WHITEPAPER

Orchestrated Application Delivery

Deliver Apps Faster and at Lower Cost by Managing Application Demand, Development and Delivery as an End to End Business Process

By David Hurwitz

September 2010

SERENA™

or · ches · trate [awr-kuh-streyt]

Definitions

- To arrange or manipulate, esp. by means of clever or thorough planning or maneuvering
- Organize; cause to happen

Synonyms: arrange, blend, compose, coordinate, harmonize, integrate, manage, present, put together, set up, synthesize, unify

Speed and Efficiency in an App Driven World

We live in an app driven world. iPhones are hardly the only environment where apps make the difference. Banks service customers on-line and via teller machines more than through human tellers. Airlines sell tickets, check-in customers and promote frequent flier loyalty through apps. Governments provide transparency and constituent service through Internet apps. And those are just some business-to-consumer examples. The business-to-business economy has long since been app driven: everything from supply chain management to securities and commodity trading to…well there's an app behind most every B2B interaction.

Orchestrating the end-to-end delivery of apps from demand through to deployment is more competitively important than is IT operations.

With so many apps already in production, it would be tempting to say that app driven innovation is mature, with little room for additional competitive advantage. That would be wrong, as the evolving mix of devices, Internet communications and markets creates a constantly shifting playing field from which new winners and losers regularly emerge.

Speed and efficiency are the keys to winning in this world. For instance, a recent Forrester report states "In the current economic conditions, efficiency is the name of the game — everyone needs to squeeze out just a little more."[1] Thus, enterprises that develop and deliver apps faster and more efficiently than their competitors are guaranteed a competitive advantage, no matter how entrenched their competition may be.

Indeed, orchestrating the end-to-end delivery of apps from demand through to deployment is more competitively important than is IT operations. This is because data centers are being virtualized and then out-sourced to the cloud, turning run-time environments into commodity services capable of yielding little or no differentiation.

Orchestrating apps-that-matter – "killer apps" in the vernacular – to run in these commoditized data centers is now the game. Do that better than the competition and the benefits will pile up.

The App Dev Budget Squeeze

IT budgets for innovation are not growing, in no small part because maintenance of existing apps is growing too much. In fact, maintaining the current app portfolio threatens to eat up more resources every year. For instance, a typical financial services organization easily has between 5 million and 10 million lines of code in 10,000 and more separate programs.[2]

The maintenance for these apps can easily dominate the portfolio, and the overall app delivery budget. The result is that there is insufficient money, people or time to work on anything new.

The use of offshore and outsourced resources are two popular methods to address this budget squeeze. However, these techniques are often insufficient and can be counterproductive absent proper process orchestration.

Global App Delivery

Globalized demand for app-enabled business processes is increasingly paired up with a globalized supply of business analysis teams, system design teams, software engineering teams, user interface development teams, testing teams, and operational environments. Indeed, Forrester reports that "24% of IT executives expect to increase spending on outsourced application development and maintenance in 2010."[3] The good news is that this highly distributed development and delivery arrangement leverages the geographic availability of reduced cost, specialized skills, and corporate resources.

The bad news is that this same structure is prone to local optimization, poor collaboration, and natural inefficiencies. For instance, teams often select their own tools, especially as open source products like Subversion allow them to fly under the budget radar. More significantly, teams optimize within themselves, narrowing their communications to outsiders in the process. This leads to local efficiencies, as each team tries to become more efficient at what they produce, but global sub-optimization, as many teams miss the big picture of what the organization needs to achieve, for instance being taken by surprise at the essential need for auditability and traceability. Then when reality intrudes, perhaps in the form of an audit, they are forced to scramble.

The result is both top-line and bottom-line costs. The latter come from hidden costs that pile up between groups and in the operational opacity that senior leadership must penetrate. The former comes from the organization's inability to innovate quickly and predictably, thus leaving money on the table for faster moving competitors to scoop up.

"Siloed paper-based practices for the management of each stage of development and operations are evolving to more automated interconnected processes."

- Jim Duggan, Gartner

Process Orchestration vs. Centralized Repositories

An unappealing alternative has long loomed for executives charged with orchestrating app delivery processes. Centralized vs. decentralized application lifecycle systems each have virtues, though both often suffer from a lack of process orchestration.

Centralized Application Lifecycle Management (ALM) systems typically revolve around a centralized repository, into which all artifacts must go. While this supports a one-source-of-truth goal, it elevates artifact management above orchestrated process management in the design of the overall app development and delivery system.

Artifact management remains vital, especially at the requirements, change, build and release management levels. However, it is best distributed, especially given the rise of grassroots repositories like Subversion and the ability of modern gold-vault systems to control the path to production.

More important is to provide a process orchestration system into which various point solutions can integrate. Then various groups can independently decide whether to stay with their legacy systems and/or upgrade to more modern or capable alternatives. Meanwhile they can participate in the enterprise-wide app orchestration process, making the whole operate as greater than the sum of its parts.

D2D: The Process Behind App Orchestration

Killer apps don't originate within and then emerge magically from development. Indeed innovation properly emerges from development's close and iterative engagement with the business. However, a development organization that develops great apps by successfully and regularly collaborating with the business can still fail if doesn't get releases into production with confident precision.

As Gartner says, "As teams grow larger or have to deal with larger volumes of software, siloed paper-based practices for the management of each stage of development and operations are evolving to more automated interconnected processes."[4]

End-to-end orchestration must traverse Demand through to Development and on into Deployment. This process is known as Demand-to-Deploy or D2D for short. This process can be thought of in similar fashion to Quote-to-Cash, Procure-to-Pay and other well-known business processes.

D2D encompasses the three phases of the full application lifecycle: Demand Orchestration, Development Orchestration, and Deployment Orchestration. While these process phases should ultimately be linked for maximum value, they must always be iterative and close-looped. That is, there must be feedback mechanisms between Development and Demand (i.e., from analysts and developers back to users), between Development and Deployment, and between Deployment and Demand.

This last is important for two reasons. Users need to be notified when features or fixes they are awaiting go into production. And every release triggers additional demand for enhancements and fixes.

Modern app orchestration products should both improve a specific function and be process-enabled to “snap into” adjacent functions as they get modernized.

Auditability Required

Regulated enterprises must pass regular audits and process checks, with more coming. For instance, a recent Forrester report states that “Regulation in financial services and healthcare will increase as a result of the lingering effects of the recession and politically driven reforms.”[5]

Given that failure is not an option with these audits, the cost in staff time to prepare and participate in them can be immense, and often undocumented. That is, development and operations staff are often asked to manually create audit trails and to assist in presenting them to auditors, work that may not be accounted for in resource planning and that takes away from the core mission of app dev and delivery.

Addressing this elevated cost-of-compliance requires building auditability into the app orchestration system itself. That way, audit trails and traceability are byproducts of normal process execution, whether from requirements-to-feature, or build-to-release, or completely end-to-end.

Developers may not always like engaging with a process management system, but they always hate having to manually document their work. The ideal app orchestration system solves both problems: Normal work management creates an audit trail while not asking analysts, engineers, or operations people to go out of their way during or after undertaking activity.

Start Anywhere

Every enterprise-class app dev and delivery organization owns a rich stew of tools, systems, tote boards, and spreadsheets. It comes with the territory, as does a profusion of platforms: Linux, Windows, mainframe. To add to the cacophony, most enterprises utilize a mix of agile and traditional methodologies, with many incorporating lean techniques for process improvement.

So where to start orchestrating the overall process? Wherever it hurts most. Modern app orchestration products should both improve a specific function (e.g., requirements management or change management) and be process-enabled to “snap into” adjacent functions as they get modernized. Thus it may make sense to start with an aspect of demand orchestration, knowing that it can feed development orchestration down the line. Or start with mainframe change management or release automation or any of the dozen other app orchestration processes and functions.

Fortified with ROI from initial orchestration initiatives, additional initiatives become easier to justify and implement. Plus the end-to-end traceability and auditability becomes more valuable the farther out the end points of the orchestration engine reach.

Integrated Understanding

Are we getting better, faster and more efficient? Where are our bottlenecks? What dependencies are elevating our risks? Are our costs tracking to plan? Will we release apps on time?

Visibility into these and other KPIs are essential for an organization to improve. Individual ALM products generate partial answers, but given the integrated D2D nature of app delivery, only the ultimate linkage of supporting systems into a complete process orchestration can provide the breadth and depth of answers that CIOs require.

Thus the need for D2D process orchestration isn't just for workflow purposes. It is also for business intelligence purposes.

Conclusion

Better apps marry up with better processes to enable competitive advantage. IT organizations can be positioned to uphold their end of the equation by improving their ability to orchestrate the delivery of apps from demand through to deployment. This might seem daunting after wrestling with a generation of disconnected and then overly centralized tools, but it needn't be given the recent rise of orchestrated ALM.

Orchestrated ALM marries best-of-breed ALM systems with a process orchestration engine to enable enterprise reach, phased implementation and distributed usage. It builds on the desire of individual teams to use team-oriented open source tools while providing auditable and enterprise-class systems where they are necessary, e.g., controlling the path to production.

The better apps needed for orchestrated app delivery come from modern Orchestrated ALM suites. Together with solid app delivery processes (e.g., agile and lean), they can deliver the efficiency needed to address the app dev budget squeeze, while also powering the delivery of killer apps for the business.

Orchestrate app delivery this way for a year or two, and the competition will wonder what hit them.

Endnotes

[1] Jennifer Belissent, Ph.D., Forrester, 2009: "The State Of Global Enterprise IT Budgets: 2009 To 2010 Business Data Services Global"

[2] Scott Nelson, Gartner, 2010: "Signs Indicate a Train Wreck Is Coming, Unless You Modernize IT"

[3] Liz Herbert, Forrester, 2010: "Systems Integration Spend Rebounds In 2010, Sourcing Execs Must Drive Successful Vendor Selection, Governance, And Pricing"

[4] Jim Duggan, Gartner, 2010: "Key Issues for Application Life Cycle Management"

[5] Paul Hamerman, Forrester, 2009: "Trends 2010: Financial Processes And Applications, Economic Prospects Appear Brighter, But More Regulatory Oversight Looms"