



Agile Development Calls for an Agile Suite Solution

Authored by:

David A. Kelly and Heather Ashton

Upside Research, Inc.
www.upsideresearch.com

Contents

❑ **Executive Summary**

Agile development has been a revolution in software development for many organizations. Management is looking to find a way to achieve a high-level view of all the various development activities within the organization, while driving higher throughput on a lower budget.

❑ **Common Pitfalls of Agile Software Development**

If not well-managed, the potential benefits of Agile can disappear when the teams become larger and geographically dispersed, replaced by headaches and challenges that are common to traditional application development processes.

❑ **Optimizing Agile Software Development**

The solution to this challenge of scaling Agile development across a larger organization is to use professional software development tools to optimize the Agile process.

❑ **The Business Case for Professional Agile Suites**

There are a number of areas where organizations can see an immediate and significant impact to their development efforts when using an Agile Software Development Management.

❑ **The Agile Suites Advantage**

In the end, Agile is still software development. While it is a stark contrast from traditional methodologies, Agile software development requires some of the same attributes of traditional development to make it successful.

Copyright 2010, Upside Research, Inc. All rights reserved. No part of this report may be reproduced or stored in a retrieval system or transmitted in any form or by any means, without prior written permission.

Disclosure statement: This report was sponsored by, and may be licensed to, a limited number of software companies, including ones referenced in this report.

No quotes or information can be excerpted or used without explicit written permission from Upside Research, Inc.

Executive Summary

Agile software development is making inroads into organizations as a beneficial software development approach. What began as a revolution in 2001 has now become a regular part of many engineering organizations' development projects. The benefits are attractive: increase overall throughput of the development organization by up to 40%, and work better, smarter, and faster in a world of shrinking budgets. For most companies, gaining those types of benefits and successfully implementing Agile development processes requires more than just retraining people and moving to Agile methodologies.

The fast pace of Agile development doesn't scale across the organization without the right type of software tools in place to support your teams. From requirements through design, through building, through deployment to release, Agile-focused automated lifecycle suites can help increase both the return on investment of the Agile development processes and the success of Agile efforts. Just as a successful enterprise wouldn't consider building traditional software solutions without proper version control, release management, and testing capabilities in place, organizations struggle when pursuing Agile without appropriate tooling.

The rapid cycle times of Agile development are not a good fit for the traditional lifecycle management tools. Early adopters often applied them to Agile projects, but these force fit-type solutions can be error prone and required lots of manual intervention, bleeding off resources as the projects grew. In addition, they often don't provide the type of high-level, Agile-oriented view that's critical for good management of highly iterative Agile projects, and for monitoring adoption and improvement across a large development team.

This Upside Research report highlights how senior developers and management can gain a higher level of control and productivity out of the entire software engineering organization by optimizing Agile software development through the use of professional Agile Suites. The report identifies several leading vendors of professional Agile software development suites, and also builds a business case for using an Agile Suite to further improve efficiency and deliver better end results for all software development.

THE AGILE REVOLUTION

Agile development has been a revolution in the software world. Since its arrival in 2001, Agile software development has focused on the rapid delivery of quality software that is aligned with customer needs and company goals. Using continual development approaches and shortened cycles, Agile seeks to compress traditional software development practices by focusing on the team interaction and iterative processes. By following an approach similar to “Lean Manufacturing,” Agile techniques reduce the scope of projects, time-box the delivery date of this limited set of requirements to a few weeks, and gain quick feedback from the customer. This cycle repeats on the same shortened development cadence and is highly responsive to customer needs. The result is often faster time-to-market, high quality code, a greater match with business requirements and a leaner, faster, software development cycle.

As software engineering and development organizations face the reality of shrinking budgets and need to do more with less over the past few years, the promise of Agile development has become increasingly appealing. It’s rarely possible to add headcount to projects as requirements increase, putting a greater burden on improving the development process to increase productivity. Agile development processes, such as Scrum, have been highly effective in achieving increased productivity, often at lower cost. Studies have shown that Agile development can increase overall throughput of a development organization by 10 to 40% percent¹.

COMMON PITFALLS OF AGILE SOFTWARE DEVELOPMENT

While Agile can be a powerful tool for keeping up with rapidly changing business and software release cycles, increasing the relevance of software for an organization, and doing more with less, it does have limitations.

Agile’s scaling mechanism is to start with relatively small teams, and build a hierarchy of teams. For example, the sweet spot for successful Agile software development might be a relatively small team of perhaps 10 developers plus a similar number of QA people doing work in close proximity. This enables the team to work closely, achieve rapid development and make fast builds. Three or four of these teams work in parallel, with a “team of teams” made up of management representatives from each team. This is a challenging mechanism, is sensitive to the skill sets of the representatives from the product teams, and discourages interdependence.

In most cases, the small teams do not need a substantial development toolset to get started with Agile. Successfully managing the “team of teams” requires more information and control than ad-hoc methods can provide, and that drives development organizations of,

¹ According to a QSM Associates, Inc. report titled “The Agile Impact Report,” average speed-to-market increased 37% and team productivity increased by 16% using Agile processes.

say, fifty developers or more, to more significant tooling to ensure process compliance, high quality, and project visibility.

Without improved control of the process and software artifacts, visibility and communication (which are critical parts of the Agile process) can break down as organizations try to scale their Agile development efforts. The results can mean less productivity, rather than more. It can be difficult to manage multiple Agile efforts, especially when there is no visibility into the status of each project, and it can be impossible to maximize development resources and investments. The potential benefits of Agile disappear, replaced by headaches and challenges that are common to traditional application development processes.

Even organizations that have had initial success with Agile using their existing development tools and now want to scale that success to other development teams across the organization are likely to run into problems. Without the proper tools for managing the dependencies across the entire development realm, it becomes very difficult to repeat successes with large-scale Agile efforts.

Another consideration is that often not every team within the organization is going to move to Agile development, or not at the same time. Many organizations will have a mixture of Agile and traditional development efforts and tools in place. This places additional stress on traditional tooling, requiring significant investment of management time and developer scripting to maintain two different development processes to provide a high-level view for management.

Having a high-level, consolidated view of inherently local Agile development efforts is critical for widespread successful Agile adoption.

OPTIMIZING AGILE SOFTWARE DEVELOPMENT

The solution to this challenge of scaling Agile development across a larger organization is to:

- Provide **continuous training** in the process to individuals in their various roles,
- Manage with an emphasis on the **process, not the people**, and

When to Consider Agile Tools

If you currently have fewer than ten Agile developers, who all are located in the same building or room, your current software development tools may be enough.

Consider professional Agile tools if any of the following apply:

- You are scaling up to more than a handful of Agile developers
- You have multiple Agile teams (of any size) working together or in parallel
- The teams are spread across more than one location
- Your current software development tools aren't working for your Agile projects

- Use **professional software development and management tools** to optimize the Agile process. Agile Lifecycle Management Suites optimized for Agile processes provide automation and management where it is needed most, and can manage, monitor and report on the degree of process adoption and success in the Agile development teams.

A comprehensive Agile Lifecycle Management Suite solution should contain at least the following three components:

- A tool for **project management** of the Agile projects that provides visibility across teams to the VP of Engineering or senior management. The ideal tool will provide information about current status to product owners and business sponsors, giving them up-to-date views of how a project is progressing. Leading Agile project management tools include Rally Software, Mingle, and Scrumworks.
- A **software configuration management** and collaboration tool that supports local and remote development teams seamlessly. This tool will enable the source code to be managed and tracked as change packages or Agile stories, and allow teams to work more closely together, and share their iterations more frequently. Leading Agile software configuration tools include Rational ClearCase, Rational ClearCase UCM, AccuRev, and Subversion.
- A tool that effectively **manages the frequent builds** in an Agile project. This is especially critical since Agile builds are much more frequent than traditional software development approaches and generally utilize continuous integration. Leading Agile build and deployment management tools include BuildForge, AnthillPro and Electric Cloud.

Each of these types of tools exists in traditional software development, but unless the tools are designed for Agile's unique format and needs, and the tools are linked together to provide management with a cohesive, top-level view of all Agile development activities, the solution will fall far short of effectively supporting Agile development and delivering the benefits hoped for.

THE BUSINESS CASE FOR PROFESSIONAL AGILE SUITES

Investing in a professional Agile Lifecycle Management Suite trades off capital investment versus the labor cost to do-it-yourself. In most cases, the cost of the upfront capital investment, typically less than 5% of the loaded cost of a developer per year, has a rapid payback from the benefits that Agile Suites can deliver. The real risk of failure to the project of adopting or scaling Agile is the people cost. Tooling has proven to be an effective risk mitigation strategy.

In fact, there are a number of areas where organizations can see an immediate and significant impact to their development efforts when using an Agile Suite instead of traditional lifecycle management tools. For example, depending on the existing environment for development, organizations will see a dramatic savings in hardware, needing much less hardware to manage a geographically distributed infrastructure, compared to traditional lifecycle management solutions. Many of these professional Agile Suites also require lower bandwidth than traditional tools.

Another area that supports the business case for professional Agile Suites is the increased developer productivity and throughput that result. Studies have shown the increase in productivity to be as high as 40%. Even a minimum 10% productivity improvement, for a developer with an annual salary of \$120,000, translates to a cost savings of \$12,000 per year (not including the cost of the Agile Suite). For a team of developers, the numbers are even more impressive.

A more controversial area of tangible cost savings in Agile adoption has been the decrease in the need for middle management and the ability of organizations to redeploy highly skilled middle managers to front-line project leadership roles. Agile has such a well-defined process structure that teams ("Scrums" in the most common variant of Agile) are relatively self-managing, greatly increasing the management branching factor from an industry norm of seven professionals per manager to twenty or more in an Agile environment. Agile teams within traditional IT functions are often their own "relationship manager" with their customer, reducing additional development overhead while delivering high customer satisfaction.

One of the most important, but more intangible benefits of using a professional Agile Suite is that it provides the development teams with the ability to focus on the right tasks and features of a product at the right time. Longer, more traditional development cycles often

A Quick Look at Three Agile Suites

AccuRev's AgileCycle

Built as an Agile-specific toolset, AgileCycle combines three best-of-breed solutions from Rally Software, AccuRev, and UrbanCode/AnthillPro, to manage Agile processes from design through release. This provides an Agile Suite solution based on a combination of proven software solutions designed for Agile environments.

Microsoft Team Foundation Server

This product from Microsoft is well-designed for Microsoft shops. It is closely integrated with Microsoft SQL Server and provides a familiar environment for Microsoft developers. Organizations that have decided to standardize exclusively on Microsoft projects should consider this solution.

IBM Rational Team Concert

This is a new suite of Agile-oriented products built around the Jazz framework. These tools offer the promise of integration, but are still relatively new to the marketplace. Organizations should evaluate individual components against best-of-breed alternatives and keep an eye on Team Concert as the product matures over the next few years.

fail to keep up with changing customer feature requests. With a properly managed Agile project, the team is able to respond to customer and prospect demands quickly and reorganize teams and development efforts to meet those needs.

THE AGILE SUITE ADVANTAGE

In the end, Agile is still software development. While it provides a stark contrast to traditional “waterfall” methodologies, Agile software development requires some of the same attributes of traditional development to make it successful, including a disciplined development team, strong people management skills, professional tools support, and consistent and effective software development strategy and tactics.

Using a professional Agile Suite enables teams to automate the software development process, ensure compliance, and boost productivity across the development organization. In many cases, investing in Agile lifecycle management tools is a cheaper alternative to adding expensive developers. In the end, an Agile Suite can help increase throughput, ensure better consistency and compliance, and ultimately deliver better products to the customer.

About Upside Research, Inc.

Upside Research is a research and consulting firm focused on helping clients put application development, Web services, business process management, integration, and enterprise infrastructure challenges in perspective. Upside Research helps organizations find practical ways to achieve their IT goals and profit from the diversity of a changing technology landscape.

Upside Research, Inc.

www.upsideresearch.com

info@upsideresearch.com