Scale agile throughout the enterprise
A PwC point of view

December 2013
Overview

Today it’s rare to speak with a company that is not adopting some form of agile development practice. However, many organizations are either in the early stages of agile adoption, or they are struggling to scale agile throughout the enterprise—far more difficult than adopting it at the team level. Despite these challenges, agile has been making big strides since it evolved in the mid-1990s.

Agile initially emerged in reaction to waterfall methods—characterized by heavy regulation and regimentation and an overly incremental approach. Developers often complain about waterfall’s sequential design process, where progress is viewed as flowing steadily downwards, like a waterfall, through various phases of software development. Other drawbacks include minimal interaction between stakeholders and the implementation team during application development, and a lack of utilization of team resources throughout the lifecycle of the product. In short, waterfall’s rigid approach, tedious processes and heavy costs sparked the need for a better development approach.

Many think that agile—with its iterative and incremental approach that enables teams to quickly adapt to change and reprioritize functionality throughout the process—is the solution. With agile, frequent deployment of code allows users to see the end product as it’s developed, and self-directed teams are better able to estimate, commit, and observe timelines. Fully functional software is delivered more quickly, while high levels of collaboration between the business and development increases quality.

While the benefits of agile are compelling, many businesses struggle to become truly agile—particularly when trying to scale agile throughout the enterprise. Agile adoption is easy to start, often taking root as a pilot initiative by a single team. Successful pilot programs, in turn, can foster expectations that agile practices can be easily replicated across the business. Yet, many organizations struggle to achieve the benefits of agile when deploying products that require enterprise-level planning and collaboration.

PwC’s Architecture Driven Agile (ADA) framework is one approach that allows organizations to leverage leading practices while achieving enterprise standards and performance levels.
As teams work to scale agile, challenges and uncertainty increase and require complex navigation during each phase. First, software depends on more than just the development team. Other groups like application and portfolio architects, technical support staff, shared service testing teams and program management offices (PMOs) also need to be engaged. Additional challenges in scaling agile include a lack of integration between IT and the business; varying agile interpretations and tools across different departments and product portfolios; and ineffective planning and estimation processes, which can draw concern from the C-suite.

Successfully scaling agile is an evolutionary process, requiring balance across many different aspects of the organization. It demands a tailored approach that leverages core agile approaches while breaking from traditional agile elements in areas like planning, architecture and governance. Importantly, no one size fits all. Effective agile implementations require organizations to identify and adapt the agile framework that meets the specific needs of the business. Organizations that approach agile strategically and map methods to their business objectives can reap significant rewards.

**Agile SDLC vs. Traditional Waterfall SDLC**

Typical phases involved in a Systems Development Life Cycle (SDLC)

**Waterfall:** Sequential development process where all required activities in the preceding phases is complete

**Agile:** Software development method based on iterative and incremental development encouraging rapid and flexible responses to change

<table>
<thead>
<tr>
<th>Planning</th>
<th>Design</th>
<th>Development</th>
<th>Testing</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Define scope and boundaries for development</td>
<td>Articulate scope into detailed requirements required to deliver functionality</td>
<td>Translate requirements into code</td>
<td>Validate that developed system addresses requirements</td>
<td>Deploy to production environment</td>
</tr>
</tbody>
</table>

Source: PwC
One approach that we leverage to help organizations scale agile is PwC’s Architecture Driven Agile (ADA) framework, which allows organizations to leverage leading practices while achieving enterprise standards and performance levels. While ADA is by no means the only agile framework that we leverage and tailor for clients, ADA is particularly well-suited for environments where a strong focus on architecture can be beneficial. ADA leverages core agile essentials across people, process and tools while breaking from traditional agile in three critical areas: architecture, release planning, and governance.

It’s helpful to look more closely at how ADA leverages agile essentials while also breaking from traditional agile elements. ADA leverages the following essential agile elements:

- **People:** ADA embraces skilled, multi-disciplinary teams; engaged, committed resources; focused, self-organizing teams; clear, simple definition of roles; and team accountability.

- **Process:** ADA fosters direct communication and collaboration, including use of scrums; close business stakeholder collaboration with development teams; story-driven development; and short iterations and frequent software demos.

- **Tools:** Essential agile tools that ADA leverages include automated functional test frameworks; test-driven development and automated testing; continuous integration; pair programming (or peer review) and a sustainable pace; spike solutions; and collaborative workspace.

Meanwhile, ADA breaks from traditional agile in the following areas:

- **Architecture:** Traditional agile does not consider architecture as a key part of the process, but assumes that architecture guidance is being provided in the background. Through ADA, we help organizations use architecture to establish a strategy and approach to link business goals and capabilities, and identify necessary changes to business processes and technology systems. ADA also embraces the following steps:
  - Rather than developer-driven design decisions, create a strong architecture blueprint and foster architect collaboration to drive the development sign-off process.
  - Develop a common architecture layer to understand interaction points, minimize duplication, and ensure optimal sharing of assets.
- Ensure consistent architect roles across the organization and lifecycle stages to ensure an understanding of scope and responsibility.
- Provide end-to-end oversight to avoid duplication that often occurs across groups.
- Define expectations for architecture and understand how it scales as teams become larger and interactions become more complex.

**Planning:** Standard agile teams plan in isolation and, while teams are coached to coordinate releases, there are no mechanisms beyond the scrum of scrums to formalize communication. Organizations can develop a release planning and management process where capabilities and stories are constantly fed by both business analysts and architects, and continually evaluated and prioritized based on regular team feedback. Additional steps in the planning phase include the following:
- Initiate constant regular business-IT steering committee reviews to help make critical program decisions (e.g. trade-offs in architecture and capability delivery, re-prioritization of scope/schedule, etc.).
- Create visibility and effective management of enterprise dependencies through a coordinated and standard planning cycle.
- Develop a release plan to keep elements like functionality, scheduling, and budget on track to provide a structure for story-based estimation.

**Governance:** Self-directing agile teams are very efficient and have remarkable ownership over deliverables, but often the seams between teams can be seen in the finished software. As agile scales throughout the enterprise, it’s vital to adapt governance processes to eliminate these seams. Organizations need to align the agile organizational structure with the broader enterprise. They need to help teams remain agile with the ability to coordinate enterprise systems, QA and shared services. Additional governance actions while scaling agile with ADA include:
- Establish a flexible governance strategy with a target state in mind. Similar projects should have similar governance processes across the enterprise as well as consistent management reporting.
- Set objectives that are actionable and that link to business objectives. Ensure that objectives can be understood and executed by delivery teams.
- Prioritize initiatives that foster collaboration. Pick one or two pilot initiatives that require heavy collaboration. This helps teams to exercise collaboration with a high level of visibility, which helps drive the right behaviors.

In addition to architecture, planning and governance changes, ADA requires team structure modifications.

**Team Structure:** Organizations should establish regular business-IT reviews to make critical program decisions (e.g. trade-offs in architecture and capability delivery, re-prioritization of scope/schedule, etc.). Following are additional steps that organizations can take with teams:
- Establish dedicated cross-functional delivery teams (“pods”) that include architects, business and quality analysts, developers, user interface designers, and project managers.
- Invest in highly skilled technical resources and bring in outside expertise to fill skill gaps.
- Physically co-locate teams in the same room with business resources sitting next to developers.
- Hold daily scrum meetings to immediately identify and resolve issues and risks.
In addition to ADA, a number of different agile methodologies have emerged, e.g. Extreme Programming, Scrum, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development and Pragmatic Programming. PwC is well-versed in numerous agile frameworks, working with clients to identify the right agile practices and adapt them to the specific needs of each company. Each organization requires a unique approach, e.g. ADA may be best for an architecture-centric environment; the Scaled Agile Framework (SAFe) may be appropriate if there is a need for portfolio planning; or XP might be the most suitable starting point if there is a need for a strong focus on engineering practices. Regardless of the chosen framework, it is worth repeating that adaptation is crucial for success.

Some of the more widely used agile methods include the below:

- **Scrum** is an agile development method, which concentrates on how to manage tasks within a team-based development environment. Scrum is the most popular and widely adopted agile method as it is relatively simple to implement and addresses many of the management issues that challenge IT development teams.

- **XP** (Extreme Programming) is a more radical agile methodology, focusing more on the software engineering process and addressing the analysis, development and test phases with novel approaches that make a substantial difference to the quality of the end product.

- **Dynamic System Development Method (DSDM)** is probably the most complete agile methodology, whereas Scrum and XP are easier to implement and complementary as they tackle different aspects of software development projects and are both founded on very similar concepts.

- **Scaled Agile Framework (SAFe)** is another framework that uses leading practices from Scrum, XP, and lean to implement agile practices at enterprise scale. This framework is applicable for more complex enterprise scaling initiatives as it enables an improved ability to plan, align, and execute distributed initiatives.
While agile methods vary, they all align with the philosophy that “in the new economy, to move aggressively into the era of e-business, e-commerce, and multi-device, companies have to rid themselves of their Dilbert manifestations of make-work and arcane policies.”

Whether you are just starting your journey with agile or you are already engaging with agile at the enterprise level, PwC can help. Our deep capabilities with agile frameworks including ADA, SAFe, XP and others, combined with our business domain, technical and industry-specific experience, enable us to work with you to develop and implement a holistic agile approach customized to the specific needs of your business. With the right investment of strategic planning and experienced execution, scaling agile throughout the enterprise can successfully extend the benefits of local agile on a much bigger scale — and help drive significant competitive advantage.
For additional information

Contacts

**Michael Mariani**
Principal
Technology Consulting
(602) 364-8486
michael.j.mariani@us.pwc.com

**Tim Mattix**
Principal
Technology Consulting
(312) 320-4400
timothy.mattix@us.pwc.com

**Curt Jacobsen**
Director
Technology Consulting
(213) 304-8670
curt.s.jacobsen@us.pwc.com

Additional PwC contributors
Walter Sun
Steven Legnine
Mario Gouvea
James Rehfeld
Raymond Hearrell

Visit
http://www.pwc.com/us/en/increasing-it-effectiveness/
applications-strategy.jhtml