

Software Development
August 2004

You're Still Needed

Agile management has much in common with its traditional counterpart. Navigating boundaries, dealing with risk, and supporting the team will secure success.

By Esther Derby and Diana Larsen

What's left for me to do when my team goes agile and becomes self-organizing?" asked our colleague Tom. "I've been managing for 15 years ... I can't just go back and be a coder. Am I out of a job?"

Agile methods do turn the tables on traditional management: The team organizes and manages its own work, making many of the decisions managers used to make. What's left for a manager to do in the agile world? Plenty! They manage boundaries, team membership and risk, while championing the team.

Manage the Boundary

Boundary management starts with the project. An effective manager uses a project charter to develop a clear definition of the team's mission and its project community. The charter represents an agreement between the team and the sponsor funding the project.

The agile manager acts as the lead negotiator in writing a charter; she brings disparate points of view into alignment, ensures that the team will have resources adequate to the task, and knows clearly what work they will do.

The charter puts the project in context and lays out its business rationale as well as the problem that the project will solve. Charters document what's in, what's out, what resources are dedicated, and what "done" means.

The charter should also document four types of tests: functional, quality, performance and management. The first three tests establish what "done" means. The last type of test may be unfamiliar, but it's important information that informs the team of the software's business goals. Here's an example of a management test from the Apple Computer iTunes project: "Our new service will register at least 1 million song downloads during its first month in production." Management goals are external to the team. The team may not be able to influence the market, but awareness of the product's management vision and intention can certainly drive design decisions.

Charters help the team focus: When a team understands the reasons behind a project, they're not working on isolated tasks; they're working toward a common goal. And a good charter helps the project manager weigh the inescapable trade-offs and changes that occur on every project.

All manner of transactions flow across team boundaries: resources (machines, money, people), information, materials and ideas. Make sure that whatever passes through the boundary heightens the team's productivity

and well-being. Sometimes this means guarding the boundary to keep intrusions out. At other times, it means looking outside for necessary resources.

Manage Team Membership

Managing team membership involves hiring and incorporating new people. The effective agile manager employs strategies to bring neophytes up-to-speed with as little impact on other work as possible. If your team is pair programming, pair the newbies with your most experienced members. If your team isn't pair programming, assign a mentor programmer to be the team anthropologist—someone who explains team customs and orients new members to the software and the domain.

A manager of a company-wide initiative to make software development more responsive to customers once told us that in hiring, he looks first for solid basic programming skills; after that, attitude is everything. “People can always learn new or better skills, but if they aren't flexible, willing to communicate and open to new ways of doing things, they don't have a place on my agile team.” Each prospective team member is first interviewed by the manager and two or more team members, then spends a half-day pair programming and interacting in the open workspace. The manager makes the final hiring decision—based on the team's recommendation.

Managing team membership also means moving people off the team when they inhibit productivity and morale. On one of our past projects, the XP team brought in two team members from other parts of the organization. One very bright, highly experienced programmer contributed good ideas to the team but was unwilling to entertain anyone else's.

The other individual was always “too busy” to pair or attend standup meetings, preferring to stay after hours and work alone on weekends. His desire to appear super-productive hurt the team. Often, his work was full of code smells, and other team members took on extra work to rewrite it. His predilection for overtime provided an object lesson in the value of maintaining a sustainable pace and the downside of the “hero” culture.

The manager could see tension escalating as a result of these two individuals' inability to work as team members, and sat down with them to “coach” them off the team. One left the company; the other relocated to a different department.

Sometimes, managing team membership means actively keeping people out—for example, when senior management decides to throw more bodies at the project to meet a deadline. Adding more people late in the project usually slows progress and depresses productivity, as experienced team members drop their work to bring new people up to speed.

Manage Risks

Agile methods reduce requirements and schedule risk by producing working software every few weeks, but the untoward events that plague traditional projects affect agile projects, too. What if a key piece of equipment fails or a vendor doesn't deliver a promised component on time? What if the company merges with an organization in another time zone (or another country)? Because agile managers don't spend all their time updating a project schedule, they can play an active role in managing such risks.

For example, the higher the number of “indispensable” technology experts on your team, the higher the risk associated with losing one or more of them. A 28-person team we recently worked with had two developers who had become bottlenecks precisely because of their expertise: one was a data warehouse guru with the only strong Sybase skills on the team; the other was a whiz on a legacy system that no one else understood.

Though the team was effectively using pair programming for other parts of their effort, each time something touched the database or the legacy system, it had to go through one of these individuals. The bottleneck and resulting backup put iteration goals at risk.

The manager moved into action. Armed with information about the effects of the bottleneck and the risk of losing one of the essential team members, she negotiated for the time and money necessary to cross-train other employees on the scarce, but vital skills.

Be a Team Champion

Maintaining a common vision, agile managers encourage their teams to explore what's possible rather than relying on yesterday's approaches.

Running interference for the team is also vital. Handling PR duties, for example, ensures that your team keeps working on the software rather than spending its time explaining the process. And keeping outsiders informed is an effective way of greasing the wheels for your team: Educate the people outside your team who affect your project to help them understand the roles, practices and disciplines of agile development.

Sustainable pace—a.k.a. the 40-hour week—is a vital tenet of XP and other agile methodologies. If you work in an environment that measures productivity in hours or checks for cars in the parking lot on weekends, you must advocate for a cultural change and revised expectations for a sustainable pace.

The agile manager also works to remove the environmental roadblocks that hinder productivity. For instance, if you want your developers to consider the team's needs before personal goals, a performance appraisal process that focuses on individual contributions will be counterproductive. Begin talking with your HR professionals early on to enlist their help. They may be able to do research for samples of alternative processes that are supportive of a team structure and fit well with their current systems.

Facilities management and workspace standards can help or hinder a team. Bring the folks from facilities in as creative problem solvers. Explain the need and rationale for an open workspace, the space to pair program or extra freestanding panels to hold “information radiators.” Champion your team's needs for a workspace that supports teamwork.

In our experience, agile teams waiting on deliverables from traditional teams are at a disadvantage. Code from traditional groups, particularly if they're working on a legacy system, is likely to have more bugs and code smells, and be tougher to integrate. Work with the managers of those groups to negotiate turnover standards that let your team focus on building working software rather than finding bugs in turned-over code.

Not So Different

Now that we look at it, agile management doesn't seem so different, does it? Sure, the focus will shift, and the team will take on much of the work planning and progress tracking that, in traditional approaches, is strictly managerial domain. But you'll still be doing what good managers have always done: coaching and encouraging team members, making sure they have the resources they need, removing obstacles, and influencing peers and managers outside your team.

Esther Derby (derby@estherderby.com), a certified Scrum master, coaches technical people making the transition to management and works with teams to find pragmatic solutions to the obstacles they encounter building software. *Diana Larsen* (larsen@futureworksconsulting.com) works with software development organizations and teams to strengthen their ability to create and maintain company culture, manage change

and improve project performance.