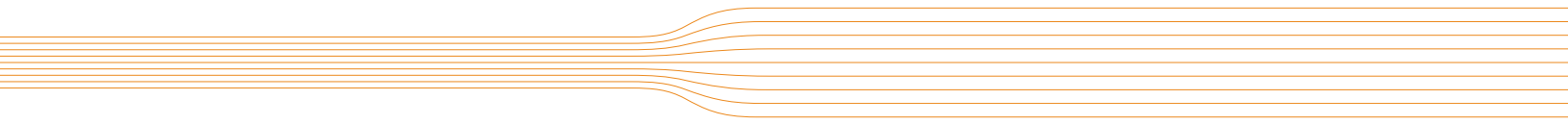
ALM IN GEOGRAPHICALLY DISTRIBUTED DEVELOPMENT ENVIRONMENTS

June 2007



SERENA™

TABLE OF CONTENTS

Executive summary

A formal Application Lifecycle Management (ALM) and change management infrastructure significantly increases the chances of project success for geographically distributed software development teams. Without a comprehensive and robust process for managing software development, such teams and their projects are likely to fail, negating any cost savings the organization had hoped to gain from its offshoring, outsourcing, or multishoring initiatives.

This paper describes common reasons for application development project failure and the challenges associated with distributing development among dispersed teams. It then describes how ALM can help these teams to produce applications that fully meet the needs of users—on time and within budget.

The development landscape

Today, a customer's first experience with an enterprise more often occurs through a website or other custom application than through a face-to-face or telephone meeting. Forrester Research has found that organizations that invest in innovative and engaging application development are getting more business, keeping their customers satisfied, and improving their efficiency.[1]

To meet organizations' growing demand for innovative applications, IT departments are increasing spending. According to Forrester Research, enterprise IT groups in 2007 expect to spend 25 percent of their entire budget on new software development projects, while overall spending on new projects is set to rise from 25 percent in 2006 to 33 percent in 2007.[2]

Because the pace of business is accelerating, organizations require those innovative applications now—and they can't afford poor quality or high cost. The need to do more, faster, and for less has fueled outsourcing. But today's teams are geographically dispersed, amplifying the traditional difficulties of managing software development across organizational, technical, and functional silos. Only 19 percent of firms with distributed development environments report that it's "easy" to manage the process.[3] The other 81 percent struggle to coordinate projects across different development methodologies, work processes, and time zones.

According to research and consulting firm E-Business Strategies,[4] outsourcing projects encounter trouble for a number of reasons ranging from poor vendor choice to neglect of personnel-related issues. Many projects fail due to management issues, including:

- Failure to establish stringent requirements and service-level commitments,
- Inability to maintain a periodic review cycle, including milestones and checkpoints, that would provide insight into the progress of each project
- Poor cross-group collaboration and communication.

Some organizations attempt to prevent these problems by keeping development in-house, but moving operations to a less expensive location. These "captive" development centers fare no better than their outsourced counterparts. Their processes have often been tailored to work in the parent's single-site environment, and don't translate well to the offshore captive center.[5] As a result, captive centers are often used ineffectively and inefficiently.

Without effective collaboration and communication, outsourcing and offshoring will not deliver their projected ROI. In a recent informal poll taken at ALM Expo, more than 50 percent of attendees stated they worked with teams that were more than six time zones away.[6] At this distance, teams can't communicate effectively in real time.

[1] Forrester Research, "The Business Case for Rich Internet Applications," February, 2007.

[2] Forrester Research, "The State of Application Development in Enterprises and SMBs," February, 2007.

[3] Forrester Research, "The Challenges of Software Change Management in Today's Siloed IT Organizations," November, 2006.

[4] http://www.ebstrategy.com/outsourcing/projects/topten.htm

[5] Forrester Research, "Shattering the Offshore Captive Center Myth," April, 2007.

[6] ALM Expo 2007 "Practical ALM 2.0" Q&A polling question, May 2007.

Success, failure, and ALM

Despite these challenges, an increasing percentage of software development projects conclude successfully, according to *SD Times*' recap of The Standish Group's 2006 Chaos Report.[7] Whereas in 1994 only 16.2 percent of software development projects could be classified as "successful" (meaning they were completed on time, within budget, and met user requirements), by 2006 that figure had more than doubled to 35 percent. Meanwhile, failure rates declined from 31.1 percent in 1994 to 19 percent in 2006. This is encouraging news. But when only one-third of projects are successful, there's still room for improvement.

While more companies are experiencing absolute success and fewer are experiencing absolute failure than a decade ago, the percentage of companies encountering *some* degree of success or failure has remained relatively stable: 52.7 percent in 1994, falling slightly to 46 percent in 2006. That means that most of the companies surveyed are still achieving suboptimal results.

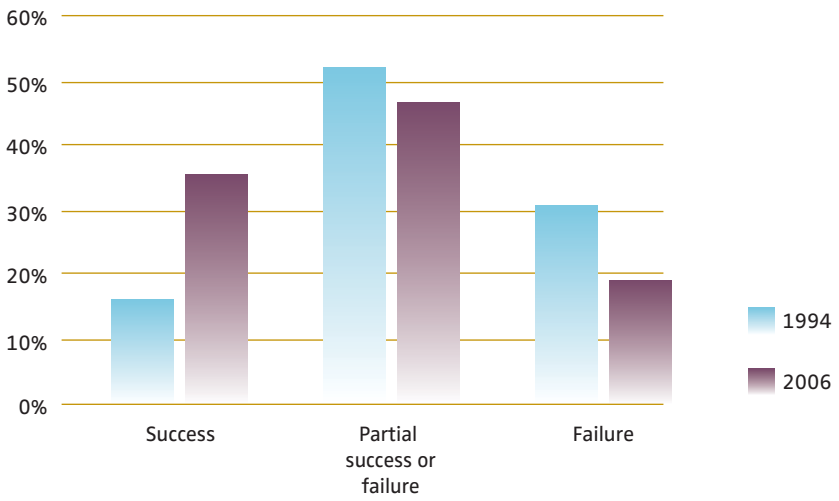


Figure 1. While software project success rates have risen and failure rates have decreased over the past 12 years, the most common result is only partial success

The Butler Group's Michael Azoff notes a change in development methods and tools that coincides with this increase in project success: "This period [between 1992 and 2006] has seen a number of major changes in software development: open-source software projects; the Agile development movement; and advances in tooling, notably Application Lifecycle Management (ALM) tools."[8] As ALM tools appear to have been a factor in increasing project success rates, continued advances in ALM hold promise for improving success rates still further.

[7] David Rubenstein, "Study: Less Chaos in Development Shops," *SD Times*, March 1, 2007.

[8] Michael Azoff, "Application Lifecycle Management Making a Difference," Enterprise Networks and Servers, February 23, 2007.

How ALM works

ALM encompasses the entire development lifecycle, integrating a set of disciplines (such as requirements management and simulation, design, version control, build, and release management) with processes, policies, and procedures to provide reporting and traceability across the lifecycle. This helps teams manage change across job roles and development processes, which in turn can help projects come in on time, within budget, and to the satisfaction of users.

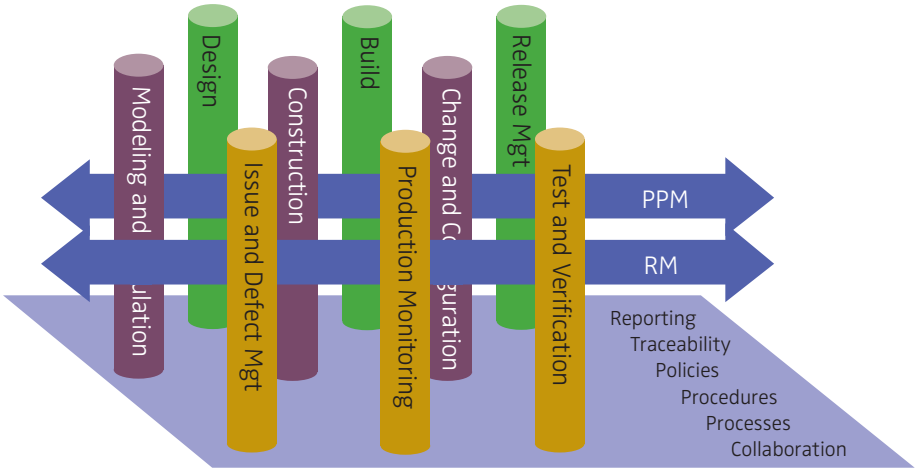


Figure 2. The ALM disciplines are brought together with processes, policies, and procedures to provide reporting and traceability across the lifecycle

ALM helps teams synchronize and automate the relationships and dependencies among development disciplines—including requirements, modeling, development, build, and testing, all the way through to deployment—across global sites, teams, platforms, and methodologies.

Most early ALM tools were poorly integrated point solutions. Vendors sometimes cobbled on additional features to address more stages of the application lifecycle, but with limited success. Now, a new generation of ALM tools is evolving to overcome these shortcomings to deliver both true integration across disciplines and better support for quality management processes.[9] These new applications are especially suited to overcoming the challenges faced by development organizations in distributed environments.

[9] Forrester Research, "The Changing Face of Application Lifecycle Management," August, 2006.

The challenges of development in distributed environments

Organizations need a common frame of reference that bridges technical, organizational, functional, or geographic silos. Cross-silo communication has always been a challenge for software development teams, which have struggled to overcome miscommunication among parties involved in different disciplines or phases of the project. Geographic distribution adds another dimension to this difficulty.[10] Poor telecommunications, cultural differences, accents, and varying degrees of language ability can all hamper communication between distributed teams.[11]

A study by Forrester Research, commissioned by Serena, shows that the majority of enterprises—58 percent—use software developed or maintained by firms in different locations. In addition, many businesses confine work on each application to a single location.

According to the study,

> ''For many of these shops, geographic distribution is a fait accompli no matter how difficult it might prove. Many firms report that the cost savings from labor arbitrage are so compelling that they cannot possibly be outweighed by increased cost in areas like software change management. And in other cases, the knowledge of applications and their histories that happen to reside at far-flung sites is so invaluable that it's worthwhile to maintain those sites for that reason alone.''[12]

To mitigate geographical challenges, some organizations package outsourced work into very small tasks and converse frequently with offshore teams to monitor progress. While this approach may increase the likelihood of success, it requires constant vigilance from the onshore team. By one reckoning, offshore projects demand 20-30 percent of managers' time.[13]

The simple fact is that a Boston-based team running an Agile development project will likely encounter issues when coordinating and working with a Bangalore shop using a traditional waterfall methodology. Basic communications tools will not suffice; these teams need tools that connect all phases of application development in addition to bridging communication gaps.[14] This is why groups like Forrester believe that ''firms need to implement a sophisticated workflow and new collaboration tools to govern globally dispersed projects,'' including investing in change management systems.[15] Organizations with geographically distributed development require an ALM framework that can automate, synchronize, and track projects, taking into account every associated job function, component, and methodology.

[10] Joseph Feiman, ''Concepts and Tools That Enable Globally Distributed Application Development,'' October 6, 2005.

[11] Brian Nicholson and Erran Carmel, ''Offshore Software Sourcing by Small Firms: An Analysis of Risk, Trust and Control,'' IS Perspectives & Challenges in the Context of Globalization, 2003.

[12] Forrester Research, ''The Challenges of Software Change Management in Today's Siloed IT Organizations,'' November, 2006.

[13] Sudin Apte with John C. McCarthy and Ronald J. Furstoss, ''Offshore Product Development Has Arrived,'' August 3, 2006.

[14] Feiman.

[15] Apte.

Next-generation ALM works in distributed environments

Many ALM tools address isolated disciplines, resulting in fragile and expensive integrations that never quite meet the needs of the customer. For example, a developer might have to use a requirements management tool, an issue/defect tracking tool, and a version control tool in the course of building a software solution. Because each of these tools has its own data, the organization can't benefit from traceability and reporting across disciplines.

To date, no single tool addresses each aspect of ALM. However, a new generation of ALM tools takes a process-centric view of software development, hiding discipline-specific components and offering users a single, unified interface through which they can perform all the activities of their roles.

For geographically distributed teams, the right ALM tool will:

- Accommodate multiple development methodologies simultaneously
- Offer a common repository for traceability
- Tightly couple visualization with requirements management
- Facilitate communication about change, workflow, and function.

SERENA DIMENSIONS 10

Serena Dimensions 10 is a scalable ALM solution that automates the full development lifecycle and enables visibility and traceability across all its stages. Uniquely suited to distributed development environments, Dimensions 10 combines visualization, requirements, and configuration management tools with build, deploy, and reporting capabilities. It also integrates requirements with configuration management in a single repository, a key to delivering quality software.

Serena Dimensions RM for Requirements Management enables control of the requirements management process with flexible and enforceable standard definitions and comprehensive impact analysis. It delivers visibility and traceability throughout the application lifecycle with a simple and familiar browser-based interface.

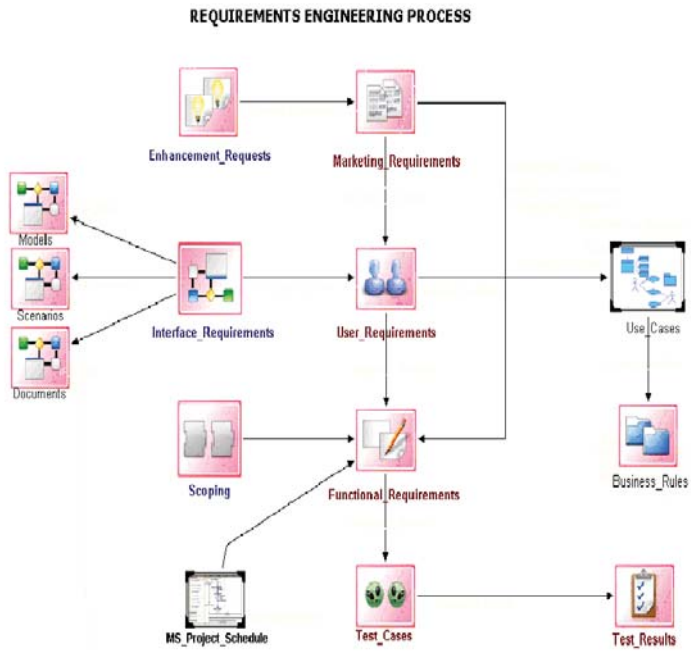


Figure 3. Serena Dimensions RM helps organizations define and standardize their requirements management processes

Serena Dimensions CM is a comprehensive software change and configuration management tool. In addition to industry-leading configuration management capabilities, it also offers integrated task management for repeatability, predictability, and accountability.

Dimensions CM incorporates a sophisticated dashboard that provides visibility into development projects and tasks. The dashboard includes both high-level visibility and drill-down capabilities to show more project-oriented data.

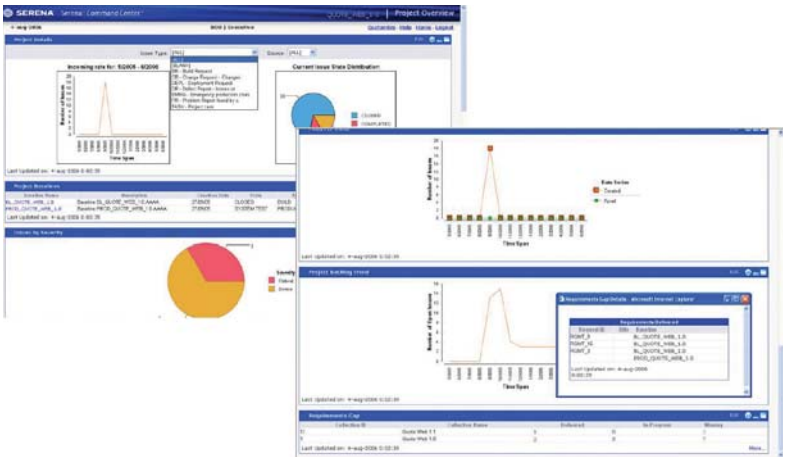


Figure 4. Serena Dimensions CM provides visibility across the application lifecycle

SERENA TEAMTRACK

Geographically distributed projects can benefit enormously from tools that help streamline, synchronize, and automate processes across teams. Serena TeamTrack enables effective cross-team coordination for more disciplined, predictable, and efficient development by synchronizing activities and providing greater visibility into project status, risks, and trends.

Together, Serena Dimensions and Serena TeamTrack support teams developing business-critical software, helping to control change from project inception to completion.

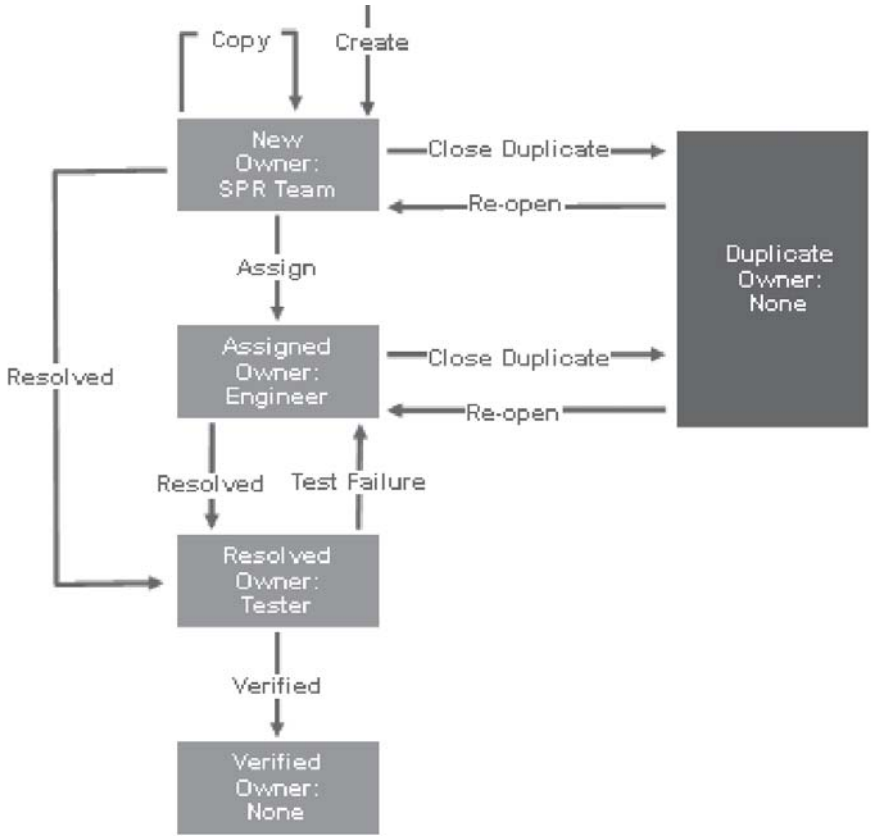


Figure 5. TeamTrack helps globally distributed teams communicate and collaborate effectively, so no task falls through the cracks

Conclusion

Over the past decade, ALM has helped organizations to double their software project success rates and reduce failure rates. Although ALM cannot guarantee success, lack of a solid ALM infrastructure can contribute to project failure.

Geographically distributed development projects are at a particularly high risk for failure. Their communications challenges compound the difficulties of governance and management across ALM disciplines. ALM tools that help to automate and synchronize processes in mixed development environments using mixed methodologies can help teams to overcome the challenges of working with strangers in different time zones.

Only rigorous and robust management of every step of the application development lifecycle will enable organizations to overcome the challenges of application development with geographically distributed teams. If companies are to realize the cost savings potential of outsourced development—and end up with high-quality applications delivered on schedule—they must implement comprehensive ALM tools and processes.

ABOUT SERENA

Serena is the leader in Application Lifecycle Management for distributed and mainframe systems. More than 15,000 organizations around the world, including 96 of the Fortune 100, rely on Serena software to automate the application development process and effectively manage their IT portfolios. For more information on Serena software and services, visit: www.serena.com

CONTACT

Learn more about ALM for the enterprise by visiting http://www.serena.com/US/solutions/alm-solutions/index.aspx or contacting one of our sales representatives in your area.

Serena Worldwide Headquarters
Serena Software, Inc.
Corporate Offices
2755 Campus Drive
Third Floor
San Mateo, California 94403-2538
United States

800.457.3736 T
650.522.6699 F
info@serena.com

Serena European Headquarters
Serena Software Europe Ltd.
Abbey View Everard Close
St. Albans
Hertfordshire AL1 2PS
United Kingdom

+44 (0)800.328.0243 T
+44 (0)1727.869.804 F
ukinfo@serena.com

Serena Asia Pacific Headquarters
360 Orchard Road
#12-10
International Building
Singapore 238869

+65 6834.9880 T
+65 6836.3119 F
apinfo@serena.com

SERENA™