Scaling agile isn't such a stretch

By Lisa Morgan

November 26, 2012 — Being agile is more critical than ever as businesses compete for customers. The true level of agility can vary greatly from company to company, team to team, department to department and person to person. As organizations scale agile out from pilots and small groups to critical projects involving hundreds of developers, the dynamics can change in ways that may not be anticipated.

When a pilot has gone well, people may assume it's a recipe for success, but later on they discover it isn't a panacea after all. For one thing, agile practices require changes in mindset and behavior, especially among those who have been deeply entrenched in traditional methods. It may also be that agile pilots are treated differently than other projects.

"Quite often a pilot project gets a lot of attention: You have access to more resources in terms of external coaching, training and attention from senior management because everyone is watching and wants to make the project successful," said Flowmotion business productivity coach Collin Lyons. (Flowmotion is a consultancy that helps companies manage change.) "When you expand agile [out], you can underestimate how much those things contributed to the pilot project's success."

What's more, the type of people who worked on the pilot may differ from the people assigned to larger projects.

"One thing that happens when agile adoption starts is smaller teams become very collaborative, and they also have a lot of champion resources," said Charlie Li, vice president of global quality and testing services at Capgemini. (A champion resource is a highly skilled person.) "Agile [initiatives] tend to have people who are well-rounded: QA people who can often code and developers who can work on architecture or design, [whereas] offshored and outsourced skill sets are very specific. When you have 20 people, it's easy to say you're going to hire 20 champions. But if you're rolling out to 50,000 IT individuals, they can't all be champions because the cost structure won't work."

Capgemini publishes an annual World Quality Report that has indicated that the need for multi-skilled talent is growing. In the first year, 2008, respondents considered testing skills important to QA. The next year, they said QA engineers should also have software development skills. Last year, the ideal skill set extended to domain expertise.

"This year, people want a left-handed Japanese ballerina that can do all of the above," said Li. "Well-rounded people have been considered champions, but in the future they may be considered the norm."

Pilots may also be populated with people who elected to work on the project. "The people who choose to work on a pilot are the kind of people that want to try new techniques and are willing to step out of their comfort zone," said Thomas Stiehm, an agile consultant and CTO at Coveros, a product and service provider that helps build secure software applications using agile methods.

"[When you move] agile out to large groups, some people won't want to change and they don't want to be agile. The resistance to change can manifest in a number of ways that can lead to poor adoption results that doom enterprise-wide adoption."

Some may view agile practices as too informal. Some may have seen agile projects fail in the past, or they may view agile as an excuse to do less documentation. Meanwhile, managers or executives may assume that existing resources and skill sets can adapt to agile more easily than they can in reality.

"Certain technologies and tools can be used effectively in a small team, but when you try to roll that out to a large organization, there are politics, issues, cultural changes and skill changes that become very challenging," said Li.

Shinetech, a China-based software development outsourced service provider, seeds members of the initial smaller groups into larger groups to ease transitions, according to John Vanderpool, VP of Global Operations.

Agile: More than a fashion statement

Expanding agile out also means expanding agile up. While the term "agile" has become fashionable, not everyone has adopted agile ways of thinking and working.

"Businesspeople may think [expanding agile out] is just another change-management initiative because they don't recognize it is a fundamentally different way of doing things," said Lyons. "It affects how human resources brings people on, how finance releases funds, and how a salesperson pitches what's being sold in the next software release. A common mistake is underestimating that it's a cultural shift."

As agile expands, some processes may become inefficient or not applicable. Because the details of agile software development don't translate to marketing or finance, it's better to focus on agile outcomes that everyone can comprehend.

"People often assume that it's the techniques and practices that [make agile projects a] success, but they're missing the point. It's the outcome that matters," said Lyons. "You have to talk at the principle level so people outside software development understand agile in context."

When outcomes (rather than specific techniques) are used as a template, teams can have the freedom to choose how best to achieve that outcome. "There's a perception that repeatability is inherently good [when there are variances] in the types of projects that people are doing, how those projects are set up, the scale of the projects, and whether the projects are legacy or greenfield," said Lyons. "People don't implement agile because they want standups or pair programming. They want the outcomes agile delivers."

Executives and managers may not realize that while they want teams and individuals to make the changes necessary to become agile, they may not want to change themselves.

"A lot of organizational policies have ossified procedures that don't create pain until the organization is trying to move quickly," said Brian Button, VP of engineering and director of agile at IT consulting firm Asynchrony Solutions. "When agile teams tend to press on those things, it's the higher-level management that needs to start changing the culture of the overall organization. When I talk to higher-level management, I try not to talk about agile because they're more concerned about outcomes: delivering better software to customers faster and more predictably more so we can ship products on time."

Preparing for transformation

The road to agile adoption may be paved with changing work habits, expectations and tools. One thing that can frustrate the transition is a perception that agile is causing more problems than it is solving.

“Even at the pilot level, agile starts to surface problems in the organization, but people may think that [agile is causing] those bumps in the road,” said Lyons. “When decision-making is slow, pushing things out to production is slow, or getting decisions made on requirements is slow, all of that can be seen as agile being the problem when in fact it’s just showing flaws in the organization’s ability to move quickly. If you’re an executive that ‘gets’ agile, you love it.”

If executives aren’t willing to tolerate the failures agile exposes, teams and team members will be less likely to experiment or change. “If there’s no executive willing to put his job on the line to make it happen, you’re going to have some very serious challenges,” said Button. “The top level has to give the rest of the organization permission to change. Otherwise, it will be fought all the way down.”

In addition to a cultural adjustment, more or different tools may be brought in. “Agile involves so much collaboration that you need a traceability tool that can link what happened in one release to a specific business problem in a past release, identify the parts of a project that are more prone to error, or reveal which code design caused performance issues so we don’t do it again,” said Capgemini’s Li. “Those types of analysis are mandatory so you can continue to improve the way you develop and the way you release. Otherwise, you’re simply driving blind, and you can’t go back and make improvements.”

More reporting is common as agile expands because managers want to understand who is assigned to what as well as the status of projects. “As you get more non-developers involved, you need more reporting,” said Yair Flicker, president of SmartLogic Solutions, a Web and mobile development firm. “If you’ve got dozens of developers, they need to spend some developer time writing reports.”

Jeff Marcus, CEO and CTO of Sparkway, disagreed, because more time spent writing reports means less time coding.

Generally speaking, tools can be a point of contention, especially when they force people to work differently. Individuals and teams typically have favorites, but the enterprise usually requires something that provides visibility and traceability across projects and teams. If tools are hard to use, people will find excuses not to use them, or productivity will take a hit.

“I’ve used a lot of really sophisticated tools and seen people get sucked into a vortex of complexity,” said Marcus. “I think about how I can make things simple. If I need something more complex than the tool than I’m using, then maybe I need to simplify my approach rather than getting a more sophisticated tool.”

If a more sophisticated tool is needed, avoid buying tools based on feature sets or slick demos. Lyons recommended that before buying tools, enterprises should keep using the tools they have until they understand why they need one.

SaaS options have become popular for new tools because they make for easy onboarding, although the subscription models may not align with the needs of agile groups or enterprises. While monthly subscriptions are popular, enterprises are apparently pushing for pay-per-use models that are based on weekly, daily or even hourly usage, Li said.

Managing risks

Risk management is one reason to ponder how agile practices may be affected by scale. “Once more resources are affected and applied, there’s a greater cost and greater exposure so failures will have a bigger impact as the project increases,” said Marcus. “Simple projects tend to succeed and complex projects tend to fail, so the core question is why do larger projects seem geometrically more complex than small projects?”

For one thing, more people are involved and more people usually mean more politics. Dependencies are another factor.

“Each project has a risk of failure. When you have two interdependent tasks, then the risk doubles, so you’re twice as likely to have a failure,” said Marcus. “The resolution is to have small, independent groups with self-contained goals and minimal dependencies. To reduce the dependencies and streamline the critical path, you want to run everything as its own agile business.”

People may assume that a small group ran better than it did, especially in the absence of dependencies. In a small group, it’s relatively easy to identify and rectify issues, so they may go unnoticed.

“Small groups have the luxury of flexibility that the larger, more interdependent teams don’t,” said Marcus. “Large projects with many interdependencies cannot be flexible. For one thing, you need to buffer those subprojects with lots of extra time, because if one fails, many of them will fail. If I ate up my buffer, the risks go up, so it’s better to reduce those dependencies. You also want to make use of automated testing because when you want to pivot, you can at least do it with confidence and discover your problems early.”

While Marcus advocated minimizing dependencies, Coveros’ Stiehm said large groups need to learn how to manage those dependencies. “Large-team agile requires a different kind of rigor around defining subsystem or component boundaries than small-team agile,” he said.

“You can’t have 100-plus people with collective code ownership of a multimillion-lines-of-code project like a team of four can own 100,000 lines of application code. You need to have more formal agreements on the boundaries of responsibility with specific teams owning specific subsystems. Within those teams the same rules of code ownership and change apply, but between those teams there needs to be more formal change processes. Teams need to put commitments with each other in place regarding how they will work together and how their parts of the system will work together.”

Infrastructure is a barrier to globally distributed teams because not all locations may be getting the same level of service. Some organizations have addressed the problem with mirroring and in-memory enhancements. SmartLogic’s Flicker keeps backup routers and switches on hand in case a failure happens.

“The most precious resource is developer time. We don’t want to lose half a day because that’s pretty significant,” he said. “We have a suite of CI servers and a suite of deployment scripts, and we use the cloud pretty extensively for deploying and setting up servers. That way, we just click on a button on a website and we have the computing power we need.”

Flicker and his team have standardized processes, so when a new project starts, the tools are ready to go and the developers can start writing features instead of setting up deployment scripts and continuous integration.

Corporate system administration groups have told Stiehm that it will take four months to deliver the servers his team needs for a three-month development project, which obviously doesn’t work. Like Flicker, Stiehm advocated cloud-based provisioning and access to resources.

Who should be involved in the expansion initiative?

Expanding agile out involves a lot of people outside development and IT. Everyone needs to be involved in the transitions, including an executive sponsor with political clout. And, everyone has to perceive some level of agile success or they may fall back into old habits.

To demonstrate progress, Shinetech defines quick wins and short-term goals with teams, and it enhances the work environment. Flowmotion’s Lyons crafted an improvement backlog in cooperation with his clients that identifies problems, prioritizes them, and has criteria for determining whether the problem has been solved. That way, the organization can move through a prioritized list and prove what progress it has made.

“Everyone wants to say they’re agile because it’s a buzzword. Who *doesn’t* want to say that they’re agile? Every CIO I talk to wants to release things faster,” said Capgemini’s Li. “Obviously, there’s a lot of room for improvement, meaning, are they faster than last year? Perhaps yes, but are they adopting agile methodologies across their IT organizations? Probably not.”

As with any other technology-related change, the CIO or his or her boss is considered the ideal executive champion to push things forward because they have the power to make decisions and set expectations that affect the entire enterprise, which a team manager lacks. “Executive participation and support is extremely important. The organization values what its leadership values, so if the leaders aren’t part of the adoption, the teams will not see value in it and will not work to make it successful,” said Stiehm.

“Most of the individuals on the teams also have to buy into agile as well. Not everyone will buy in, but a successful adoption does require most of the people in the organization to buy into it.”

SmartLogic’s Flicker has been in situations in which non-technical technical people—such as in marketing—tried to dictate the technical process, which didn’t work well. “A technical person needs to dictate the technical process, but that person also needs to understand the business issues,” he said.

"You definitely need somebody from the marketing team because they dictate the features that go into the product and represent the end users. They're responsible for doing all the market research, so they should be speaking to customers/end users to make the project more successful. They should also have some sort of feedback loop so they know that the features they requested are getting done and when they're getting done."

Getting everyone on board with agile often involves training to expedite the understanding of agile and agile ways of working. People are sent to a few days of training, after which they are expected to be agile, but the problem with batch-method training is that the people who attend the training may not retain everything they learn.

"The half-life of knowledge is very short," said Lyons. "People start using what they remember, but they can only use so much immediately so everything else they learned in training is gone. What works is a giving them the little bit of knowledge that they need to know right now to address their current situation and have them apply that knowledge now in context. Then a couple weeks later, after they've learned that, we teach them something else."

The total amount of time Lyons spent training is the same as batch classes, but it's spread out over three months so the knowledge is applied and retained.

A word about security
Because some may consider agile less disciplined or rigorous than traditional methods, agile may be viewed as exposing the organization to more security risks. "It's a misconception that agile is an undisciplined way of working. In fact, it's extraordinarily disciplined because so much stuff gets exposed and there are so many opportunities to recognize issues," said Lyons.

"You need to bring in the security and operations people. These are the people who have an interest in non-functional requirements of a system. Historically, they would have an opportunity to review and approve certain things or contribute to the definition of requirements. In an agile context, that doesn't change, it just means they need to be included all along."

Some Capgemini clients, particularly those building desktop applications, are not yet including security as a standard systems development life-cycle item. Apparently, mobile development is starting to change that trend.

"Security has been ignored simply because for a long time it's been unclear who should be doing security testing: The security team or the QA team. It's taken clients years to decide where it really sits," said Li. "Most organizations have decided that, for organizational security of machines, it's the security teams, but from an actual user standpoint, it's more of a QA activity."

If an organization wants to be more agile than discovering security issues after the fact, it has to bake security into agile processes. "At scale, the problem is usually that application security isn't valued as much a new features, at least not until something happens and it is too late," said Stiehm.

"Most large organizations have an application-security group that has application-security standards and practices. While the people in that group are often very knowledgeable regarding application security, they put in place practices that slow the team down, and most software developers don't like being slowed down by practices they don't value or understand."

If the application-security group lacks political power, its application-security requirements may be ignored. Add to that the deep desire and financial incentive to get to market fast, and the end result may well be that security problems surface in production.

"A lot of people say agile practices increase security risks, and even if security has been nailed at the small-group level the issue becomes more complex with scale. It's a containment problem," said Sparkway's Marcus. "The more autonomous you make working groups, the simpler it is to contain sensitive information. If you've got a lot of interdependencies where people have to trade information a lot, it's hard to manage. If you have autonomous groups that only share some information, it's easier to manage."

There are ways to build security into software, such as using frameworks and libraries, or using automated tools to make security part of the build. If teams understand what it takes to build secure software, and if they understand the benefits of building more secure products, it may be possible to solve most problems at the team level. However, security should be viewed as important at every level.

"You can work [security] into your workflow as you're producing software so you don't have to do those kinds of checks at the end. And if you do those checks at the end, they should take less time if you built security into the app," said Asynchrony's Button. "At scale, the cost of a breach probably goes way up, and then all it takes is this one loophole in your system to let that first person in. That just means you need to train your people better and give them better tools to find things earlier. Have your testing group pound on it and let your developers learn from that."

When SmartLogic Solutions worked on a massive project with JP Morgan Chase, a third-party penetration testing team was hired to supplement internal efforts.

The best way to ensure a smooth transition to agile is to limit its scope initially and succeed with that implementation. Rather than trying to "roll agile out like a carpet" (as Lyons put it), consider what will likely change with scale and make adjustments as necessary to enable continuous improvement. Also, remember to focus on outcomes.

"It is a learning opportunity to see what it takes [to become agile] and to decide if you want to put your organization through that change process. The answer could be no," said Coveros' Stiehm. "Change is hard on all levels. Even the best organizations will reject change because they aren't ready to go through a hard part right now, because there just isn't enough incentive."

The real incentive is getting to market faster with better-quality products than the competition, but achieving that goal is easier said than done.

Related Search Term(s): agile

Share this link: http://sdt.bz/37185