



Managing Service Oriented Architectures with Application Lifecycle Management

An MKS White Paper
By David J. Martin
Vice President
Product Management

Introduction

Despite many improvements to software development methodologies, project planning and management tools over the past decade, enterprise IT organizations continue to be challenged in their ability to meet business goals and to deliver projects on time and on budget. According to the Standish Group the percentage of failed projects is still too high.

Standish reports that more than 30 percent of software projects are cancelled before completion. More than 50 percent of projects will come in at 200 percent of their original estimated costs.

So why is software project failure a continued problem within the IT organization?

The breadth and complexity of legacy applications is one common cause. Legacy IT applications frequently have a large and difficult to maintain code base, with entwined dependencies and poorly documented procedures accumulated over time. The larger and more complex the system, the greater the maintenance burden, and new development efforts create an ever-growing integration and testing cycle with each subsequent release. Development teams frequently underestimate the time and effort required to ensure that new functionality will not disrupt existing capabilities.

The siloed nature of technology teams is a second significant contributor to IT's inability to effectively deliver against their project goals. Within an IT department, there are many decision makers, many projects, many teams and many disciplines supposedly working toward the same business goal, yet operating in isolation or in some cases, to counter purpose. Departments, teams and individuals operate within their own closed processes and disparate decision-making frameworks. These inefficiencies stifle revenue growth, cost control, and innovation; and present a constant efficiency and productivity hurdle.

In its research report on IT silos in the banking industry entitled, *Banking Silos Create IT Governance Roadblocks*, Forrester Research says – “Within most large banks, each of the LOBs is responsible for business processes, even those that exist across business units, like account opening and cross-selling. Correspondingly, most banks don't invest deliberately in software applications that support processes that cross LOBs. This decentralization creates many versions of the same process in different applications, all of which have to be developed and maintained by IT.”¹

Clearly management needs to improve control over their departmental activities ranging from project planning, project execution, development and IT operations. Yet it is a daunting task to achieve this control and extract useful metrics if the data is scattered throughout dozens of diverse tools supported by disparate processes.

Service Oriented Architectures (SOA) is an approach to bridge these challenges by enabling IT organizations to be more adaptable to the business and to help them become more agile; responding to the need for faster response times, cost control and greater business flexibility.

- SOA is a set of enabling technologies (repositories, message brokers, software services).
- SOA's primary benefit is that of agility – enables development organizations to more rapidly assemble and deliver software as re-usable services to drive benefit to the business more rapidly and efficiently.
- Agility is manifested through re-use – services are developed to satisfy a business function, then incorporated into various applications via assembly-based tools (“Write one, use many times”).

- SOA requires not only new technologies, but a fundamental change of culture. IT organizations must move from a culture of 'build a better mousetrap' to one of 'make it easier to find the mousetrap'.

However, in order for SOA adoption to succeed, IT organizations must take into serious consideration the management implications of transforming to, and servicing of this platform. Adoption of specialized SOA tools has its challenges, and this in itself poses a significant risk to the overall success of the initiative, such as:

- How can we make it as easy as possible for our department to adopt this change without turning their world upside down?
- IT already has a glut of tools within its domain, and adding another set of tools specifically for SOA is simply compounding the management challenge and will quickly erode SOA's benefits.
- Have organizations investigated the true cost of adopting SOA-specific tools and how will they integrate them into their established practices?
- And once immersed in SOA, which configurable items need to be managed and how across their respective lifecycle?

This white paper discusses the key management challenges surrounding SOA adoption and how application lifecycle management (ALM) can provide a comprehensive management platform for SOA transformation, adoption and ongoing administration. Application lifecycle management is a necessary factor, minimizing disruption to development organizations and reducing the complexity of SOA deployments. This paper is not intended to be a general primer on SOA itself.

Why ALM is an Essential Ingredient for SOA Initiatives

'By 2009, more than 85 per cent of development platforms will be enabled for SODA-style development'

Change Management Battles Complexity: Linking Development and Deployment, Jim Duggan, Research VP, Gartner, May 2006

A critical aspect of SOA is the ability for software teams to easily re-use previously developed services as a way of achieving agility.

SOA of course, is a more broad sweeping initiative affecting an organization beyond its software development realm. That said, good SOA practices should be grounded in an application lifecycle management framework.

Let us turn to an example to illustrate the value ALM provides in an SOA context.

A new project is struck within a banking organization to deliver an online Mortgage Application and Renewal process for bank customers. The first stage of the project is to describe the Project Charter, defining the scope, requirements, resources and such. The next step is to plan the delivery phase of the project. It is at the stage when the requirements are defined, that teams could establish *how* the requirements will be met; either through a build-from-scratch approach, or through the use existing software services. This is where SOA's true benefits begin to manifest themselves.

ALM is a well established practice in most large enterprise IT organizations today. Its concepts and the various ALM stages are reasonably well understood, moving a development project from requirements

definition, through a set of distinct steps and staged toward system delivery. Software change and configuration management (SCCM) is also a fundamental capability used by software teams today. It is rare to find an IT organization or development shop that does not have an SCCM system (or at a minimum version control) in place to manage changes to systems.

Curiously in the many discussions and debates surrounding SOA today, ALM is a missing piece of the conversation. It is a somewhat rhetorical question, but when planning such a radical transformation to a new style of development such as SOA, why would an IT organization not consider leveraging existing ALM and SCM tools and processes currently in place?

SOA adoption is itself a challenging task requiring radical cultural change. To implement tool change as well is inviting mayhem. IT organizations should therefore actively seek to minimize disruption by turning to ALM as the management backbone from which to build out their SOA development processes.

'Key Issue: What challenges to the planning and control of change management are presented by shifts in architecture and development techniques?'

- Change Management Battles Complexity: Linking Development and Deployment, Jim Duggan, Research VP, Gartner, May 2006

ALM platforms today offer IT organizations a comprehensive set of capabilities and best practices for SOA adoption. Additionally, ALM implemented on an enterprise scale (i.e. one platform and standardized processes deployed across the entire organization) provides the following capabilities, which will be crucial to the successful transformation to an SOA-capable organization:

- Requirements definition and management
- Modeling support (how can the system be conceptualized and designed?).
- Software configuration management (development and versioning of system artifacts)
- Change management (manage changes throughout the lifecycle)
- Test management (manage the overall test plan to ensure software meets requirements)
- Full traceability over progression of system development activities from requirements, through to testing, then deployment

ALM and its associated development best practices have long been well established disciplines and they must be leveraged to their fullest extent in SOA transformation, no matter what style of development and management is currently in vogue. SOA concepts are not revolutionary – they have roots in earlier re-use initiatives such as CORBA. And many of the adoption challenges connected to these earlier attempts at re-use began with a lack of integrated development and management tools to facilitate such adoption.

Even today, some IT organizations are establishing separate SOA sub-organizations with their own toolsets and processes. This approach is wrought with challenges. The failure to incorporate existing teams along with ALM frameworks and best practices will result in even further fragmentation of tools and processes, further extending the siloed nature and complexity of IT organizations today. This segregation of SOA can potentially create a culture of technical “elitism”, resulting in staff morale issues that will, ironically, doom the initiative before benefits could ever be realized.

There are certainly a number of specialized SOA tools on the market today. Each offers a unique solution to the overall SOA management challenge. These tools include asset repositories, service brokers, and application servers, to name but a few. Without question, a number of these technologies are necessary for SOA adoption, however others — specifically asset repositories — feed the chaos, further inspiring a culture of tool fragmentation. Asset repositories contain service definitions and

metadata describing the purpose of a service, ownership over that service and other important information.

Asset repositories however, are completely disconnected from the versioning or software change and configuration platform used daily by software developers. This requires a developer to use one vendor's approach to query and locate a service, and yet another to engage in daily development activity. This is an expensive, complex and highly inefficient approach. Developers would need to use multiple tools to get common answers; and this approach would be potentially unreliable and risky, as the SCM and asset repositories would require constant synchronization.

This example simply illustrates the fragmentation and roadblocks at the tools level facing IT organizations today as they move toward SOA, and are potential jeopardies to early SOA success.

Incorporating an enterprise ALM platform as the management backbone of an enterprise SOA initiative can:

- Reduce further fragmentation of teams and tools
- Enable organizations to leverage their existing management best practices
- Create sustainability for the initiative due to "integration" into day-to-day processes and activities
- Enable teams to focus on SOA development itself rather than integrating disparate tools together
- More quickly realize SOA's ROI (Return on Investment)

Furthermore, an enterprise-specific ALM platform that is built on a single platform and tightly covers all aspects of the application lifecycle — such as MKS Integrity — has additional capabilities and benefits which are tuned specifically for SOA needs.

Moving from Theory into Practice — The Role of ALM in Supporting SOA Development and Deployment

Migrating a development organization to SOA from a tools, best practices and cultural perspective is one half of the challenge facing IT leaders today. Once in practice, SOA brings forward additional management challenges for an IT organization that again can be very effectively navigated by leveraging an application lifecycle management platform and ALM best practices.

Most IT managers will concur that it is challenging to manage the complex development lifecycle of a large enterprise system. Applications are touched by many hands over many years, impacted by many development tools, and absorb many changes that may or may not have sufficient documentation or just cause.

Now imagine a world where applications are constructed of many individual moving parts. Each component can potentially have multiple iterations and will likely be used as a building block for a number of systems. Let's expand that complexity to factor in the many artefacts related to a service — such as, but not limited to, unique requirements that may be levelled against a service or its iteration, the associated documentation of that service, test plans, approval cycles governing its use, and even incident or problems associated with a service once in production.

SOA offers the promise of many benefits — agility, efficiency, cost effectiveness — but the business risks of unmanaged SOA run equally large.

An application lifecycle management system needs to be the management platform for daily service-based development activity. Management of SOA goes far beyond identifying and retrieving a specific

service. IT leaders must be assured, particularly in this age of compliance, that business risk associated with service oriented development is minimized. An ALM system can not only identify and retrieve a component, offering similar functionality to an asset management system, it can move beyond into active management of that service (and its iterations) through its complete lifecycle. ALM systems enable development organizations to baseline services, share services and various iterations in a parallel manner across multiple development paths, assemble shared services into an application with complete audit ability and control, and automate and control the deployment of services into a production environment. Finally, a single ALM system governing every stage of the lifecycle and connected to a portfolio capability gives CIOs and their development managers the ability to truly monitor SOA initiatives against agility and efficiency goals. The virtue of having all services-related lifecycle data in a single repository allows managers to extract services development and operational metrics such as: where a services is used, how much effort goes toward development/maintenance of an individual service, incidents logged against service — all highly valuable information in SOA initiative performance monitoring.

MKS — One ALM Platform for SOA

MKS Integrity is the only solution on the market today capable of providing complete coverage across each stage of the application lifecycle through a single solution and architecture. MKS Integrity offers a breadth of lifecycle support from requirements management to deployment of change to production — with all activities driven and supported by continuous process.

It offers a number of market unique capabilities that are ideally suited for managing SOA initiatives; specifically:

Enterprise Support

- Multi-platform enterprise architecture (Windows, UNIX, Linux, iSeries and ZOS integration)
- Granular security (component, file, process)
- Role based permissions and approvals
- Enterprise authentication (LDAP, ADS, Windows, UNIX)
- Audit logging
- Enterprise scalability
- Collaborative platform (email, logged discussions, process automation and escalation)
- ALM processes integrated in one platform; unprecedented visibility

ALM Best Practices

- Versioning and SCM repository (lowest common denominator of development; used by every developer in the enterprise)
- Requirements Management
- Change Management
- Test Management
- Deployment Management
- Management Reporting and Dashboards
- Comprehensive metrics across lifecycle
- Portfolio Analysis and Management

SOA Specific Capabilities

- Component identification and versioning support (organize source code into reusable services)
- Review, approve and publish services into the repository
- Search the services repository to find required services
- Baseline (checkpoint) services (i.e. Create version 2.0 of Credit Checking Service)
- Service (project) sharing capabilities (use version 1.1 of Credit Checking Service)

- Assembly of shared services (incorporate Credit Checking, Forms Validation, Mortgage Submission, Mortgage Approval services into one “application” set)
- Automated and controlled deployment of services into production
- Ability to extract services development and operational metrics (where used, how much effort to develop/maintain, incidents logged against service, etc.)

Conclusion

With the continued challenge facing IT organizations in meeting their business objectives, SOA has emerged as a new way to enable them to become more agile in meeting these needs. While the technologies and practices hold much promise, it is imperative to embark on this initiative with caution, and to incorporate ALM best practices and capabilities to ensure its success.

Properly managed, an SOA initiative can lead to:

- More rapid delivery of business functionality to meet business objectives
- Reduced development and testing costs, which contribute to shorter project cycles
- Better software reliability and reduced maintenance costs resulting in enhanced customer and end-user satisfaction.

MKS’s enterprise ALM solution, MKS Integrity, offers an innovative, comprehensive and logical approach for managing SOA initiatives through its advanced platform, experience with large implementations and unrivaled field organization.

Related Reading

You will find the following whitepapers in the MKS Customer Community, on MKS’s website at <http://www.mks.com> or by request from your MKS Sales Representative:

[Software Re-use and Component-Based Development with MKS Integrity Suite™](#)

[Cataloguing and Re-use of Software Assets Using the MKS Integrity Solution](#)

[Scalable Process For Enterprise Software Development](#)

[SCM and Refactoring](#)

[Parallel Development Strategies](#)

[An Innovative Approach to Managing Software Requirements](#)

[Regulatory Compliance and IT – Converging Challenges](#)

Footnotes

1. January 5, 2006, Banking Silos Create IT Governance Roadblocks, by [Mary Pilecki](#) with [Bruce D. Temkin](#), Adele Sage, Forrester Research