



Project Management: Big Benefits from 'Microdeliverables'

Paul Glen for Computerworld

Too often, failing projects surprise us. How many times have you thought a project was going along fine only to discover when the delivery deadline drew near that everyone on the team was two months behind schedule? In situations like that, you wonder, "How could I have missed that this project was running two months late? What planet was I on where it appeared to be on time?"



Given that approximately three-quarters of all technical projects fail to meet their schedules, budgets or feature-set goals, you'd think we would be better at spotting groups that are "off the rails." The reality is that it's not easy to determine when a project is in trouble, and problems that seem obvious in hindsight are murky at the time that they occur.

Monitoring project progress is an important part of a leader's role. Knowing when and how to intervene in failing projects is critical to the overall health of any technology organization. Whether the intervention is to cancel a hopeless effort or to correct team skill or resource imbalances, managers need to spot difficulties early in order to avert disasters.

Of course, projects don't slip two months in one day. They fall behind a little every day, and the slippage accumulates until we notice it. So the question is, how can you notice the problems and fix them when they're molehills rather than mountains?

Most project methodologies call for monitoring task completion rates, counting hours expended and tracking the budget to check on the health of a project. Unfortunately, I find that these methods are inadequate to gauge real progress. Estimating task completion is notoriously subjective. The last 10% always seems to take 80% of the time. The number of hours expended has nothing to do with progress; effort rarely equals results. And although it's important to know how much of your budget has been spent, any positive correlation between the percentage of budget expended and percentage of project completed is generally coincidental.

The best method that I've found is to use what I call "microdeliverables." Most projects are planned with a series of tasks that lead to major deliverables: the documents, deployments or code that the tasks create. But these deliverables are usually the result of many people's work over a period of weeks or even months.

Microdeliverables are much smaller, individual efforts. When you plan for microdeliverables, each person on a project has responsibility for some physical product every few days. Then you can gauge the health of the project by checking whether the microdeliverables are done or not. You don't have to wait for months until a big deadline looms to check the health of a project.

When planning for and using microdeliverables, there are a few simple rules to follow:

1. **Never let anyone go longer than a week without owing a microdeliverable.** Any time a person goes longer than a week without a deliverable, he goes into a black hole of unknown progress. You can't really gauge how he's doing, and you're more likely to be surprised.
2. **Microdeliverables are either done or not done.** When measuring progress, there are only two states for microdeliverables. They're either 100% complete, or they're 0% complete. Progress is marked only by final approval of the item. Otherwise, you get into the subjective world of guessing how close to done things are, which is inevitably inaccurate.
3. **Progress isn't measured in effort, but in microdeliverables.** The only meaningful measure of progress is whether microdeliverables are done on time or not. If they're coming in late, the project is late. If they're on time, the project is on time.
4. **A microdeliverable is the responsibility of only one person.** If the deliverable is owned by more than one person, it becomes a problem to figure out where the real difficulties lie.

Using these simple rules, you can begin to identify project problems quickly and accurately and avoid the surprises that are otherwise all too common.

Paul Glen is the author of [Leading Geeks: How to Manage and Lead the People Who Deliver Technology](#) (Jossey-Bass Pfeiffer, 2003) and principal of C2 Consulting in Los Angeles. www.cxo.com