

# Orchestrating Mainframe and Distributed SCCM and Release Management

Reference Code: CYIT0093

Publication Date: July 2011

Author: Tony Baer

---

---

## SUMMARY

### Catalyst

Over the past 20 - 25 years, most organizations have accumulated highly diversified application portfolios on a variety of platforms, with the mainframe remaining just as relevant as ever. In so doing, the application portfolio has grown far more complex; that in turn makes the decisions necessary for maintaining and growing it far more complicated. Applications originally built for standalone operation now have interdependencies with other applications and platforms and reach further into the enterprise and out to the partner and customer base. The result is that there are more business and operational considerations riding on decisions whether to change or replace applications and when and how to release them. In many sectors, this decision has regulatory compliance ramifications. The challenge is that, as systems have grown distributed, so have the decision making processes and tooling used for managing change. Enterprises require a consolidated view that gives them the full picture.

### Ovum view

The traditional silo'ed approach to making change management decisions has hit the wall. Change requests are not simply about deciding the merits of new version or code changes. Each change decision requires organizations to balance the need for change vs. the availability of resources, the

health of other projects that are currently live, the capabilities of different teams for delivering specific types of projects on time, regulatory implications of a change and the ability to update live production systems without causing disruption to the business. It requires an integrated approach that can help the organization make rational business decisions, while preserving an audit trail that is critical, especially for organizations that are subject to regulatory mandates.

## Key Messages

- Changes to applications are often made in isolation without regard for the full ramifications of change decisions.
- The practice of software change and configuration management faces growing stress thanks to structural changes in the competitive environment.
- Mainframe applications should be managed, not as an island, but as first class citizens of the enterprise applications portfolio
- With the mainframe alive and well, enterprises should provide a single focal point for managing change across all platforms.
- Serena's orchestration platform is an approach to managing the applications. Part of a new wave of integration solutions across ALM vendors, Serena provides a federated approach that can rationalize change decisions across platforms

## THE MORE THINGS CHANGE...

### Mainframes remain pivotal

Over the past 20 - 25 years, most organizations have accumulated highly diversified application portfolios as IT systems spread from the back office throughout the enterprise. In so doing, their platform environment has broadened and changed over the years, as distributed systems have emerged alongside to complement -- but not necessarily replace the mainframe. Long the workhorse of enterprise transaction processing, the mainframe has built a reputation as a highly reliable, trusted and scalable computing environment. Yet, workloads have changed as IT systems have spread from the back office to across the enterprise. With the consumerization of IT, employees are demanding the same anyplace, anytime computing capabilities that they are enjoying with their smart phones. As a result, enterprises are under pressure to extend their applications to new interaction scenarios and environments.

All of these changes have made IT environments more complex. Although mainframes were initially perceived as high-cost, rigid platforms; they are continuing to prove their worth as much of



the world's back office processing still executes there. Historically operated in standalone mode, mainframes are increasingly being integrated into the fabric of the rest of the IT environment. For instance, a mainframe accounting system will likely coexist alongside a customer relationship management (CRM) application that tracks all the touch points, and a business intelligence (BI) reporting system that provides analytics.

Legacy systems that survived the Y2K transition are likely to remain in place unless a major acquisition drives a reordering of enterprise systems. Even then, mainframes are likely to remain in the mix.

Most enterprises lack the appetite to rip and replace; instead the emphasis has shifted to rationalizing and maximizing the value of application assets that are in place. This is especially the case for organizations that must rationalize application portfolios in the wake of mergers and acquisitions.

In place of migration, the most popular modernization strategies focus on extending existing systems to support new use cases on new platforms such as smart mobile clients or providing process or rules-driven integrations with line of business systems on distributed platforms. Service-enablement has become an increasingly popular strategy for adding new relevance to the legacy application portfolio.

Software isn't the only thing being extended. The role of the mainframe platform has also morphed. Thanks to co-processors, they can also run Java, XML and Linux workloads that are consolidated from distributed machines.

## **Change and Configuration Management more essential than ever**

The discipline of change and configuration management is well-established in the software development world, although not always consistently practiced. Thanks to the concept of the "change package," changes are supposed to be judged in context. Besides software code, a change package should also reference the documentation, software and hardware interdependencies, design models and related artifacts. From this core body of data, it ought to provide the information for performing change impact analyses.

Change management has become a critical challenge as many organizations struggle to see the forest through the trees. For instance, some enterprises handle up to tens of thousands of change requests per month. For such organizations, it is essential to be able to readily differentiate requests by level of importance and driver -- such as whether a request is essential for enabling a major competitive thrust or for satisfying a compliance mandate. For organizations that manage



dozens of development teams spread across the globe, it is essential that they have visibility to team performance before committing another change to the backlog.

### **There are significant operational impacts**

The practice of software change and configuration management is under growing stress thanks to structural changes in the competitive environment. Mainframe applications are increasingly being called upon, not simply to coexist with applications on distributed platforms, but increasingly to interoperate with them. The business logic that resides in mainframe application portfolios is increasingly being repurposed. Thanks to service enabling, the logic is being exposed and orchestrated through business process and rules engines to dynamically execute in new environments, under new contexts and to new audiences.

Consequently, the impact of a change to any single application can have multiple, often unforeseen consequences. Up to 80% of unplanned downtime is caused by process or people issues -- such poorly documented or planned changes to software applications. Poorly planned changes to software can cause harmful interactions with other software applications or cause system failure as a consequence of mismatched configurations with hardware or OS images.

As legacy applications are service enabled, it becomes even more difficult to anticipate the full downstream impacts of even a single code change. Challenges are compounded as deployment environments are virtualized or run in the cloud.

Not only must change be carefully managed, but the process can no longer be operated in isolation from release management. Changes in software may be targeted for environments that are themselves in ongoing states of flux. This is especially true with the embrace of virtualization strategies and the growing use of cloud computing resources. Therefore, code changes impact not only adjacent applications, but also the underlying physical and virtual configurations of software and platforms that may run on or off premises.

### **Changes to software have business consequences**

SCCM has traditionally been considered the domain of the application development organization. Yet, in many organizations there are serious policy or compliance ramifications to software change. The economic costs of downtime resulting from poorly planned changes to software, IT infrastructure or deployment strategy has prompted many organizations to institute formal change management strategies that are managed by formal entities such as change advisory boards. Furthermore, in many sectors, such changes may have regulatory compliance ramifications, with a few examples including:



- A retail bank makes changes to an online banking system that integrates customer stock brokerage accounts. It becomes essential to validate that such changes do not open new paths for money laundering which could violate provisions of the U.S. Patriot Act and similar regulations.
- Property and casualty insurers modify underwriting applications adding links to credit reporting agencies that may be in violation of insurance laws in some states but not others.
- Healthcare providers modify access control logic to electronic patient record systems that may violate privacy provisions of the HIPPA act.

As businesses face the need to add new functionality or capability to handle unexpected load spikes, there is the need for policies, workflows and formalized processes that help IT make smarter change management decisions and release strategies.

Managing change is not simply a matter of approving new functionality, adding new infrastructure or adopting virtualization or cloud deployment strategies. It is also a matter of who makes the decisions; whether those decisions are made using documented, repeatable processes and whether changes are implemented with the right sequence of activities (such as the assurance that they are properly tested prior to final approval). In highly regulated sectors such as healthcare delivery or financial services, the players who are authorized to approve or implement change may be highly proscribed.

#### **Don't forget the release process**

Approving change is just half the battle. Once a software change is authorized, there is the matter of release to production. The release process itself requires a consistent sequence of events to ensure that IT infrastructure can accommodate the changes without causing unplanned downtime. Many organizations have policies regarding uptime, as it impacts the level of service that they provide to their customers, and in turn, the revenue stream itself.

Not surprisingly, given the business and operational ramifications of software change, many organizations have instituted Change Advisory Boards (CABs) that must formally approve change requests of any kind.



## **ORCHESTRATION EMPOWERS BUSINESS AGILITY**

### **The world is a fast changing place**

Few will argue that the competitive environment is changing more rapidly than ever. Innovations in technology, fluctuations in consumer taste or the sudden occurrence of natural disasters or geopolitical events can suddenly turn the competitive environment upside down. As the competitive environment changes, so does the business's requirements for software. Not surprisingly, agile processes have emerged that place a premium on the rapid delivery of new software capabilities, complemented by the embrace of continuous testing and integration practices for accelerating delivery times.

All of this places strain on traditional SCCM processes. Ironically, as the marketplace places demand on change, the need to effectively manage change is greater than ever, yet the comprehensiveness of existing processes can often slow down the delivery timetable.

### **Orchestration: Applying a business context to SCCM and release management**

Clearly, change requests are not simply a matter of making decisions about the merits of new version or code changes. Each change decision requires organizations to balance the need for change vs. resource availability; the health of other projects that are currently live; the capabilities of different teams for delivering specific types of projects on time; regulatory implications of a change; and the ability to update live production systems without causing disruption to the business.

Orchestration of the application lifecycle is a capability that Serena has developed with its new Serena Business Manager (SBM) product, which is designed to help IT organizations make smarter decisions throughout the lifecycle of developing and deploying applications. SBM is an example of the approaches to process integration that are emerging in the ALM market. It applies a federated approach that, using web and RESTful services, integrates with Serena and third party tools to provide broader and deeper visibility to IT managers, project managers or change advisory board members. More importantly, SBM populates data and views through the tools that team members are already using. That can include tying project and portfolio management views to quality management, requirements management, agile project planning, build management, IT service desk and related systems.

Using SBM, decision makers view the cascade of events that are triggered by any specific change request. They can review the queue of change requests, compare their relative importance, and



then examine the ability of different teams to assume the task. Once assigned, team leaders can drill down to monitor at the project level, whether it involve the development, test or release cycles. Orchestration can be used to support policies, rules or regulations that may demand observance of specific workflows that ensure consistent management of application delivery, while preserving critical decision-making audit trails.

### **Accountability is critical**

Preserving audit trails is obviously important for projects and/or industry segments that are subject to regulatory mandates. However, the preserving of audit trails carries benefits far beyond regulatory compliance, and can therefore prove useful for organizations or projects that are not under strict regulatory eyes. The competitive environment is constantly changing. Customer preferences change, new game-changing products or services reshape markets, and companies merge and acquire. Furthermore, for many IT organizations, there is generational change, as baby boomers retire, taking their systems and application knowledge with it.

Consequently, understanding, not only whether a change was made, but who made it and why, is becoming critical. Serena provides capability through its ChangeMan SSM offering, which tracks all changes to specified z/OS datasets. It also offers the capability to back out of changes when conditions necessitate.

## **CUSTOMER EVIDENCE -- LARGE US BANK HOLDING COMPANY**

Like many sectors of the economy, the retail banking sector has rapidly consolidated over the past 15 years. A large US bank holding company that evolved from a regional bank to one of the top 20 institutions in the US now manages assets that are well over \$150 billion. Like any organization that has undergone M&A, the rate of change requests snowballed as the organization faced the need to gradually consolidate multiple financial product account systems previously maintained by independent entities.

The organization already had a core of strong change management practices as a consequence of an enterprisewide Y2K remediation initiative in the late 1990s. With the initiative having heavily targeted the mainframe, the organization relied on Serena's ChangeMan ZMF tool as their change management platform.

As the organization got its Y2K efforts under control, it also found itself in the midst of a wave of acquisitions; that coincided with the embracing of distributed applications. Not surprisingly, given the wave of acquisition and the changing economy, distributed development was initially conducted on a wide range of platforms and SCCM tools including Telelogic Synergy (now part of



IBM Rational), Microsoft Team Foundation Server (TFS), and open source code repositories such as PVCS.

Furthermore, as the bank's customers embraced online and mobile banking, the need for maintaining 24 x 7 uptime became a reputational objective. With industry consolidation and customer embrace of self-service, maintaining uptime became a necessity for customer retention.

The result is that the bank recognized a need to rationalize its change management strategies, and extend them to the increasingly important area of release management. Building on its experience with ChangeMan ZMF, the bank adopted Dimensions CM for managing change with its distributed systems. To ensure a global view of change, the bank adopted SBM to provide a unified view of change requests, issue tracking and quality management through integration to HP Quality Manager. The bank uses SBM to enforce a workflow ensuring that all approved change requests undergo the necessary testing before changes are frozen.

The next step of the bank's change management project will extend orchestration to tie in one of the biggest sources of change requests: trouble tickets filed by the IT Service Desk. The organization has embraced ITIL in forming its incident management processes but, until now, the linkage between incident management and software change management has been informal. The goal in this case will be to empower the IT Service Desk so that problem resolution will grow far more seamlessly.

## **CHANGE CAN BE YOUR FRIEND**

Mainframe platforms are not only thriving, but are evolving from their standalone heritage to becoming first class citizens as part of broader, heterogeneous IT environments. In addition to venerable legacy applications, they are increasingly running mixed workloads comprised of applications written in modern languages designed for more frequent enhancement and update.

The broader role of the mainframe has in turn transformed the role and purpose of change management. Traditionally, it was implemented to demystify legacy code by mapping change. No longer can change management be implemented in isolation on a single platform. Reflecting the broader role of the mainframe as part of the enterprise application fabric, change management must evolve to an integrated process that is driven by policy and implemented using consistent workflows. From its heritage as a provider of change management solutions on the mainframe, Serena has advanced its approach forward with an approach that utilizes orchestration so that change management becomes a process by which organizations can more readily respond to changes in their competitive environment.