

From Agile Pilot Project to Enterprise-wide Deployment: Five Sure-Fire Ways to Fail When You Scale

By: Robert Holler and Ian Culling

Introduction

A majority of software professionals who evaluate agile practices on a pilot project are so satisfied with their experience and the associated results that they would like to replicate this success throughout their organizations. However, transitioning from small-scale victories to large-scale successes can be a complex process. While there is no perfect formula to scale that guarantees success for every company, there are several common mistakes that companies should avoid when making the transition:

1. Skimp on training and education
2. Practice Agile In Silos
3. Ignore other areas of the Business
4. Encourage organizational complexity
5. Fail to scale the infrastructure

These common mistakes can derail even the most carefully prepared strategy. This paper seeks to discuss each pitfall, the consequences it could bring to an organization trying to expand agile practices, and steps companies can take to avoid them.

Contents

Introduction	1
Executive Summary	2
Mistake #1: Skimp on Training and Education.....	3
Mistake #2: Practice Agile In Silos	3
Mistake #3: Ignore Other Areas of the Business.....	4
Mistake #4: Encourage Organizational Complexity	4
Mistake #5: Fail to Scale the Infrastructure	5
Conclusion.....	6

Executive Summary

Scaling agile is a significant undertaking for any technology organization. The implications are far-reaching and diverse enough to have entire books dedicated to them. Consequently, the aim of this paper is not to present a magic formula for scaling but instead highlight several common mistakes that organizations make when expanding agile beyond a limited number of teams.

Perhaps the most common mistake companies make relates to training and education. Training can obviously be time-consuming and expensive, but when changing a fundamental business process, ensuring team members truly embrace the Hows and Whys of agile is critical. While much emphasis is expended on training pilot teams, many organizations assume that these pilot team members have all the information they need to bring the rest of the organization up to speed. This assumption fails to take into account not only the value of extensive experience and broad expertise that a professional agile coach can bring, but also the limited bandwidth of internal team members, whose primary responsibility is developing code.

Another common mistake is to attempt to scale without addressing fundamental process inconsistencies that may have evolved amongst various agile teams in the organization. The move to agile is often initiated at the team level and teams often make the transition to agile methods without knowing that others in the organization have taken the same steps. This grassroots effort is effective for initial adoption but can create a series of challenges when trying to share information and standardize processes and practices throughout an organization. General process consistency is critical in streamlining the enterprise-wide adoption of agile development.

Executive management often assumes that agile only applies to software development teams. Once agile begins to be adopted throughout an organization, however, companies are often required to (re)examine lines of communication, human resource (HR) policies, management practices, and even organizational values to ensure they align with the team-centered nature of agile development. Understanding that agile impacts multiple departments and organizational standards is crucial to scaling successfully.

A similar problem many companies confront is how to manage more and more agile teams while avoiding unnecessary layers of bureaucracy. Transitioning to agile requires that organizations relinquish levels of control in return for greater adaptability, improved visibility, and accelerated delivery. There are some simple yet effective management practices that do not entail significant overhead and can make scaling more straightforward and less onerous for the overall organization.

The final obstacle discussed in the paper is the failure to implement the necessary infrastructure that will support a full-scale agile implementation, especially in terms of enabling tools and technologies. In a few instances status quo may be sufficient, but in most cases, new and more complex business needs drive a series of technology challenges that must be addressed organization-wide versus simply at the team level.

Scaling can obviously be a daunting challenge, one in which many teams fall into similar traps. Organizations should strive to deliver adequate training and access to agile expertise, ensure process consistency, implement appropriate organizational changes, keep simplicity at the forefront, and deploy tools and technologies that align with new and evolving business demands. Adapting the above to the needs of the overall organization will help keep teams on track in their scaling efforts.

Mistake #1: Skimp on Training and Education

Perhaps the most common mistake companies make when scaling agile is to skim on internal training and education. When first beginning agile, pilot project teams typically are encouraged in their pursuit of agile training. Often the entire pilot team is sent to a class to learn about the processes and/or the practices of various agile development methodologies.

Teams may be trained in test-driven development, agile planning and estimation, Scrum, automated testing, and more. As a follow-up to these training activities, many early adopters also bring in agile coaches and consultants to reinforce their training. The teams often work with these experts until the new practices or processes are established habits that can deliver reliable results.



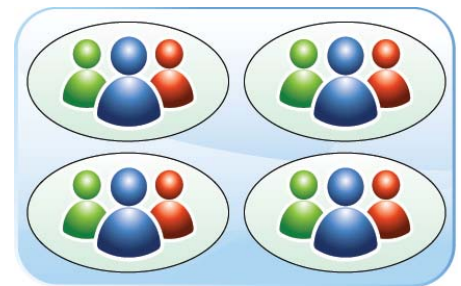
Unfortunately, these introductory educational strategies and activities tend to dissipate as organizations scale. As more and more teams transition to agile, cost-conscious managers attempt to “make do” with internal resources. Never having experienced an agile transformation on a broader scale, these early adopters often rationalize that the pilot project team members who have received special training and coaching will be able to roll out agile development to the rest of the organization. However, even agile experts, those with at least five years of agile experience, admit they continue to learn every day, so imagine the disadvantage of a small team with less than six months experience attempting to lead an agile scaling project.

In the end, there is no substitute for experienced agile coaching and consulting. Most teams scaling agile face similar challenges, so bringing in experts who have already successfully proliferated agile across enterprises can make all the difference in the speed and ultimate success of deployment. A relatively small upfront investment in planning, onsite coaching, and on-going mentoring can keep teams from spinning out of control and wasting time attempting to overcome challenges that many experts have faced on previous projects. In addition, whether we want to admit it or not, internal colleagues – even trained ones – often don’t carry the same weight as outside consultants who may have quite literally written the book on an agile methodology. When an industry authority offers advice, teams are significantly more likely to follow it because it’s objective and based on broad experience.

Instead of placing the burden of scaling agile on those internal managers and developers who have been part of a successful pilot project, a more potent recipe for continued success would be to select those team members who could make the transition to good teachers and pair them with external coaches. These internal coaches can become crucial agile champions who will carry the agile torch once the external consultants leave and can serve as the ultimate difference between long-term success and failure. Nurturing and promoting these motivated team members can help institutionalize both the practices and values of agile development for lasting change.

Mistake #2: Practice Agile In Silos

Because agile is often a grassroots-driven phenomenon, processes and practices sometimes evolve out of sync. Different teams within the same organization



try different methodologies, follow separate standards, or even implement specific practices in dissimilar ways. These teams often operate in silos, unaware of what others in the organization may be doing in terms of agile. As a result, they may have developed their own working methodology, with their own preferred practices and tracking and reporting systems. This works in small environments but can be disadvantageous as agile methods spread through the organization. Even small inconsistencies can lead to significant difficulties in integration, tracking, and productivity. They can also limit a company’s ability to build an adaptable organization that is able to leverage consistency training and education, project team and team member reassignment, and enterprise planning and reporting.

To combat this “silo syndrome,” a majority of companies that have succeeded in scaling agile have leveraged an internal service-oriented team dedicated to the consolidation, promotion and dissemination of agile knowledge and practices. Often emanating from early adopters and pilot team members, this reference team has some of the characteristics of prior process groups in terms of maintaining a body of knowledge, but philosophically operates in much more of a service role. Since every company tailors agile methodologies and processes to work within their specific business model and constraints, it falls to this core team to spread the word and act as a resource to new teams as they transition to agile development methods. With in-house access to their company’s unique brand of agile, customized to their specific needs and culture, new teams can obtain best practices, recommendations, and strategies from this readily available repository.

Having this reference team and centralized body of knowledge also provides a significant leverage point from which the entire company can learn and grow. Tools in this storehouse might include process and technology guidelines; planning, estimation, training, and reporting standards; and engineering practices, infrastructure, and tool recommendations. This type of consistency facilitates a common language for both teams and managers to communicate and collaborate. Organization-wide confidence in knowing that everyone is following the same general principles can be crucial to scaling agile successfully.

Mistake #3: Ignore Other Areas of the Business

A mistake many organizations make when scaling is to focus all of their efforts on their technology teams. Yes, agile is a software development methodology but successful scaling of agile methodologies is also generally accompanied by a shift in corporate culture and infrastructure for the entire company. Processes,



policies, titles, reward systems, and career paths that were geared more towards individual contributors may no longer make sense in the team-focused environment of agile methodologies. External stakeholders – for example, business unit leaders, customers, and executives – are often more consistently integrated with teams in agile development processes. An agile manager needs to be more of a leader and facilitator and less of an autocrat. Senior architects will still write code but may also carry out manual testing. Entry-level testers can get involved with software design. A manager's focus, then, shifts from allocating human resources to the right task for the appropriate amount of time to ensuring that teams are cohesive and collaborative. Similarly, cross-departmental coordination will likely need to be extended and optimized to facilitate more robust communication and ongoing collaboration.

Scaling agile development methods throughout your organization may also mean changing your compensation structure and/or HR policies to match the way you are doing business. For example, in a traditional waterfall environment, project managers are often incented simply to get the product to market on time. In an agile environment, the entire team is accountable for their results and needs to be incented for delivering the most valuable product possible in the given timeframe. On an agile team, "the most valuable product possible" includes not only the delivery date but also product quality, customer satisfaction, and the ability to adapt to changing priorities whenever necessary. Thus, when people are responsible for a variety of success factors and succeed or fail as a team, there are going to be challenges to traditional performance metrics.

Aligning a reward system that is more consistent with an agile approach will take some effort but is extremely valuable for successfully scaling agile.

Since the individual is fundamentally subordinate to the team in an agile workplace, there could also be an impact to traditional superstars as well as the well-known software stereotypes of cowboy coders and lone wolves. Teams produce results when individual members work well together, but close collaboration with team members may be a new concept to many. Managers need to pay close attention to the cultural changes that are part of moving to agile software development and allow time for them to become the new standard. This is a shift that will need to happen not just on agile teams, but across teams, departments - and even in the executive offices.

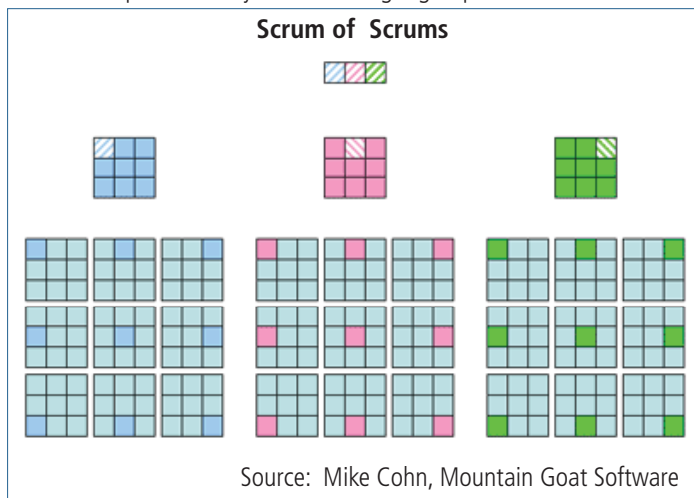
Agile is a brave and rewarding new world. Before long, the changes it brings will affect multiple areas of the business. Be prepared to both facilitate and allow the time necessary for it to infiltrate every level of the organization.

Mistake #4: Encourage Organizational Complexity

The natural inclination when scaling almost anything is to introduce management layers, new policies, additional processes, and other unnecessary checkpoints. Not surprisingly this kind of overhead and complexity typically has an inverse relationship to adoption speed. Simplicity is both one of the primary values of and critical success factors for scaling agile. Maintaining a commitment to simplicity when scaling agile processes and practices can drastically accelerate the internal rate of adoption. Adding needless complication is an invitation for failure.

With agile, companies relinquish bureaucratic control in order to achieve greater adaptability to changing market conditions and internal constraints, deeper visibility into the development process and ongoing project progress, and better long-term predictability of delivery dates and working software. In an effort to maintain consistency and control, it is natural to want to add excess management oversight and require unnecessary documentation. Don't do it. Or at least make every effort to minimize these tendencies. Agile software development consists of multiple small teams continuously delivering increments of working code or functionality in a short timeframe. As agile scales, each of those small teams will likely roll up to a larger product or project, and then roll up again to the company portfolio. It doesn't have to be any more complicated than that. Don't let the fact that more and more teams are implementing agile change its underlying premise: small teams consistently delivering working software. As mentioned above, ensuring general consistency in practices and metrics will often negate the need for supplemental rules and regulations.

A common strategy for overseeing large-scale agile shops comes from one of the more popular agile methodologies, Scrum. It is called the Scrum of Scrums and is a management approach that can be applied to virtually any agile approach. On a Scrum team, the Daily Scrum is a time-boxed meeting that happens at the same time each day so that team members can update one another on their progress and discuss any impediments. The Scrum of Scrums allows delegates from each Scrum team to come together and update one another on their team's progress and discuss any obstacles that might be overcome with help from the other teams or management. This same process can continue up the management chain as appropriate or necessary. The concept remains simple but easily scales for larger groups.



Another cautionary note regarding complexity is the dangerous lure of micromanagement. One of the benefits gained from moving to agile is unprecedented visibility into development team activities. The detailed information on tasks and hours can tempt management into trying to monitor individual team members' work. Management should resist that urge! Time spent investigating why one person is meeting his estimates and another is always going over his estimates may give managers the illusion of more control but typically does not yield better results.

Such interference runs counter to agile principles and can lead to trust issues within the organization. Instead, allow the teams to organize themselves around specific goals and plans. Let the managers closest to the teams guide them toward the goals that have been set. Don't confuse day-to-day project visibility with knowing the rationale for every decision the team makes. There is probably a good reason the team did not complete a feature on Tuesday as planned, and it is up to the team to manage their details in context of the project's overarching themes and objectives. Recognizing that teams are achieving positive results overall and understanding what teams can deliver on an incremental basis is a much better use of time. This insight will provide improved forecasting for future projects and allow much quicker response time to changing business demands.

Mistake #5: Fail to Scale the Infrastructure

The selection and implementation of tools and technologies, like any software development effort, should be driven by business requirements. Failing to consider infrastructure changes that will grow with the business and support future requirements can severely hamper any scaling effort.



With specific respect to agile development initiatives, whether IDEs, source control tools, collaboration tools, testing tools, or management tools are being investigated, teams should consider both their current and upcoming needs and make decisions accordingly. Several factors for consideration include:

- The volume of source code and teams requiring access
- The number of code branches that need to be maintained
- The locations of sites, remote teams and stakeholders that will be supported
- The number of teams, projects, clients, and/or product portfolios being managed
- The quantity of features, defects, tasks, tests, etc. being tracked
- The collaboration and reporting needs of the organization - from executives, to managers, to developers, to testers, and so on

As an example, a whiteboard and a simple source control tool may work fine for one co-located team, but multiple teams working on many projects may require a more sophisticated infrastructure. As the company scales, having teams use too many different tools can create unnecessary business inefficiencies. This is particularly true when knowledge must be transferred among team members, across teams to other departments or business units, and then rolled up to executives and outside stakeholders.

The test for infrastructure and tools in this context quickly becomes how well they scale. The source code management tool, test tool, wiki, and Excel spreadsheet a single agile team uses may work extremely well in a localized context, but the problem domain expands exponentially when scaling across multiple teams, projects, distributed locations, and hundreds of team members. In other words, the infrastructure needed to support larger agile organizations typically has more sophisticated business and performance requirements. Finding a sophisticated management tool that is as lightweight as the agile process it is supporting, however, can be a challenge.

Paradoxically, any management tool has to be powerful enough to aggregate and orchestrate hundreds or thousands of artifacts (i.e. stories, defects, tasks, tests, issues, customer requests, goals, etc.) as they flow through the product delivery lifecycle. At the same time, all stakeholders – executives, product managers, project managers, users, developers, testers, writers, etc. – should be helped by the tool rather than bogged down by its complexity. In addition, the tool needs to provide real-time visibility into a big-picture view of all projects under development. Implementing tools that work well within the existing infrastructure and can be accessed worldwide is critical to broad-based success. As tools and technologies are chosen for the scaling organization, a rule of thumb should be to match the tool to the growing business requirements and select one that will support the organization throughout the entire agile lifecycle.

Authors' Biographies

Robert Holler

Robert Holler is president and CEO of VersionOne, recognized by Agile practitioners as the leader in Agile project management tools. Mr. Holler has more than twenty years of enterprise software experience, and for more than a decade, Mr. Holler has been VP of Development, CTO, and CEO of several leading-edge enterprise software firms.

Mr. Holler's career in the software arena encompasses working as a consultant, developer, tester, technical writer, DBA, development manager and senior executive with organizations such as Andersen Consulting, Ockham Technologies, Tango Networks, and Clarus Corporation. Through these positions and experiences, Mr. Holler became familiar with Agile methodologies and iterative development processes early on, and it was his enthusiasm for Agile software development that led to the founding of Atlanta-headquartered VersionOne in 2002.

A frequent speaker at industry-related conferences and corporate functions on the implementation and benefits of Agile methodologies, Mr. Holler's writings on the subject have been published in both traditional and Agile-specific publications. Besides extensive experience in the agile and software development arenas, Mr. Holler also holds a Master of Science in management from the Georgia Institute of Technology and a Bachelor of Science in electrical engineering from Auburn University.

Conclusion

After proving the value of agile through a successful pilot project, the next step for many organizations is scaling agile throughout all development teams. While there is no silver bullet for quickly and easily spreading agile adoption, there are some common mistakes that can stand in the way. Given the broad process, engineering, and management impact that agile can have on an organization, the agile roll-out needs to be viewed holistically. Without significant consideration given to appropriate training, cross-department communication, modified management practices, process consistency, organizational simplicity, and adequate infrastructure, the spread of agile in your organization can be slowed, if not prevented.

Start small, keep it simple, and choose processes and tools that can help you scale smart to meet your changing needs.

Ian Culling

As Chief Technology Officer of VersionOne, Ian has over 20 years of broad information technology expertise including responsibility for software product management, design and development as well as enterprise information technology. He has significant practical experience with the introduction, scaling and adaptive execution of agile methods, having initially implemented strict Extreme Programming (XP) with a single team in 2000. Since that time, Mr. Culling has progressed to lead and coach both small and large organizations in their transition to agile methods.

Prior to joining VersionOne, Mr. Culling led the adoption of agile methods at Alogent Corporation as VP, Development, leveraging aspects of Scrum for scaling across multiple teams and products. This, combined with select XP developer practices and approach for planning and tracking, resulted in a Scrum-wrapped XP implementation, now a fairly common model within the agile community.

About VersionOne

VersionOne is recognized by Agile practitioners as the leader in Agile project management tools. Since 2002, we have helped more than 10,000 teams and 70,000 users in 50 countries from companies such as Adobe, BBC, Siemens, Disney, Dow Chemical, IBM, Lockheed Martin, Sony, 3M and Business Objects provide greater value to their customers by simplifying the process of planning and tracking Agile software projects.

To learn more about how VersionOne can help simplify and streamline your Agile software projects, visit www.versionone.com.