



How to Spot a Failing Project

AtTask CEO Scott Johnson Interviewed for CIO Magazine

July 17, 2007

Rick Cook, CIO

Usually, when an IT project fails, management is the last to know. But eventually, like a fish left too long in the refrigerator, the failure becomes all too obvious. When the situation reaches that point, your only option is the IT equivalent of pulling everything out of the refrigerator and scrubbing it out with baking soda.

But it doesn't have to be that way. Conventional wisdom to the contrary, project management is getting better. More projects are succeeding, fewer projects are failing outright, and projects are returning more of the IT dollar invested.

Still, only about one-third of all projects are complete successes. Often, the difference between success and failure is spotting the critical early warning signs that a project is in trouble. Here's a quick look at some of the earliest symptoms that all is not right with your "fish"—and what you can do about it before you have to break out the baking soda.

Be Reassured: Failure Isn't Preordained

The good news is that things are getting better. The most widely used measure of IT project success is the Chaos Report from The Standish Group International. The biennial (once every two years) report is based on a worldwide survey of several thousand medium to large companies. "We've measured project success every two years since 1994," says Jim Johnson, Standish Group chairman.

The original study (which is still sometimes quoted as if it were current) was shocking. In 1994, the researchers found that 31 percent of the IT projects were flat failures. That is, they were abandoned before completion and produced nothing useful. Only about 16 percent of all projects were completely successful: delivering applications on time, within budget and with all the originally specified features.

"As of 2006, the absolute failure rate is down to 19 percent," Johnson says. "The success rate is up to 35 percent." The remaining 46 percent are what the Standish Group calls "challenged": projects that didn't meet the criteria for total success but delivered a useful product.

"We've gotten better for several reasons," Johnson says. "The whole discipline around project

management is becoming more of a profession. We're understanding the process better. We're getting more articulate users who can describe what they want better. And we have some very good tools, like UML, that can help the user."

The Internet plays a role as well, according to Scott Johnson, CEO of AtTask, an Orem, Utah, maker of project management tools.

"Communications can go back and forth quickly," Scott Johnson says. "You don't have to wait for a status report meeting to find something has gotten hung up. You don't have the bottleneck of an individual trying to broker communications."

Scott Johnson points out that new project management tools tend to be easier to use than the ones of a few years ago. In addition, they often include features such as sophisticated trend analysis that can spot problem projects early.

Another change, Standish Group's Jim Johnson says, is the use of new project management methodologies in place of the old waterfall model where the entire project proceeds step by step from analysis to final delivery en masse. Both Johnsons are strong proponents of agile project management, which focuses on breaking projects into small chunks and delivering pieces of it fast for user feedback.

Looking for Warning Signs

While the Chaos surveys give reason for optimism, they leave little room for complacency. Nearly one in five IT projects still fails absolutely and more than two in five are partial failures.

What is perhaps more troubling is that the bigger the project, the worse the problems. "Seventy-three percent of projects with labor cost of less than \$750,000 succeed," Jim Johnson says. "But only 3 percent of projects with labor cost of over \$10 million succeed. I would venture to say the 3 percent that succeed succeeded because they overestimated their budget, not because they were managed properly."

(Perhaps significantly, agile project management is notoriously least effective on very large projects.)

Warning signs are different from reasons for project failure. Common reasons for failure include lack of management support and unclear objectives. Warnings are much more concrete and concerned with the day-to-day running of the project.

The Intangibles

The most important early warning signs are intangibles. The earliest signs a project is in trouble are hard to measure objectively, but easy to spot if you watch for them. Two of the most important, Jim Johnson says, are lack of interest in the project and chronically poor communications.

Lack of Interest

Often, Jim Johnson says, a lack of interest in the project's success results from a lack of real buy-in. Actors may have been pressured or cajoled into signing off on the project without really agreeing to it.

"Make sure everybody really agreed to what the project is going to do," he says. "Make sure everyone has the same goals even when they have conflicting agendas."

Other signals include people not showing up for meetings, not paying attention or just not saying anything.

"One of the things I've seen that helps projects along is a positive environment," Jim Johnson says. "That makes a big difference, and with a positive outlook you can do so much better."

The need for interest applies to the customers as well as the team. As Jim Johnson puts it, "You want to see active participation, active feedback and an energized user base." If that's not there, then the chances of project success are poor.

Poor Communication

Lack of communication, both formal and informal, is another early warning sign. If the stakeholders, from team members to users, aren't talking to each other, you've got a problem.

Ideally, project review meetings shouldn't contain any surprises because everyone knows—in at least a general way—what's going on with the other parts of the project.

Lack of Velocity

For Jim Johnson, and advocates of agile project management in general, "velocity" is a key concept. That usually translates into a lot of small deliverables at frequent intervals. Velocity not only makes tracking progress easier, but it's also good psychology; it reinforces a feeling of success and builds team morale.

"With IT projects, it's difficult to operate over a long period of time," observes AtTask's Scott Johnson. "You need the frequent small rewards of hitting smaller milestones. If you can plan them around things you can put in your customers' hands, that's even better."

"You want to have fast-moving items," says Jim Johnson. "You want velocity of deliverables. That's the real key."

"One of the classic signs a project is in trouble is that things aren't moving," he adds.

A “No-Bad-News” Environment

“This is a really tricky cultural thing,” says Raj Kapuor, executive vice president of the Center for Project Management, a software project management consultancy and education firm in San Ramon, Calif. “Everyone is allergic to bad news.” As a result, it’s all too easy to develop a culture where bad news is slow to percolate upward—which deprives management of vital, if unpleasant, information.

“You have to provide an environment where bad news is accepted,” says Kapuor. “That’s critical, and it’s not the job of the team members. It’s the job of the leader.” And by extension, the CIO.

Concrete Signs

All the early warning signs are not intangible. Some of them are very tangible indeed—if you know where to look.

Kapuor advocates using a dashboard tool that allows managers and others visibility into a project at the click of a mouse. “Organizations that do a good job of providing an executive visibility tend to have less troubled projects,” he says. “It prevents the green-green-green—and at the last minute red syndrome.”

Lots of Overtime

One early sign a project is slipping its schedule is teams working a lot of overtime. This is a particularly important indicator because assigning or encouraging overtime is the fastest fix the project manager has, as well as the one that attracts the least attention.

Generally speaking, a project running according to schedule should have little or no overtime. In fact, Standish Group’s Jim Johnson points out that the agile development process positively discourages overtime.

And as AtTask’s Scott Johnson jokingly observes, “One sign the project is in trouble is that employee health goes down for everyone on the project.” It’s the result, he says, of too much overtime leading to too much caffeine, too many late nights and too much junk food.

Diversion of Resources

Another sign of trouble is having resources, usually people, pulled off the project to work on something else. This can be anything from a panic response to another project gone sour to simple bits of “oh-by-the-way” work.

The problem is that, if you’ve budgeted your project properly, these diversions are cumulative. A few hours stolen here and there don’t seem like much, but they can quickly add up. “That’s

going to cascade down to the things [the project team] is supposed to be doing," Scott Johnson warns. As a result, it's important to keep precise track of the people and money being shuffled to other projects.

Ratios Trouble

Ratios are a standard financial management metric, but they are only beginning to appear in IT project management. According to Scott Johnson, a couple of key metrics for project management are the cost ratio and schedule ratio.

@Task software includes ratios that compare the budgeted time and money versus the time and money actually spent. "They allow you to show in a ratio where you should have been on cost and schedule versus where you are running to date," he says. "In a two-year project, you can know by week two that something is eating up costs or labor. It lets you know where you are versus where you think you should have been."

Metrics in general are vital to keeping projects on course. "In the absence of [clear metrics] you're relying on communication from project managers and others," warns Kapour. "That gets you in trouble because you tend to find out late that there's a problem."

Milestones Aren't Met

"Small, discrete and often" are the guidelines for the milestones for a successful project.

At AtTask, Scott Johnson says, milestones are weeks apart rather than months or calendar quarters. "Bite off a small piece, get it in someone's hands and start taking feedback," he advises.

"Discrete" means that each milestone is unambiguously defined in measurable terms with a solid deliverable. Jim Johnson prefers the term "stepping stone" to milestone. "At the end of a stepping stone you get a concrete result; a milestone might be more nebulous."

"Soft" milestones, such as percentage of overall completion, are notoriously troublesome because it can be hard to be sure the milestones have actually been met. Even worse are milestones based on effort expended rather than results. The classic example is measuring progress in terms of lines of code written.

Scope Changes (also known as the "Microsoft Mambo")

One common method of trying to shore up a failing project is to start changing the scope. The manager or participants may scale back or eliminate features, especially if the changes don't involve actually canceling requirements. This generally starts with relaxing requirements (such as response time) in an effort to reduce the development effort or to keep from discarding a problem module.

Requirements changes, per se, aren't a bad thing. In fact, they're pretty much expected in the agile method because of the constant user feedback the process produces. Indeed, changing the requirements can be a healthy thing.

The questions are which requirements are changing and why. The answers can tell you a lot about the health of the project.

And Finally...

Keep in mind that these signs are exactly that: signs. They don't mean the project has failed, or is about to fail. As AtTask's Scott Johnson points out, every "project experiences dips and gains," and you have to allow for them.

However, the signs may indicate the project needs closer attention from higher up the management chain. One useful rule of thumb for managers is to wait until at least two (short) review periods show problems before you do anything except watching closely.