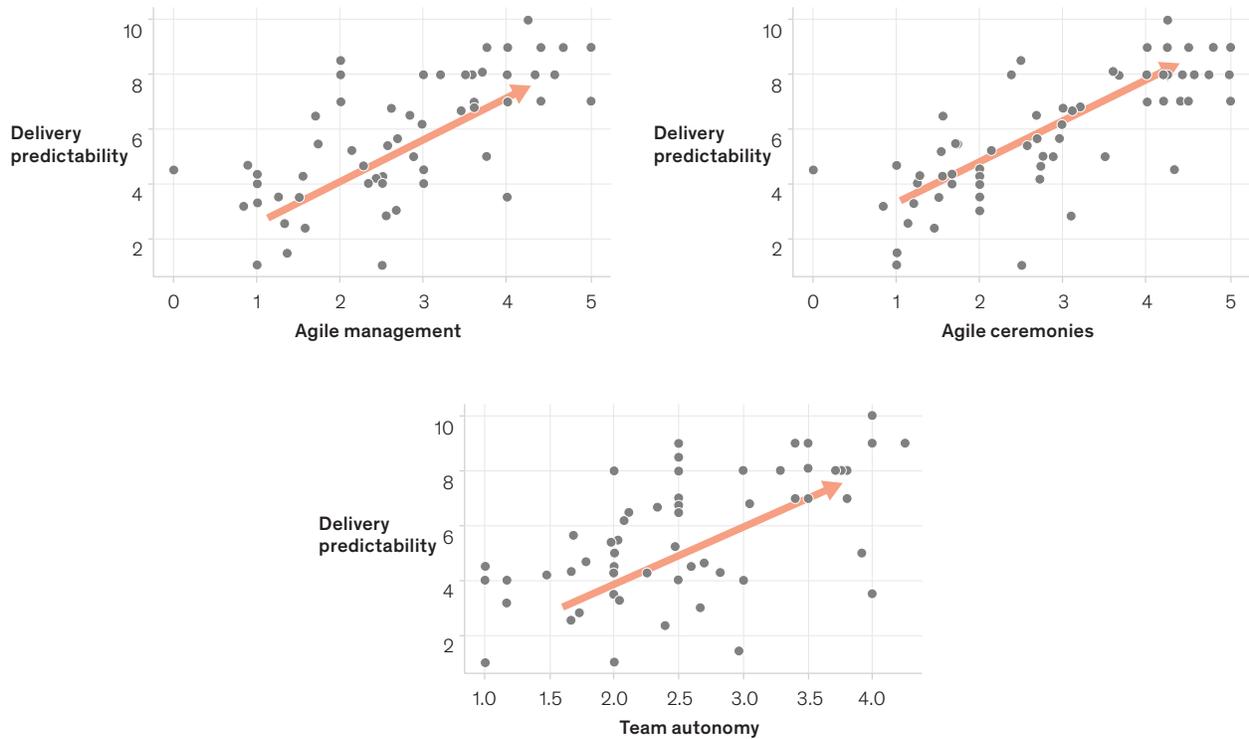
Bringing this knowledge to our most recent deep dives, it was unsurprising to see factors related to process adherence—for example, the percentage of stories and subtasks with estimates and evidence of backlog grooming into manageable work units—distinguishing teams producing higher quality and on schedule. In one organization, as percentage time estimates populated increased from 40 to 100 percent, normalized percentage schedule adherence increased 1.5 times (Exhibit 7). As percentage time estimates populated increased from 30 to 100 percent, normalized bugs as percentage of overall issues declined from 20 percent to 3 to 5 percent. While we don't see an argument that there is a causal relationship between task estimation fill rates and superior project performance, a more plausible argument is that teams indexing high on this factor employ more mature process adherence and product management discipline, which enables continuous learning.

**The relative cost to resolve software bugs soars after the testing phase.**



Source: IBM System Sciences Institute

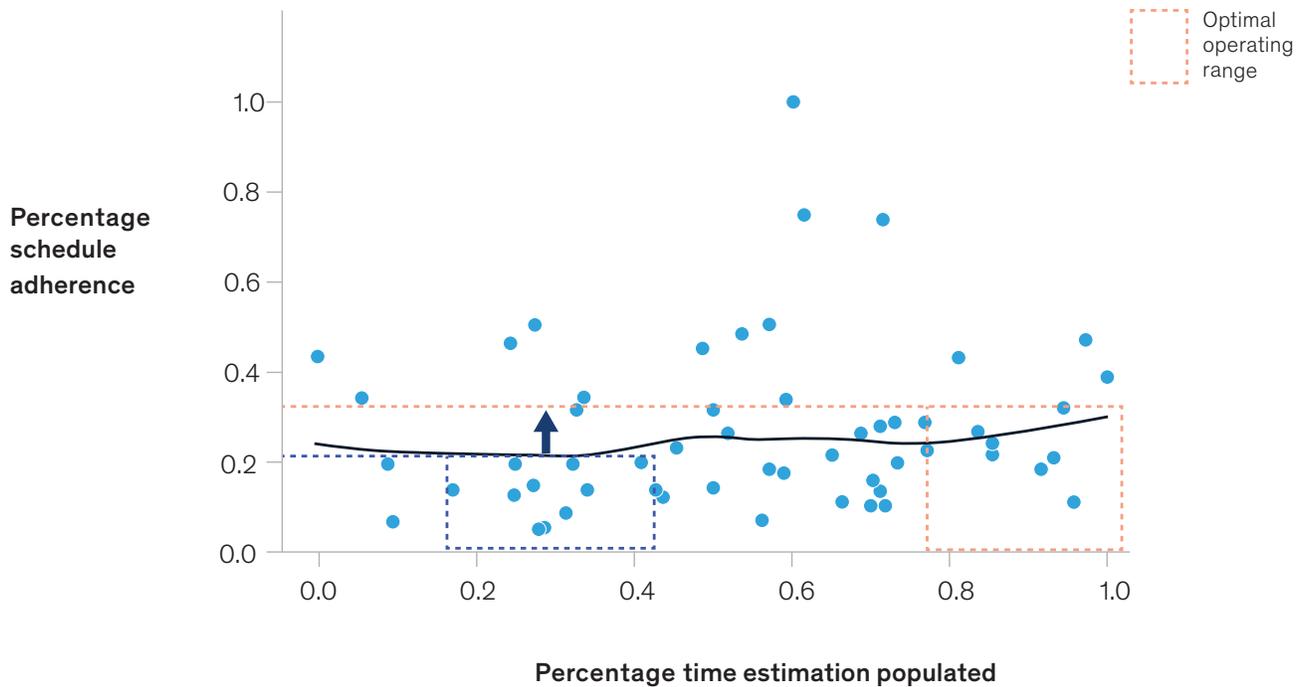
**Three agile team capabilities are significantly correlated to predictability of technology projects.**



Source: McKinsey analysis

## Capturing time estimates may lead to better schedule adherence.

### Impact of time tracking on project performance metrics



Source: McKinsey Technology Management Analytics

#### Semi-colocated and experienced team members balance quality and speed

Perhaps the most counterintuitive findings thus far are in the relationship we discovered between co-location and quality and speed outcomes. Our rule of thumb is that small, colocated teams are the best practice for projects of all shapes and sizes. However, for our initial purposes, a univariate analysis of colocation had double-edged effects on outcomes in one organization we looked at:

- Supporting our rule of thumb, co-location was associated with a lower rate of bugs. For example, comparing project teams that were 40 percent versus 100 percent colocated, the percentage of bugs in the latter group was 50 percent smaller, suggesting that co-location may well improve quality (Exhibit 8).
- The opposite pattern emerged when we looked at the normalized project time elapsed. The project time almost doubled along the same colocation percentages, implying that distributed teams may deliver faster than colocated teams (Exhibit 9).

Admittedly, our univariate analysis of quality and

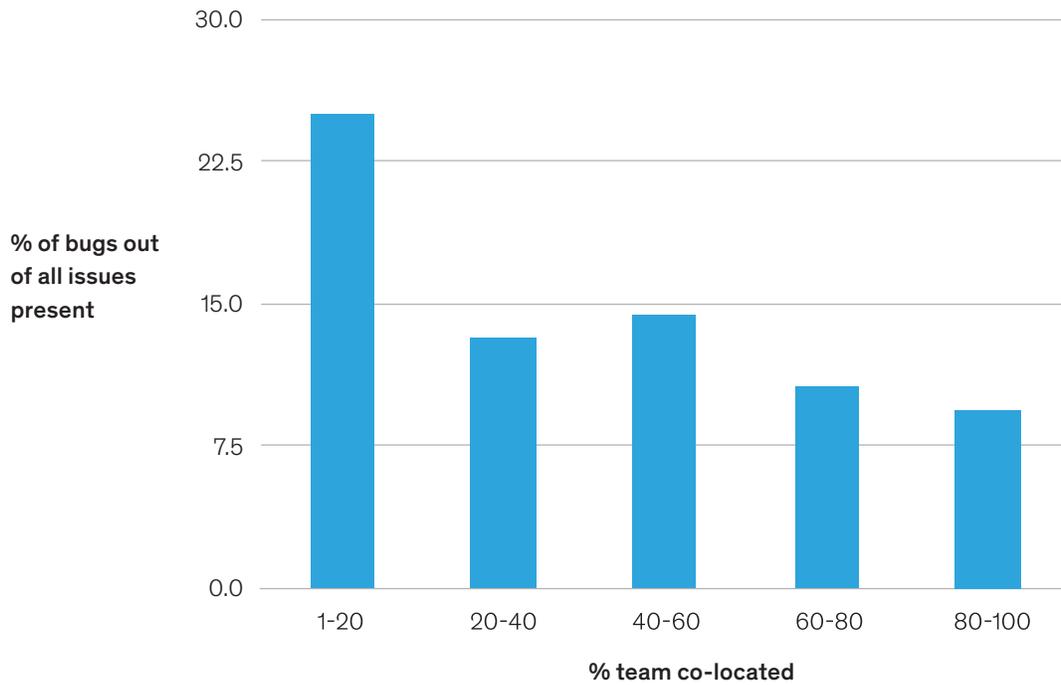
speed will require further refinement to include other architecture factors, such as the degree of code encapsulation via microservices, and the maturity of DevOps practices, which could more directly influence these outcomes. Further, a more robust analysis would also model how location decisions affect project costs relative to product market opportunity.

That said, one plausible explanation for the quality measure is that co-location minimizes communication and problem-solving hurdles, like prescheduling touchpoints for the many roles involved in development, including designers, full-stack developers, infrastructure engineers, database administrators, and more. Second, it is also plausible that the decision to 100 percent co-locate teams inherently trades off speed advantages that well-managed multi-time-zone teams may realize by developing and resolving issues around the clock.

Tenure and company project experience matter a lot, but team learning curves may vary significantly. Two measures—average team tenure (as measured by employee HR band) and cumulative projects per employee—both showed favorable relationships with

Exhibit 8

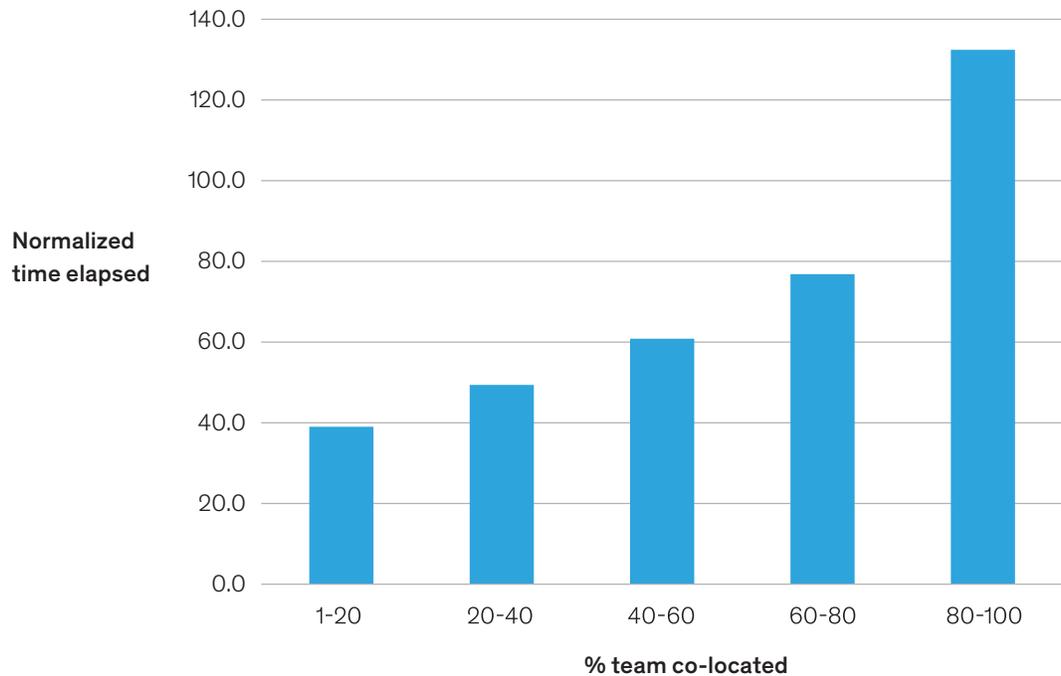
**Co-located teams seem to deliver better quality.**



Source: McKinsey Technology Management Analytics

Exhibit 9

**Teams that are fully co-located take longer to deliver compared to teams spread across time zones.**



Source: McKinsey Technology Management Analytics

quality and speed. On average, more senior teams produced fewer bugs and delivered faster. When we viewed the experience curve cumulatively over time, we calculated the learning curve for the financial-services and professional-services firms to be 1.5 and 3.5 years, respectively, to reach optimal outcome levels.

### **Case example: Software development at a leading IT services company**

A leading provider of IT services and enterprise application software in more than 150 countries was concerned about low productivity and long time to market for its software offerings. Several other key indicators highlighted the company's strategic and operational challenges across development teams:

- Doubling of customer incidents over 12 months
- Falling customer satisfaction scores, with less than 30 percent of software functionalities being actively used
- Fragmented, siloed product and engineering teams spread across more than 30 locations
- More than 100 million lines of legacy code, with less than 60 percent coverage of automated testing
- Agile processes only at the team level, with no focus on enterprise-level agility
- No way to holistically measure performance KPIs like innovation capacity, quality
- Fragmented data (stored in many different places without integration)

Identifying the main bottlenecks and productivity potential in the development process was the key to proposing and implementing any concrete improvement plans.

#### **The company's approach**

A machine-learning-based approach was used to assess levers of engineering productivity. Based on the priority areas for the software organization, several target variables were assessed: time to market, innovation capacity, and defects. More than ten data sources were integrated to enable previously impossible analyses. Then, priority levers were identified across foundational engineering practices, build/tooling, cross-team initiatives, culture, and talent.

#### **The outcomes**

Based on the analysis, the organization rolled out several initiatives, using a federated scaling and capability-building model. The analysis provided opportunity to realize a 15 to 20 percent increase in scrum team capacity, a 20 to 30 percent reduction in customer

defects, and a roughly 65 percent reduction in time to market. As a result, significant improvements would result in employee experience and customer satisfaction.

### **A promising foundation**

Clearly, technology-delivery organizations have come a long way since Fred Brooks documented his observations about engineering management. His look at programmer head count versus project schedule did not anticipate advances in product complexity, the speed advantages of effective global delivery teams, the maturing of DevOps and tooling, and the increase in agile organizations, let alone the vast amounts of digital metadata produced by tech-delivery tools, which can help teams build better products, faster.

At a minimum, these trends are long-needed building blocks for making decisions less intuition based and more fact based. They also may enable technology leaders to educate their business counterparts on a few drivers of high performance in technology delivery. More ambitiously, as companies continue the race toward digitization that confers cost and growth advantages, these data sources, methods, and management insights could evolve into an exciting field of technology-management analytics, enriched with objective outcome measures and data sources for the organizations willing to embrace them.

**Matt Brown** is an associate partner and senior solution leader in McKinsey's Waltham office, where **Ankur Dikshit** is a solution delivery specialist. **Martin Harrysson** is a partner in the Silicon Valley office, where **Shivam Srivastava** is an associate partner. **Kunal Thanki** is an analyst in the Gurgaon office.

Note: Future publications will examine the influence of other planning assumptions on quality and delivery speed, addressing questions including:

- How does the performance of colocated and distributed teams, teams in overlapping time zones, and multiple- and single-site project teams, compare in terms of quality and speed?
- How does the performance of project teams composed of contractors and those composed of full-time equivalent employees compare in terms of our outcome measures?
- What optimal ranges exist for project multitasking and task workloads per individual per time period?