It's not the coding that's hard, it's the people
The still-risky business of software development

Patrick Thibodeau

October 6, 2011 (Computerworld)

The data about software development is sobering. Many projects end up over budget and behind schedule, and one study puts the failure rate at one in five.

The path to better projects may be for software developers to become better people. A recent conference on agile development had "new age"-sounding sessions such as "An Introduction to Non-Violent Communication for Agile Coaches," "Fear Driven Impediments" and "Collaborating with Non-Collaborators."

The stakes involved in getting software projects right can be huge. Development projects costing millions of dollars can become such a mess that businesses turn to people such as Billie Blair, an organizational psychologist and president and CEO of consulting firm Change Strategists Inc., to untangle the disorganization.

In nearly every case, Blair said the source of all project dysfunction is the project manager.

"Anything that goes awry in a company," said Blair, "can always be traced back to the manager."

Often enough, the project manager isn't ready for the job. Engineers, IT professionals and highly skilled technical people can be appointed project managers under the mistaken belief that if they have good technical skills, "then it follows that they will be a good manager," said Blair. Businesses and organizations don't always recognize that management is a skill in itself, she said.

Project managers have to deal well with people, embrace conflicts and not run from them, know how to assist people in sorting things out, and be compelling, said Blair. Engineers and IT professionals "are wonderful at what they do and their skill set, but generally those
managing skills are not there and they have to acquire them," she said.

The people part of building software appears to be getting much more attention thanks to agile software development, which emphasizes communications, collaboration, rapid production of code and frequent feedback. It's a less rigid approach that requires a more flexible person at all levels, and not just at the manager level.

That is in contrast to some traditional development methods, waterfall in particular, which calls for getting all the requirements upfront and then moving through the development process one phase after another.

With traditional methods, "it's only at the end that you discover that you have a real problem," said Todd Little, a senior development manager for Landmark Graphics Corp., which makes applications for the oil and gas industry.

With agile development, you are constantly delivering working software at each stage of the process "that's available for discovery," and "each time you are closing in on reducing the overall uncertainty of the project."

To help accomplish that, Little focuses on the workplace culture, emphasizing highly collaborative teams.

When dealing with project problems, "very rarely are they technical challenges; almost always, the challenges are with people," said Little, who was chairman of the Agile 2011 conference in August, where the sessions mentioned above were held.

The Standish Group surveys software development for a study it releases every two years. Its most recent survey, released this year, collected data on 10,000 projects.

It found that 37% of the software development projects in its study were successful, meaning that they came in on time and on budget and that the users accepted them. It categorized 42% of the projects as challenges. These were projects that had problems such as being late, being over budget, or not having everything users wanted. The remaining 21% were failures, meaning they weren't completed or were rejected by the customer.

But the success rate of projects improved by 5% from two years ago, said Jim Johnson, the chairman of Standish. That may be due to more agile projects, as well as the types of projects undertaken during the recession, including fewer ERP installations.

Johnson offers caveats to the report's finding. Project failure, for instance, can be a good thing.

"If a firm doesn't have any failures they are not pushing the envelope," said Johnson, who cites Apple's development history, which includes failures that nonetheless have helped spawn successful products.

Johnson said an important key to a successful project is having a strong executive sponsor.

"People don't make decisions very quickly, and I think that's the death knell for projects," Johnson said. The reason the agile process works so well "is because it creates velocity" that leads to fast decisions.

Also key, said Johnson, is project optimization, or focusing on what's important. "Too many of them grow out of control, and you really need to focus on the real high-value items and work on those, and I think that's one of the things the agile process does," he said.
Ben Blanquera, who led agile development as vice president of IT at Progressive Medical in Columbus, Ohio, and is now a consultant, said the "premise with agile is you can break these things up to what is the smallest piece, the most viable product."

"The notion here is fast feedback," said Blanquera, "there has to be rapid iteration."

Standish's categorizations are debated.

The danger with the "challenge" definition, said Little, is that many of the projects that were delivered late or had less scope than originally intended "were nonetheless successful in the eyes of the business."

Agile has its own issues as well.

Barry Boehm, the director of the University of Southern California Center for Systems and Software Engineering, said agile is good at getting a project fielded early and evolving it as necessary, but it does have "some failure modes."

One problem with agile is what's called "easiest first" or "technical debt," which basically means that a developer, in the interest of pleasing users and customers, may, for instance, put everything in main memory "and the thing performs like a flash and the users love it" until eventually nothing fits in main memory anymore "and you have something that is an architectural breaker."

Developers may hold off putting in security features until a later point in the development, when they have already put in so many insecure features that there is no way to work security into it later, Boehm said.

Agile is good for smaller projects, and ones where the requirements are rapidly changing. It also works "in an organization where people feel comfortable or empowered," said Boehm.

**Patrick Thibodeau** covers SaaS and enterprise applications, outsourcing, government IT policies, data centers and IT workforce issues for Computerworld. Follow Patrick on Twitter at @DCgov or subscribe to Patrick's RSS feed. His e-mail address is pthibodeau@computerworld.com.