



Dan Forman

Rally Software Development

# Integrated Agile Lifecycle Management

October 2007

## Executive Summary

The power of agile development practices is no longer in question as many of the world's leading software-driven organizations are responding faster to ever-changing market requirements. The emerging challenge for software delivery teams comes in forming new channels of collaboration and aligning heterogeneous tooling environments.

A new generation of application lifecycle management solutions have emerged to synchronize the flow of information among agile teams through innovative delivery models, next generation Web services interfaces and open integration strategies.

This paper will explore the trends impacting software teams, how they are responding to change and a new approach to integrated agile lifecycle management.



# Table of Contents

<b>1. EVOLVING SOFTWARE DELIVERY</b> .....	<b>4</b>
1.1 Agile History and Adoption .....	4
1.2 Technical Trends Affecting Change in Software Teams .....	5
1.3 ALM 2.0 .....	6
<b>2. DELIVERING BUSINESS AGILITY</b> .....	<b>7</b>
2.1 Organizational Change .....	7
2.2 Synchronizing Processes and Teams .....	7
<b>3. CUSTOMER CASE STUDY</b> .....	<b>8</b>
Company and Industry : Large Telephony and IP Communications Provider .....	8
<b>4. CONCLUSION</b> .....	<b>9</b>



# List of Figures

*Figure 1 : The Economics of Agile Success* .....5

*Figure 2 : ' The Changing Face of Application Life-Cycle Management'* .....6

*Figure 3 : Rally Integrated Agile Lifecycle Management* .....8



# 1. Evolving Software Delivery

---

Widespread adoption of agile development practices has forced an evolution in the tooling environments that support agile teams. With cross-functional teams moving quickly to deliver a continuous flow of innovation, the walls that have defined organizational and technical boundaries are falling fast.

Traditional role-based approaches to tooling a software delivery organization are breaking under the pressure of agile teams. Why? Because agile forces a collapse of the phase-based waterfall approach to defining, building, testing and delivering software. Teams must integrate and collaborate in ways that weren't envisioned when many of today's tool suites were conceived. A range of open source and commercial tools have emerged to accommodate the specific needs of agile teams. From continuous integration of daily builds to project management, agile development mandates a new approach.

*"Today's ALM suites don't offer much support for ALM beyond what can be accomplished through brittle tool-to-tool integrations. But tomorrow's ALM platforms will do much better by providing common services to practitioner tools. These solutions will be easier to implement, maintain, and employ. And at the end of the day, they'll enable development organizations to build better software."*

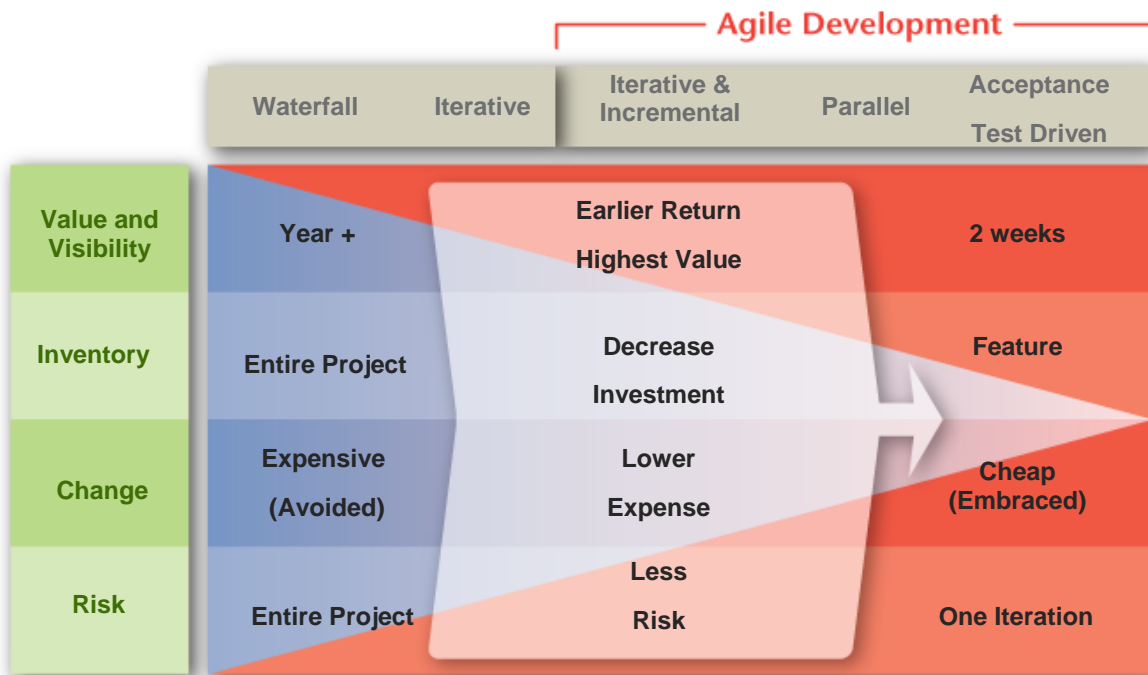
*Carey Schwaber, Forrester Research*

## 1.1 Agile History and Adoption

Agile practices borrow heavily from Lean manufacturing concepts that emphasize the need to mitigate waste and remove blockages. For software teams, this brings an intense focus to delivering fully-tested, functional increments of code that represent the highest customer value.

Most teams employ a mix of methodologies that include some elements of Scrum, XP, DSDM and many others. A few unifying themes emerge regardless of the mix :

1. *Just-in-time elaboration of requirements* : Eliminate the waste of lengthy up-front requirements definition for features that are likely to change and may never be implemented.
2. *Time-boxed development* : Iteration lengths range from 1 to 4 weeks. Teams hold this commitment sacred and deliver a demo to the entire organization at iteration close.
3. *Highest priority to acceptance* : Teams deliver the most meaningful features first and they are fully-tested.
4. *Acceptance tests inside the iteration* : Bringing quality forward eliminates last-minute surprises and ensures satisfaction of customer requirements



**Figure 1 : The Economics of Agile Success**

Forrester Research estimates that most Enterprise IT shops have adopted agile for some projects. Given the typical bottom-up adoption, agile projects can be difficult to track and almost always form in the most progressive teams first. As these teams face unknown technical challenges or loosely defined requirements, agile presents a flexible and effective solution.

### 1.2 Technical Trends Affecting Change in Software Teams

**SaaS :** Software as a Service (SaaS) has emerged as a compelling delivery model, with companies like salesforce.com paving the way on customer relationship management (CRM) applications. A range of benefits accrue to SaaS vendors : continuous feedback loops on customers satisfaction, quicker time-to-value by releasing more often and greater visibility into system health affecting all customers equally.

Gartner predicts that SaaS will grow seven times faster than on-premise software over the next three years and that by 2011, 25% of new business applications will be delivered as a service. IDC Research expects worldwide spending on SaaS to grow from \$4.2 billion in 2004 to \$10.7 billion by 2009.

The efficiencies of SaaS to the enterprise derive from the mitigated IT burden to administer and host software, end-user flexibility to customize applications and the multi-tenant architectures that serve thousands of users across multiple domains on a single piece of hardware.

**SOA :** Enterprise IT organizations are moving to adopt Service-Oriented Architecture (SOA) models that allow for reuse of standard service assets, better reporting across organizational boundaries and dynamic application building blocks for non-technical users. A 2006 Forrester Research survey of Government IT revealed that teams engaging SOA initiatives are 50% more likely to use agile

methods. With these trends and a move toward project-based IT initiatives, agile methods present a flexible approach to prioritizing, managing and delivering new functional requirements.

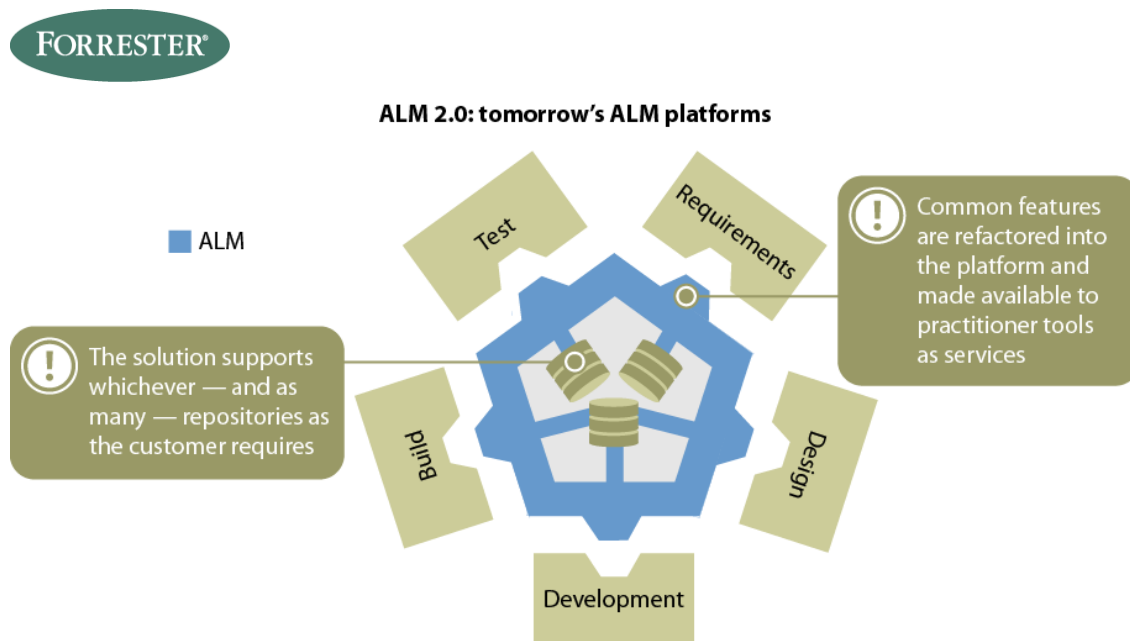
**Community Engagement** : Supporting a community in which customers provide a continuous flow of feedback can drive roadmaps that better address market requirements as they emerge. This democratic approach to product planning delivers a more responsive and relevant set of features and shuns the traditional document-centric methods of managing requirements. To enable this level of responsiveness and transparency, product organizations must rethink and re-tool existing processes.

**Next-generation Web Services** : Although Web services have supported data exchange among applications for years, new technologies like JSON are enabling computation in the browser. Just as Ajax has allowed Web applications to behave more like desktop applications, JSON, REST, SOAP and other technologies are enabling application mash-ups that allow users to leverage the information they need, when they need it and in the form they choose.

### 1.3 ALM 2.0

Application lifecycle management once implied a tightly integrated, often monolithic, stack of single-vendor tools to support functions ranging from development and build environments to testing and reporting.

Open standards for Web services integration, new delivery models and vendor collaboration are aligning to create new opportunity for software teams and the vendors that serve them.



**Figure 2 : “The Changing Face Of Application Life-Cycle Management”**



The characteristics of ALM 2.0 :

- Practitioner tools assembled out of plug-ins for right size, right price
- Common services available across practitioner tools
- Code repository neutrality
- Use of open integration standards
- Microprocesses and macroprocesses governed by externalized workflow

*Source : Forrester Research, August 2006, Trends "The Changing Face Of Application Life-Cycle Management"*

Whereas the early days of ALM focused on narrow, specialized tooling for each functional role, a new breed of ALM suites deliver a holistic and flexible approach to the needs of agile teams.

## 2. Delivering Business Agility

---

### 2.1 Organizational Change

With a move toward agile methods comes a need to rethink how the entire organization contributes to driving innovation. As development teams begins to deliver high-value features at an unprecedented pace, new challenges emerge : how will the product organization keep pace with a continuous flow of releases from the development organization ? ; How will you engage in a dialog with your customer to drive backlog priorities on a regular basis ? ; How will you bring greater transparency to how your organization responds to emerging market requirements ?

ALM solutions must enable composite views of traditionally siloed information repositories so that cross-functional stakeholders can respond to opportunity as it comes.

### 2.2 Synchronizing Processes and Teams

Techniques like build automation and continuous integration illuminate system-level quality with every new check-in. Taking action on this information requires a new approach to application development infrastructure that radiates real-time information to all relevant stakeholders as organizational boundaries and roles blur.

"With the accelerated pace of Agile development, keeping an eye on system quality is critical. Rally's approach to integrations has enabled us to synchronize teams and accelerate product releases while preserving existing workflow and investments in tools like Mercury Quality Center."

*Matt Brown, Director of development and quality assurance, Thomson Healthcare*

### 3. Customer Case Study

**Company and Industry :** Large Telephony and IP Communications Provider

**Development Environment :** 2000+ developers, globally distributed across 5 development centers with mixed tooling environments across business units.

**The Challenge :** Deliver on a top-down mandate to move the entire organization toward agile practices; build new channels of collaboration across time-zones and organizational boundaries ; and orchestrate new tooling requirements while preserving existing investments and workflow.

**The Solution :** Rally's on-demand Agile Lifecycle Management solution and open integration architecture.



Figure 3 : Rally Integrated Agile Lifecycle Management



With multiple agile teams rolling out concurrently, each with its own incumbent tooling solutions, a flexible integration strategy was the principal functional requirement for agile project management tooling. While the initial teams came to the table with more than 20 legacy tools, three emerged as key initial requirements :

*HP Quality Center* : Many of the teams rely on Quality Center for test case management and test execution. Test results must be synchronized with the new agile project management solution so that all stakeholders may react in real-time to changes in system quality.

*IBM Rational ClearQuest* : Quality management teams use ClearQuest to manage defects and track issues. When defects are created or resolved in Rally or ClearQuest, the other tool should reflect this update.

*Eclipse IDE* : Objective status visibility relies on granular daily updates from the entire development organization. Enabling developers to manage tasks, defects, test cases and notifications in the context of the IDE drives adoption and fine-grained reporting at the program level.

**The Results** : Unprecedented time to value. Before traditional ALM implementation and integration efforts would have even begun, Rally's secure on-demand delivery model, robust web services API and integration connector suite delivered a solution that aligned distributed teams while preserving existing tooling environments and workflows.

Rally provides integration with more than 15 tools to synchronize agile teams, leverage legacy tooling investments and bring needed freedom to ALM decisions.

## 4. Conclusion

---

Extensibility and automation rule in agile environments. What has traditionally been a manual process of orchestrating applications and subjective status reports must evolve to an integrated, best-of-breed approach that supports the co-existence needs of ad-hoc and established teams.

A new generation of tooling solutions are delivering on the promise of agile development through open interfaces, new delivery models and progressive integration strategies that move beyond the rip-and-replace legacy of ALM. Rally Software is leading the charge to provide software-driven organizations the tooling solutions needed to end the cycle of overspending on poorly integrated applications that limit information flow and process synchronization in agile teams.

With the freedom to start fast and scale easily, agile teams can innovate faster with today's Integrated Agile Lifecycle Management solutions.



---

### **About Rally Software:**

From an initial pilot project to enterprise rollout, Rally helps companies successfully adopt and scale Agile software development practices. Rally's family of Agile life cycle management products, which received the software industry's top honor of back-to-back Jolt Product Excellence awards in 2006 and 2007, map an incremental road to Agile adoption. The company's Agile University and world-renowned coaching services help mentor teams to create internal Agile experts, and its Web 2.0 community Agile Commons encourages knowledge sharing among Agile practitioners. Over 50 percent of the world's leading software-driven companies currently use Rally, and Rally's tools have supported its customers in more than 6,500 Agile releases. For more information, visit [www.rallydev.com](http://www.rallydev.com).

Dan Forman is Director of Partners & Alliances at Rally Software. Contact Dan at : [dan@rallydev.com](mailto:dan@rallydev.com) or +1 303.565.2862

---