How predictive analytics can boost product development

Complex product-development projects are plagued by schedule slips and cost overruns. The up-front application of advanced and predictive analytics helps companies build plans they can stick to.

Arjun Balaji, Raghavan Janardhanan, Shannon Johnston, and Noshir Kaka

R&D projects are inherently unpredictable. When embarking on efforts to design complex things, companies often have little idea how long a project will take, what it will cost, or what they'll finally be able to deliver to the end customer. Their initial project plans are sometimes no more than educated guesswork.

For example, in an analysis of more than 1,800 completed software projects, we found that only 30 percent of them met their original delivery deadline and one in five of these did so by removing or deferring feature content. The average overrun is around 25 percent of the originally planned schedule. The performance of a sample of over 1,600 integrated-circuit-design projects was even more telling. Over 80 percent of those projects were late, and the average overrun was nearly 30 percent. Moreover, those projects were almost as likely to suffer an 80 percent overrun as they were to finish on time.

Delays, and the extra resources needed to counter them, mean higher costs too. The average budget overrun experienced by a group of factory-automation-software projects we studied was more than 10 percent. A fifth of those projects cost over 50 percent more than originally expected. Then there are the indirect costs. Delayed launches mean lost sales, opportunities for competitors to get ahead, and potentially damaged reputations.

The underlying causes of overruns

We've spent more than a decade investigating the root causes of R&D scheduling and budget challenges. In that time, we've interviewed hundreds of project stakeholders, including executive managers, technical leaders, and program and project managers. They highlight many issues that boil down to two primary root causes.

The first root cause is underestimating the complexity of the project. Managers and engineering teams are often surprised by the combined impact of all the features and performance targets and the cost of integration into a finished product. Engineering intuition tends to be linear, while the cumulative effect of increasing performance, features, and quality is highly nonlinear.

The second root cause is overestimating the productivity of the development team. Planners tend to assume that the issues that befell their previous project would be cured and that no new issues would crop up. They assume that specifications will not change and that resources will be available when needed. In practice, of course, such problems do affect almost every project.

How predictive analytics can help

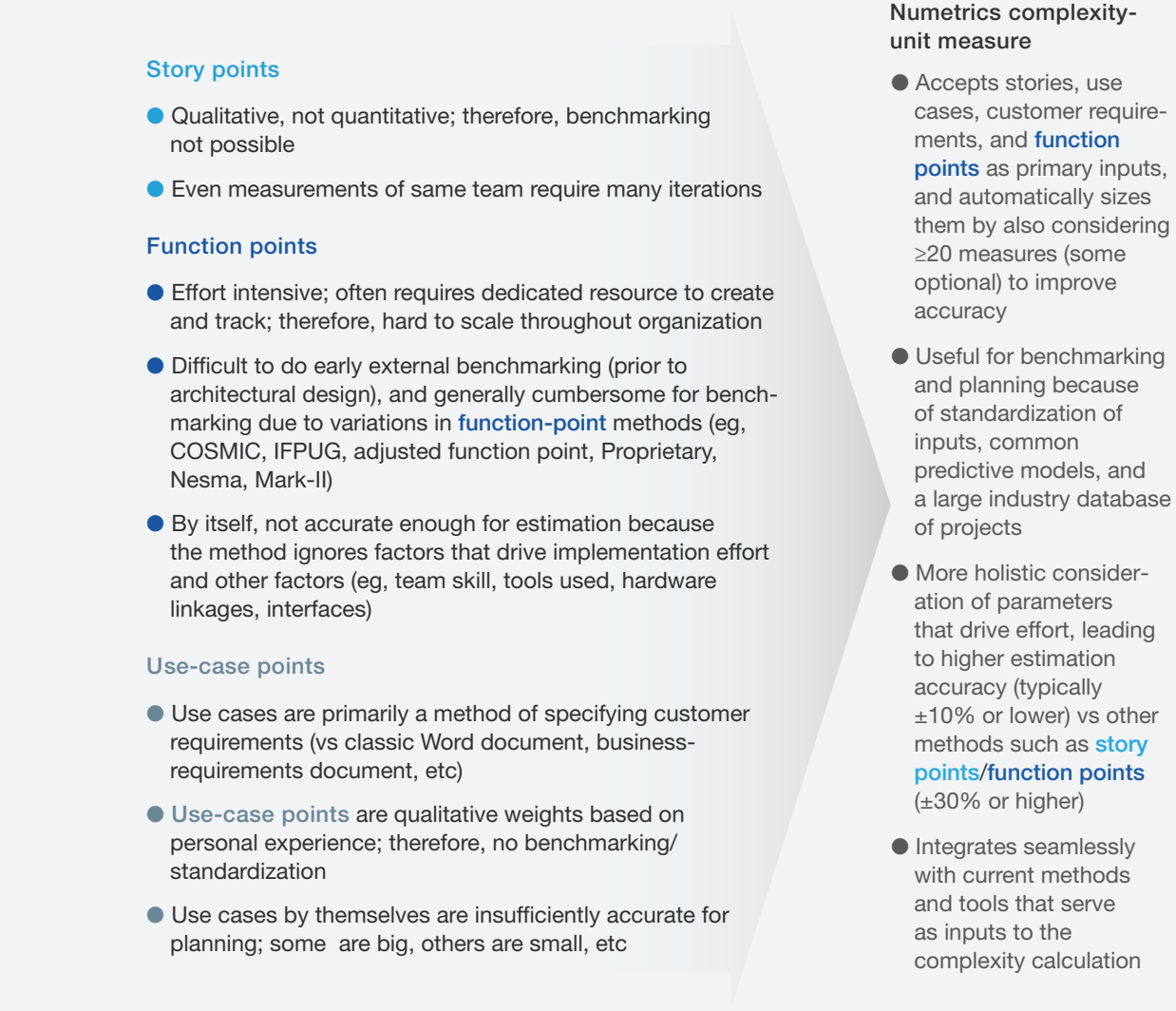
Until recently, even companies that understood and sought to address these issues didn't have effective tools for doing so. Conventional complexity metrics, like counting lines of code, story points, or function points (FPs) in software development, are difficult to estimate before the start of a project, especially one that requires many sprints from many teams to complete. Story points, by their nature, are qualitative and team specific, making estimation difficult when multiple teams are working on the same release. By their very nature, FPs focus only on function and not the actual effort drivers associated with implementation and validation, thereby leading to inaccuracies of greater than 60 percent in more than 50 percent of projects that use FP-based estimates.[1] And traditional methods often fail to account for other external factors, like the programming and development styles adopted by the development team, multisite development, and the impact of challenges the team is facing for the first time.

Today, some companies are adopting a new approach, one that uses powerful data analysis and modeling techniques to bring new clarity to the estimation of project-resource requirements. At its heart, the new approach relies on the fact that, while every

development project is unique, the underlying complexity drivers across projects are similar and can be quantified. If companies understand the complexity involved in a new project, they can estimate the effort and resources required to complete it (Exhibit 1).

Doing that is harder than it sounds. Companies must collect a significant amount of data to determine what factors really impact project effort. But for practical reasons, the only useful factors are ones easily measured, consistently gathered, and known early enough to drive budget and planning decisions.

Exhibit 1 A comprehensive complexity-unit measure solves for the limitations of effort-estimation methods.



McKinsey&Company

Developing a set of models, then, relies on an array of advanced analytics, machine learning, and artificial-intelligence techniques to predict the complexity and required development effort and schedule in a reliable way. In software engineering, for example, those models would need to understand the complexity of the system requirements, the architectures, the testing, and the potential required interactions with hardware.

Because these complexity models are based on real data, they don't make unrealistic assumptions about productivity. And their estimates automatically incorporate the effects of the everyday delays and disruptions that development teams must face. In other words, they take into account not only the complexity of the project (both the functional and implementation aspects) but also the complexity of the team environment.

These models can even identify the productivity impact of changes to working methods. Larger development teams are less productive than small ones, for example, as they must expend more effort on internal coordination and communication. The introduction of new teams, new platforms, or new development approaches can also hit productivity in the short term, even if they are intended to boost it over the long haul. With enough industry data, however, the models can see how these sorts of changes affected productivity in the past and provide a good estimate of likely future effects.

Better plans, smarter decisions

Armed with such models and a baseline of productivity levels for similar projects, a company can enter the current specification and develop higher-integrity plans for new products. It can then assess the risk of the current plan or create a more realistic staffing plan along with a good budget estimate and an achievable schedule.

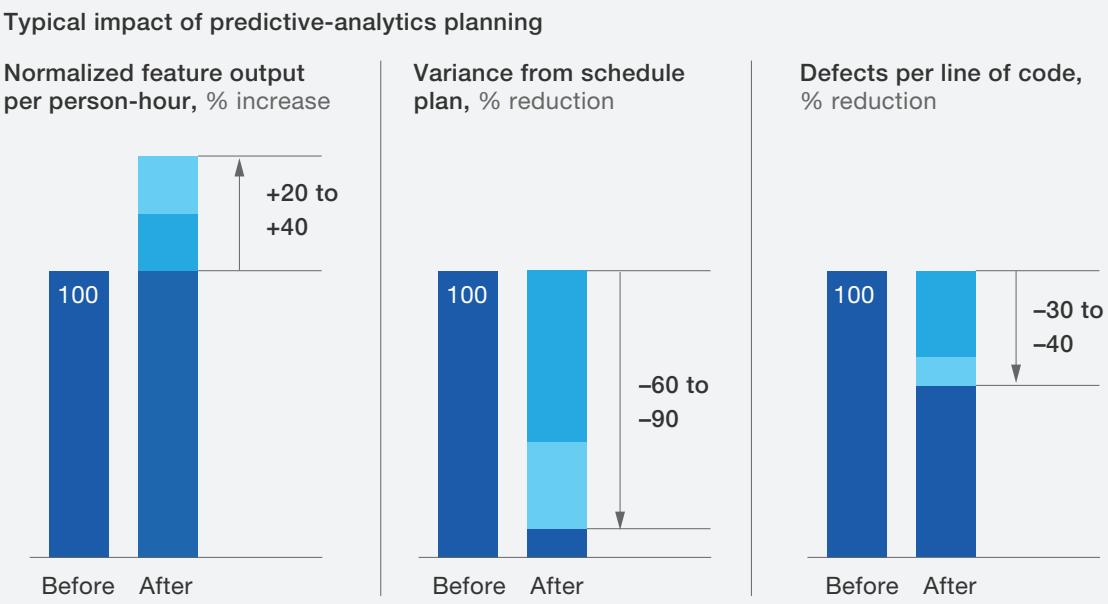
In addition, analytical models provide a powerful new way to deal with constraints. A company can model the resource requirements of multiple projects scheduled to run concurrently, for example, to see if there are any points where those projects will demand more staff than it has available for a specific role. With warning of such resource bottlenecks, it can take appropriate action—adjusting the schedules to separate the peaks in demand, bringing in contractors, or outsourcing part of the work. Similarly, the models will show if an aggressive budget or timeline can be made achievable by adding more resources. And if it can't, the company can run what-if analyses to evaluate the impact of dropping certain features or simplifying performance requirements. That allows a much more thoughtful, fact-based discussion, far preferable to missed deadlines or being forced to drop features at the last minute because they weren't finished in time for launch.

Predictive analytics at work

Predictive analytics have already have transformed the outcomes of some high-value projects (Exhibit 2). As an example, at one company, a project to create a derivative of a newly released product was originally expected to take just 300 person-weeks of effort. The project's planners arrived at this estimate on the basis that 90 percent of the new design would be carried over from its predecessor. When they reevaluated the plan using analytic models, they found that the project would actually take three or four times as much effort. The difference arose because while the amount of truly new work was small, it was widely distributed and affected nearly every part of the architecture. That meant significant extra testing and integration work, which the analytical models identified. Once the company understood the work involved, it changed its plans, keeping the team that developed the original product together to work on the derivative, and ultimately delivering it on time.

In another example, a company had a tight deadline to complete a new release for a big customer, with

Exhibit 2 Predictive-analytics-driven planning has delivered a range of positive impact on software projects.



McKinsey&Company | **Source:** Numetrics by McKinsey

competitors vying for the work. The predictive analytics models showed that with the company's current resources and project plan, it was going to miss its delivery schedule by 50 weeks. That delay would have caused it to miss the market window and lose a $350 million opportunity. Spurred into action by the finding, the company took steps to reduce the complexity of its design and prioritize the scope of the effort, resulting in a project that met the customer's minimum requirements and could be delivered on time.

Organizations that apply analytics and predictive tools to their product-development and project-planning processes see a dramatic reduction in schedule slippage. And because they can put the right number of the right people on their projects at the right time, they also enjoy R&D-productivity improvements of 20 to 40 percent. For companies, that means lower costs and lower risks—a powerful combination of benefits to have in a highly competitive environment. ■

[1] *The use of function points in the industry,* ISBSG, October 2016, isbsg.org.

Arjun Balaji is a partner in McKinsey's Bengaluru office, **Raghavan Janardhanan** is a partner in the Chennai office, **Shannon Johnston** is a specialist in the Toronto office, and **Noshir Kaka** is a senior partner in the Mumbai office.