Software development success is still elusive

Despite technology breakthroughs over the years, software development remains costly and error-prone. Some experts say that's mostly a people problem.

Patrick Thibodeau

October 24, 2011 (Computerworld)

The data about software development projects is sobering -- as it has been for decades. Too many projects still end up way over budget and behind schedule.

A Standish Group study of some 10,000 software projects found that 21% of them had failed, meaning they weren't completed or were rejected by the customer.

About 37% of the projects in the study were deemed successful: They came in on time and within budget, and users accepted the result. Another 42% were found to be challenges: They were late, over budget, not meeting users' needs, or all of the above, according to the report issued earlier this year.

The stakes involved in getting software projects right can be huge, and companies are constantly seeking new ways to avoid costly failures.

To untangle messed-up million-dollar-plus projects, IT executives sometimes turn to individuals like Billie Blair, an organizational psychologist and CEO of consulting firm Change Strategists Inc.

Blair contends that dysfunction in failing software projects usually originates with a project manager. "Anything that goes awry in a company can always be traced back to the manager," she said.

Often, a project manager isn't ready for the job at hand. Many managers are engineers, IT professionals or highly skilled software developers who were promoted under the mistaken belief that if a person has good technical skills, "it follows that they will be a good manager," Blair said.

The people issues in software development are getting more attention with the increased use of agile processes, which emphasize communication, collaboration, rapid production of code and frequent feedback among all members of development teams.
At Landmark Graphics, a user of agile processes, there's a focus on creating a workplace culture that emphasizes highly collaborative teams, said Todd Little, a senior development manager at the company. Collaboration is important in a process designed to constantly deliver working software "that's available for discovery," he said.

Little compared the constant review of agile projects to the traditional waterfall software development method, where "it's only at the end that you discover you have a real problem."

If there's a problem in an agile development project, "very rarely are they technical challenges," Little said. "Almost always, the challenges are with people."

Barry Boehm, director of the University of Southern California Center for Systems and Software Engineering, noted that while the agile process can get a project fielded early and evolve it as needed, it does have "some failure modes."

One problem is what Boehm called "easiest first," or "technical debt," which means that a developer, seeking to please users and customers, may put everything in main memory. "The thing performs like a flash, and the users love it" -- until the main memory is filled, he said. "[Then] you have an architectural breaker."

Boehm added that developers sometimes hold off on adding security features until it's too late, having already put in too many insecure features.

He agreed, however, that agile methods work well for smaller projects, as well as ones where requirements are rapidly changing, especially "in an organization where people feel comfortable or empowered."

This version of this story was originally published in Computerworld’s print edition. It was adapted from an article that appeared earlier on Computerworld.com.