DevOps for a New Millennium: A Lifecycle Perspective Supporting Business Growth in an Altered Economy

Report Summary

An ENTERPRISE MANAGEMENT ASSOCIATES® (EMA™) Research Report

Quarter 2, 2013

Platinum sponsor: IBM®

Silver sponsors: CollabNet® OpTier

EMA™ IT & DATA MANAGEMENT RESEARCH, INDUSTRY ANALYSIS & CONSULTING

DevOps for a New Millennium: A Lifecycle Perspective Supporting Business Growth in an Altered Economy (Report Summary)

Table of Contents

DevOps for a New Millennium: A Lifecycle Perspective Supporting Business Growth in an Altered Economy (Report Summary)

Overview

Cloud. Agile. Virtualization. Big Data. Mobile. Emerging from the economic wasteland that characterized the first decade of the 2000s, the IT Industry is charging out of the gate with an explosion of unprecedented innovation.

During the depths of the "great recession" – the years between 2008 and 2012 – established technology vendors were hanging on for dear life to weather stalling revenues and rock bottom capitalization based on diminishing stock prices. At the same time, "new" technologies, such as Cloud-delivered Application Performance Management (APM), virtualization/private Cloud, and mobile were seeing double- and even triple-digit year on year revenue growth.

Hard economic times drove a new pragmatism. Previously content with recouping the costs of enterprise management investments within two to three years, customers demanded payback within six to twelve months. Limited budget dollars were spent on investments with near-immediate Return on Investment (ROI). Cloud, wireless, and virtualization grew accordingly, as businesses became more adept at using technology to reduce costs.

Not coincidentally, Agile development and "DevOps" became hot topics. Often perceived as being inextricably linked, though this isn't necessarily the case, both satisfied two key requirements that surfaced during the recession – budget maximization and the "need for speed." Industries and regulatory requirements were changing so quickly that software Development and Operations teams were hard pressed to keep up.

The growth of Agile development helped to answer these challenges, but forced both Development and IT Operations to become more governed and collaborative. From the Development perspective, massive software changes had to be delivered with increasing frequency and at higher levels of quality. This, in turn, drove Operations to become far more proficient at rolling software changes into production.

The "DevOps" term is used to describe the process of managing the handoffs necessary for Development and Operations teams to work in a collaborative manner. DevOps promises to introduce agility, repeatability, quality, and governance into application delivery via collaboration between the two key teams responsible for the overall process.

Today, we are emerging from the economic "dark ages." Agile development practices have become the de facto standard for software development. Development teams are developing smaller releases in an iterative fashion, incorporating business stakeholders into the loop, and releasing code with increasing frequency – multiple times daily, in some cases.

However, while the benefits of Agile have been proven many times over, many IT leaders still have their doubts about DevOps. One reason for this is because traditional DevOps thinking focuses primarily on the Testing through Deployment stages of the application lifecycle. With conflicting demands for budget dollars, investing in a DevOps silo to bridge the roles of existing silos seems counterproductive to many CIOs.

Enterprise Management Associates® (EMA™) analysts have been researching the growth of both Agile and DevOps practices for several years. Based on this research it is apparent that while the DevOps function is alive and well, DevOps "teams" are scarce to nonexistent. DevOps has evolved as a far more cross-functional, collaborative, and lifecycle- focused activity than traditionally understood, and spans the application lifecycle versus being confined to a "point in time" handover of responsibility. This model of DevOps is far more effective for supporting the agile practices and business criticality that characterize today's application delivery systems.

In addition to being lifecycle-centric, this "new" DevOps is also more automated. Automation across the lifecycle makes tools the "bridge" between technology teams speaking different "languages." It also better addresses the Agile-centric methodologies which have gained momentum in the past several years by incorporating line of business stakeholders at key points in the lifecycle. This facilitates a more nimble and iterative approach to application delivery and provides a DevOps "fabric" that supports the "experts" who are so essential to building and delivering high-quality business applications.

To reiterate, one key differentiator from standard DevOps thinking is the idea that DevOps actively involves business stakeholders. Both the research and the case studies included as part of this EMA report bear out the fact that business stakeholders are as important as technology experts in optimizing the process. Instead of sitting back and waiting for Development to deliver software to their doorsteps, Line of Business (LOB) is far more actively involved and more visionary in identifying new opportunities for business enablement through software.

This paper elaborates on this lifecycle approach. It examines 21st century DevOps through a pragmatic lens, positioning "real-world" DevOps as an ongoing lifecycle versus a point-in time handoff. It details the key stages of the lifecycle (each stage typically encompassing its own lifecycle) along with stakeholders involved at each stage. It also discusses the leadership and responsibility transitions that occur at each stage

Methodology

This paper distills input from three primary sources:

- *EMA research:* EMA has been researching the emergence of cross-functional support and DevOps teams for several years. This paper highlights the latest findings.
- *Vendors and their products:* In the course of this research, more than thirty vendors with DevOps-related messages were interviewed and invited to provide written input about their solutions. This report includes a summary of each participating vendor's solutions (approximately half decided to participate), as well as how and where (at which stages) each vendor's product line addresses the DevOps lifecycle.
- *Case studies with "real world" IT professionals:* Vendors were also invited to provide customers for EMA analysts to interview during this process. Interviewees were asked a standard set of questions addressing their company's DevOps practices and their use of the vendor's solution(s) in supporting them.

DevOps Past and Present

Change – unprecedented rates of change – is the defining characteristic of modern IT environments. The convergence of "emerging" technologies and consumer-driven applications is driving businesses to become far more creative in designing, developing, and delivering applications.

Businesses in a wide variety of industry verticals are rolling out multiple code changes daily. Businesses are integrating on-premise and public Cloud into "hybrid" services. Agile development practices are involving line of business as key participants in the software lifecycle and "continuous improvement" initiatives are driving new paradigms for soliciting user-driven input related to software features.

Each of these scenarios impacts Development and Operations alike and each is a factor driving the growth of DevOps practices. However, like "Cloud," "SOA," and "APM," the term is defined differently by almost every analyst, vendor, and IT organization.

Techopedia's definition of DevOps is quite broad: "The term DevOps is commonly considered a combination of the concepts of Development and Operations. It is used in IT to refer to roles or processes that bridge various departments – usually Development and Operations teams – to achieve a certain project management philosophy that involves more efficiency in communications between Development teams and other parts of a greater business or organization."[1]

Today, DevOps messaging focuses almost exclusively on software deployment and the handoff between Development and Operations that ostensibly occurs once software goes into production. However, there are a variety of problems with this approach, the most significant being:

- *It does not address the levels of dynamic change (daily code releases) that have become the norm in many companies.*
- *It fails to account for the iterative nature of software delivery and the fact that almost all applications shuttle between Production and Development for updates and new features in an ongoing lifecycle.*
- *It creates a new "silo" centered around software deployment and release management which many CIOs view as being cost-prohibitive to implement.*
- *It fails to adequately address the automation aspects of the software lifecycle. Automation links processes and tools and provides continuity and context during pre-deployment and post-deployment stages of the lifecycle.*
- *It fails to account for the third significant persona in the software delivery process, which is the business stakeholder.*

Today, DevOps messaging focuses almost exclusively on software deployment and the handoff between Development and Operations that ostensibly occurs once software goes into production. However, there are a variety of problems with this approach

As the rate of change continues to accelerate, DevOps practices which DO take these factors into account are becoming increasingly business relevant. The remainder of this paper describes a new approach to DevOps that more closely approximates the cross-functional Development/Operations collaboration as it is evolving in modern IT organizations.

[1] Techopedia, downloaded February 4, 2013 from : www.techopedia.com/definition/13607/devops

Online Consumer Research Service Builds a Foundation for Agile Development and Operations with IBM Software

Vendor and Product Description

IBM Software

"IBM Rational is a pioneer, practice leader, and world-class platform provider in software and systems development, with decades of development expertise. Whether your team needs to improve the profitability of a software intensive product or service, or improve your IT productivity and reduce overhead costs, the tools and know-how from IBM Rational can help you succeed."[2]

Interviewee

Manager, Release Management and QA
Consumer Products Advertising and Research Service

Problem Scenario

This major online company runs its business on two primary applications that are updated on an ongoing basis. Almost three years ago, they "got to a point where delivering code to production involved too much interaction between multiple roles to be efficient or repeatable. Each role was responsible for multiple tasks related to the delivery of code changes to test and production environments. When you have a process like that the potential for error is significant."

In evaluating options, the Release Manager "went back to the drawing board" to identify what his teams needed to accomplish in terms of changing their processes. At the same time, they wanted to brainstorm ways to accommodate increasing volumes of code changes over time while becoming more consistent. They brought in a consulting firm for a three month engagement with a mandate to help the company develop repeatable process and select tools that supported their goals.

To kick off the engagement, the Release Manager documented two of the company's largest applications, code-wise, along with deployment practices. He then walked the consultant team through the entire process, covering code creation, check-in, and packaging for delivery. He also reviewed related Operational installation processes and QA validation procedures. His goal was to assist both internal and consulting teams in developing hardened processes that could be standardized, documented, repeated, and ultimately automated "all the way up to production."

Reining in Chaos with Automation

One focus was to simplify by eliminating as many manual steps as possible. However, deciding what to streamline and what to automate was "definitely daunting."

Deciding that tools could support simplification, the teams discussed solutions from IBM and other third-party vendors that could help the company achieve its goals. "We thought IBM tools could help us automate our processes, but weren't sure. We asked the consulting team to take us through the whole process of doing the automation end-to-end for both applications and train us to refine and automate going forward."

[2] Adapted from IBM website on 4/9/2013: http://www.ibm.com/software/rational/

The results were impressive. While they had contracted for a three-month engagement, by the end of two months they had developed automated processes for both applications. One key reason for the success of the project is the fact that the Release Manager had prepared documentation of key processes and steps before the consulting firm arrived. "I walked through each step in the process, including the timing of each step. We documented each component of the application and the systems that were part of the build, test, and production processes. In other words, we did the up-front prep work necessary to give the consultants a running start. The outcome is that we not only finished the project, we finished it ahead of schedule."

Building a Foundation for Future Growth

Another reason for the success of the project was the fact that the Manager kept upper management in the loop. "I created 'before' and 'after' diagrams of processes that had been automated. This really helped fill upper management in on what we accomplished."

To the consternation of the consultants, however, after the entire system was implemented, the internal team tore it down. "We dismantled the system until only the installed tools remained. Our consultants helped us build a framework and trained us in its use; however, it didn't take long to see how much better the process could be. I brought the system back up myself, incorporating capabilities such as naming standards for components supporting projects, libraries, and variables."

Once everything was rebuilt and operational, they continued to search for opportunities for continuous improvement. "The consultants had warned us that what often ends up happening is that as you go through the process you continue to find other things you need to improve. For example, while the process may be automated, you also find gaps.

"Three months after she said that, I picked up on what she meant. The whole process was automated and took about two hours to execute. However two hours started to seem too long. We took a look at what was inefficient, and this became our next evolution.

"At the same time, what was also interesting is that we started to pilot Agile. What we hadn't initially anticipated was that the automation already in place would make Agile adoption considerably easier. Part of moving to Agile is the fact that it requires a level of automation on the Operational side. Having this in place made the adoption of Agile development practices much more straightforward."

IBM and Process Automation

One capability cited as key to success is that fact that the IBM solutions were easily adaptable to the company's existing processes. "Once the system was in place, everyone from Development to QA to Operations became very excited. The liked the fact that a single automation stream enabled their existing processes to run reliably.

"Instead of reengineering our processes, the IBM tools allowed us to take what we were already doing and automate it. IBM Rational Build Forge is primarily a process execution engine. We see it as our 'build and release factory.' Like an assembly line in an automobile plant, it executes whatever is coming down the product line based on the processing you have in place."

He views IBM Rational Automation Framework as an extension of the factory idea. "It adds the bells and whistles that enable you to automate components specific to the IBM stack. It provides additional tools and libraries to configure WebSphere, for example, to more easily manage the entire process."

Outcomes

Capacity: Using the new platform the company performs releases and builds in 77 percent less time. It also now averages 300 new product releases a year versus the 30 to 40 it deployed in the past.

Consistency: Applications are delivered in a similar way, reducing training time and making the delivery process more predictable for all roles that touch the lifecycle. "Everything became very consistent. It is easier to train, Operations can see what is coming ahead of time, and releases are orchestrated and documented."

Velocity: "It is amazing how quickly applications can be running 'on the glass' (in production). Processes which took two hours to execute when we first built the system now take 30 minutes."

> "Processes which took two hours to execute when we first built the system now take 30 minutes."

Foundation for Agile development: Although many companies first implement Agile development practices and later realize they need Agile Operations as well, this company did the reverse. They first put the foundation in place to support existing processes, and then found the evolution to Agile to be relatively painless.

Improved pre-production testing: "We are trying to improve the testing process and make it more interactive. Testing is the new frontier – we are trying to intertwine the system and application levels to do front-end testing more effectively and in an automated fashion."

Quote

"Be prepared before you start the journey and realize that the journey never ends as there is always room for continuous improvement. Get buy in from those in the trenches, and keep looking for ways to improve"

A Traditional DevOps Model

Figure 1 depicts a traditional, deployment-centric view DevOps within a six stage application lifecycle. From this perspective, a skills handoff from Development to Operations takes place during application deployment. While this handoff is in process, personnel with both Development and Operations skills work together to ensure that:

1. *The application is deployed as designed, with associated scripting, configurations, interdependencies, installation sequences, and "go live" processes.*
2. *The application is production-ready.*
3. *The operations environment is configured correctly and robust enough to support production use.*
4. *The production environment meets the functionality, performance, and availability requirements of the business.*
5. *The application meets end-user needs and expectations.*

Since it is typically the case that neither Development nor Operations has the breadth of skills necessary to support all of these goals, cross-silo collaboration helps ensure that the "as deployed" application functions "as designed."

This view of DevOps was definitely a step in the right direction. It provided a physical implementation consistent with ITIL Release Management guidance, centering on governance, process control, and collaboration. However, it may be time to reevaluate how and where DevOps fits and how it has been implemented within modern IT organizations. As applications become increasingly distributed, Cloud-hosted, and business critical, real-world companies are extending DevOps philosophies and collaboration models across the application lifecycle.

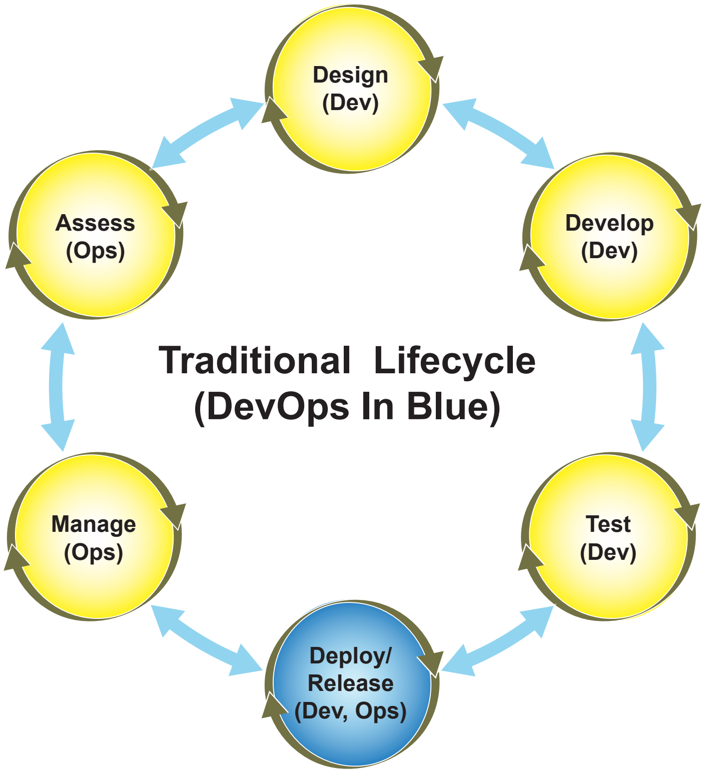
Design (Dev)
Develop (Dev)
Test (Dev)
Deploy/ Release (Dev, Ops)
Manage (Ops)
Assess (Ops)
Traditional Lifecycle (DevOps In Blue)

Figure 1

Why a New Approach?

A number of technical and organizational factors are driving the expansion of DevOps to a more lifecycle-centric approach. They include:

- ***Iterative development impacts the entire lifecycle, not just deployment:*** The broad adoption of Agile methodologies in the enterprise means that the deployment stage has transitioned from being a "point in time" turnover of responsibilities to an ongoing process. Both Development and Operations are feeling the impact of this transition. Development must find ways to build and test code incrementally and iteratively, and Operations must find ways to engineer production environments capable of absorbing high rates of change. Neither group can do this without the other. In addition, Agile participants often assume multiple roles at various stages in the lifecycle. This can introduce greater risk if automation is not in place.

- ***Abstraction of applications from hardware impedes manageability:*** Cloud, mobile, sensor systems, and virtualization are freeing applications from the bonds of the physical data center. While moving applications and/or application components "to the Cloud" can provide significant business benefits, the abstraction of these technologies makes applications more complex to manage and govern. When applications are externally hosted, IT loses the visibility and control they need to effectively monitor and manage application systems. Finding answers to the problems of an expanding application ecosystem requires collaboration across business and IT.
- ***Application heterogeneity:*** From multi-brand server farms to server clusters, load balancers, Web servers, and heterogeneous databases, even applications hosted 100% on-premise are so complex that supporting them requires skills that are both broad and deep. This is true across the entire six-stage application lifecycle, making it imperative that DevOps is approached as a lifecycle versus a point in time handoff.
- ***Automated deployment:*** With the growth of highly virtualized environments deployed as both private Cloud and public Infrastructure as a Service (IaaS), software deployment is becoming increasingly automated and metadata-driven. However, as functionality previously supported by run books and manual job submissions is automated, the primary role of IT – to deliver quality applications to the business – remains the same. Many CIOs are seeking new and better ways to deliver on this mandate, and DevOps supported by automation is a proven approach for doing so.
- ***Emergence of cross-functional teams in enterprise IT:*** As application ecosystems have become more diverse, cross-functional support teams have emerged by necessity. Called by a variety of names – Application Support, Architecture, Infrastructure Services, and Application Management are common ones – teams of cross-trained IT specialists are now part of mainstream IT. Most often, these teams become involved with the application at multiple points in the lifecycle, whenever critical skills intersect with business need. Whether they are called "DevOps" teams or go by another name, they perform the DevOps function and have the cross-functional skills profiles associated with DevOps roles.
- ***Increased focus on line of business as part of the application lifecycle:*** A key element of Agile methodologies is the involvement of line of business throughout the application lifecycle. From requirement definition to iterative check points to acceptance testing, business stakeholders are more involved than ever before. Traditional DevOps thinking virtually ignores this group, which is becoming increasingly indispensible to customer satisfaction and project success. The DevOps lifecycle unveiled in this study includes line of business as a third stakeholder, and one which assumes key leadership roles at specific stages of the lifecycle.

Redefining DevOps for a New Generation

The lifecycle model introduced in Figure 2 has two key differences compared to "traditional" DevOps. Not only are DevOps concepts extended across the lifecycle, but Development, Operations, and Business stakeholders are involved at every stage. While all three roles work together, the lead role(s), underlined in the graphic, change(s) at each of the six stages. Collaboration takes the form of cross-functional cooperative efforts, and leadership handoffs replace a full-fledged relinquishment of responsibility. Each stage of the lifecycle shares relevant data with the previous and following stages.

Each of the six stages also encompasses its own, unique "mini-lifecycle" or iterative process, typically supported by specialized tools. However, in the lifecycle approach, interoperability is an essential capability for any tool, as it enables information to be shared across the stages.

The six stages are as follows:

1. ***Assess:*** Feedback from business stakeholders and Service Level Agreement (SLA) assessments are used to determine priorities for new applications and modifications to existing applications.[3] Development and Operations stay in the loop because such improvements likely require new or modified code as well as changes to operational infrastructure.
2. ***Design:*** Development and Business take the lead in this stage, with Operations available to provide input as needed. Development has primary responsibility for incorporating Business requirements into a software design. Business has responsibility for educating Development about specific requirements as well as for reviewing and signing off on the design. The role of Operations during this stage is to be aware of the modifications to anticipate potential impact on production systems and assess requirements for infrastructure enhancements (Capacity Planning).

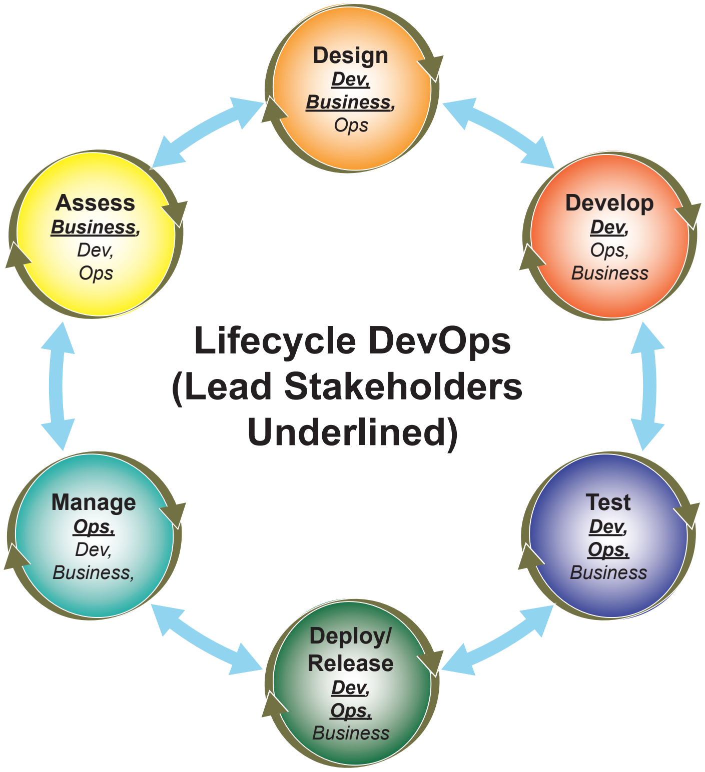

Figure 2

3. ***Develop:*** Development is the lead during this stage and responsible for building software that meets the needs of the business. Particularly in Agile shops, Business stakeholders perform functional reviews and sign off on application functionality on an iterative and ongoing basis. On an "as needed" basis, Operations personnel build and support Development and testing environments.
4. ***Test:*** Development and Operations are the leads during this stage, as final preparations are made for integration testing and release to production. While Development and QA teams perform unit and integration testing, Operations participates in integration and load testing to assess operational readiness. Acceptance testing becomes a critical role for business users, and all three groups must collaborate to agree on a final "go live" plan.
5. ***Deploy/Release:*** This is the traditional "DevOps" handoff stage, but in this scenario the handoff is a change in lead roles versus a turnover of responsibility. This stage is roughly equivalent to the ITIL Change and Release Management processes, although in many companies these functions are not formalized.

[3] Ongoing improvement is also a key tenet of ITIL v4

6. ***Manage:*** EMA research has shown that for applications written in-house, Development typically remains "in the loop" in terms of application support for the life of the application (although this is not the case for packaged applications). At the same time, production application quality depends on far more than code. It is the sum total of code quality, quality of the infrastructure/hosting environment, and user experience. Cross-functional collaboration becomes critical, because this is where applications actually touch end users. Also during this stage, monitoring tools gather performance and availability data for real-time management as well as for identifying trends and issues over time. This information is then fed into the next (Assess) stage, where opportunities for ongoing service improvements are identified.

"Real World" DevOps: Hiding in Plain Sight

EMA research has found this extended lifecycle to be far closer to the reality of application development and support in modern companies. While relatively few companies report adoption of "DevOps" per se, a significant majority DO report the existence of "cross-functional IT teams that work together to solve application-related problems."

EMA surveys have been tracking the rise of cross-domain support for more than five years. As applications became increasingly distributed, heterogeneous, and complex, leading-edge IT organizations recognized the fact that traditional application support practices were simply not working. Tribal knowledge and whiteboard service models cannot support the levels of scale, change, and complexity underlying the average distributed application, while war room problem-solving can burn hundreds of hours of expensive staff time to solve a single application problem. CIOs recognized the need for a better way and evolved their organizations to more effectively adapt to changing business needs, modern development methodologies, and dynamic operational environments.

EMA continues to assess the personnel, process, and tools-related factors that bring order to the chaos of managing today's application ecosystems. Cross-domain processes and tools are emerging as key differentiators for improving operational efficiency, enabling IT organizations to more nimbly adapt to business and industry changes, and continuously improving the quality of applications delivered to the business.

A cursory examination of the research gives the initial impression that DevOps is a non-starter. However, a deeper analysis reveals that it is far more prevalent than it appears to be at first glance.

As Figure 3 shows, only about 30% of companies report that they are implementing DevOps or plan to do so in the foreseeable future. Of that group, approximately 85% believe their organizations are capable of effectively transitioning to DevOps (Figure 4); however of those that don't, the top reasons are tools related (Figure 5). The top two concerns are that Development and Operations tools are not well-integrated and that they lack the general automation readiness needed to support the DevOps function.

A cursory examination of the research gives the initial impression that DevOps is a non-starter. However, a deeper analysis reveals that it is far more prevalent than it appears to be at first glance.

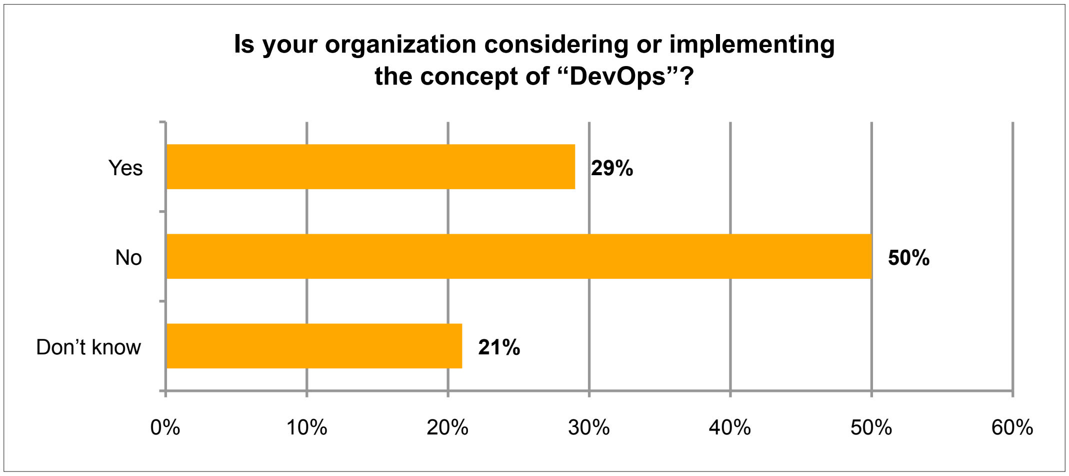

Figure 3

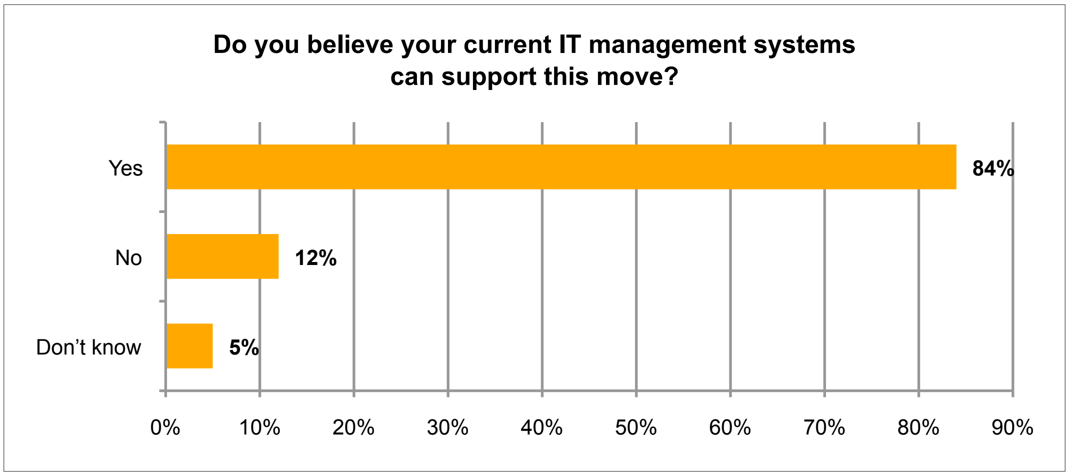

Figure 4

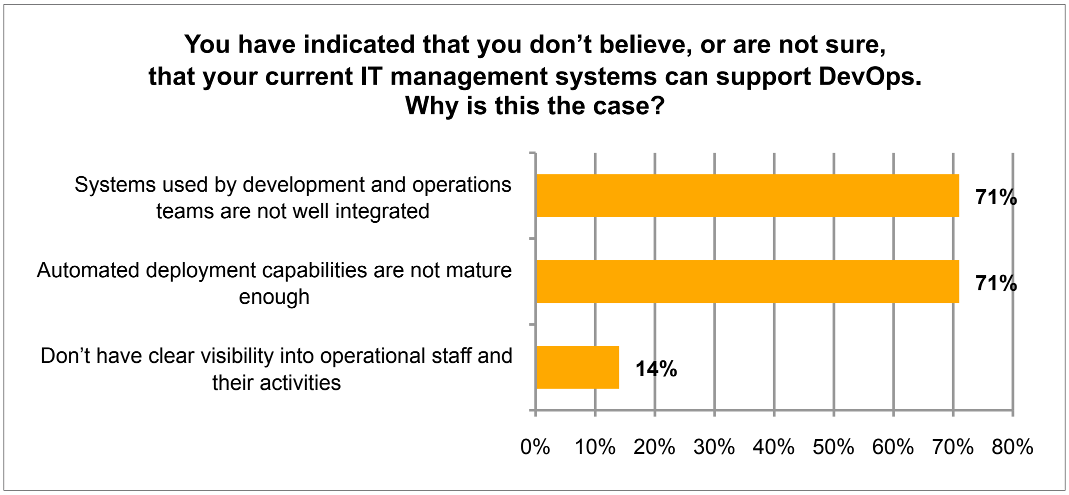

Figure 5

However, as Figure 6 shows, although only 30% of companies are actively engaged in "DevOps," almost 90% "leverage a team-based approach in which specialists with cross-functional Development and/or Operations skills work together." Figure 7 reveals that for the majority (almost 60%), this is a dedicated versus ad-hoc team.

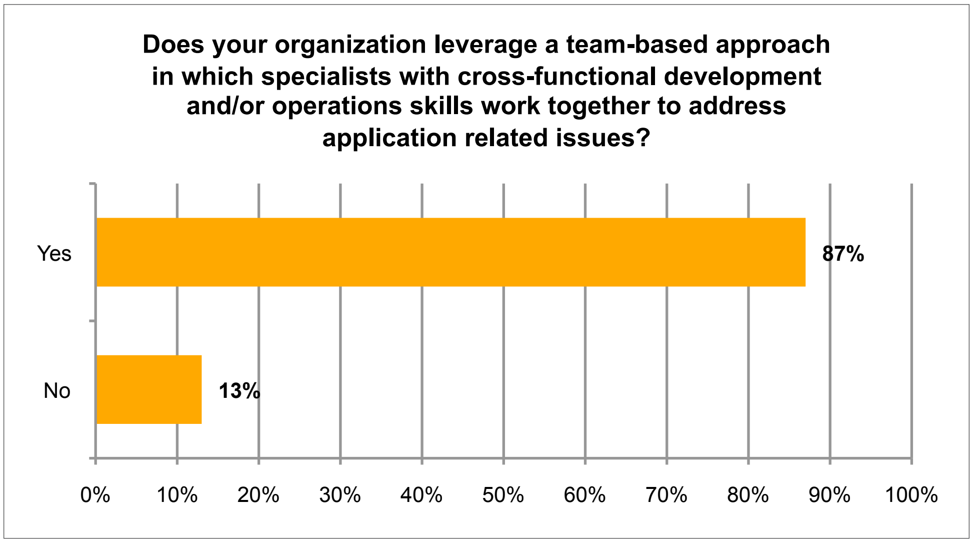

Figure 6

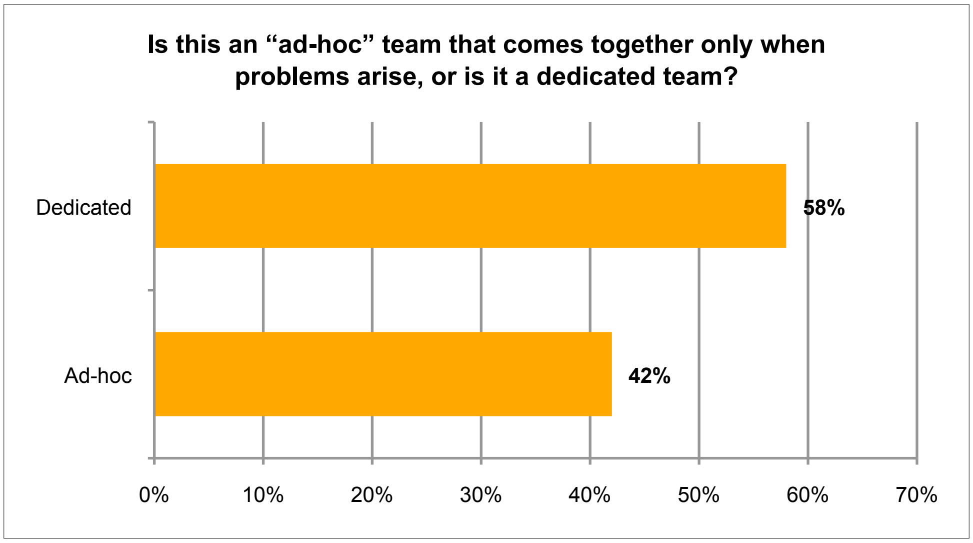

Figure 7

As Figure 8 shows, only 5% of companies call this group "DevOps." More common names include Infrastructure Services, Architecture teams, or Centers of Excellence. A clear conclusion is that although dedicated teams with cross-functional Dev and Ops skills are active in the majority of companies, the "DevOps" term is infrequently used.

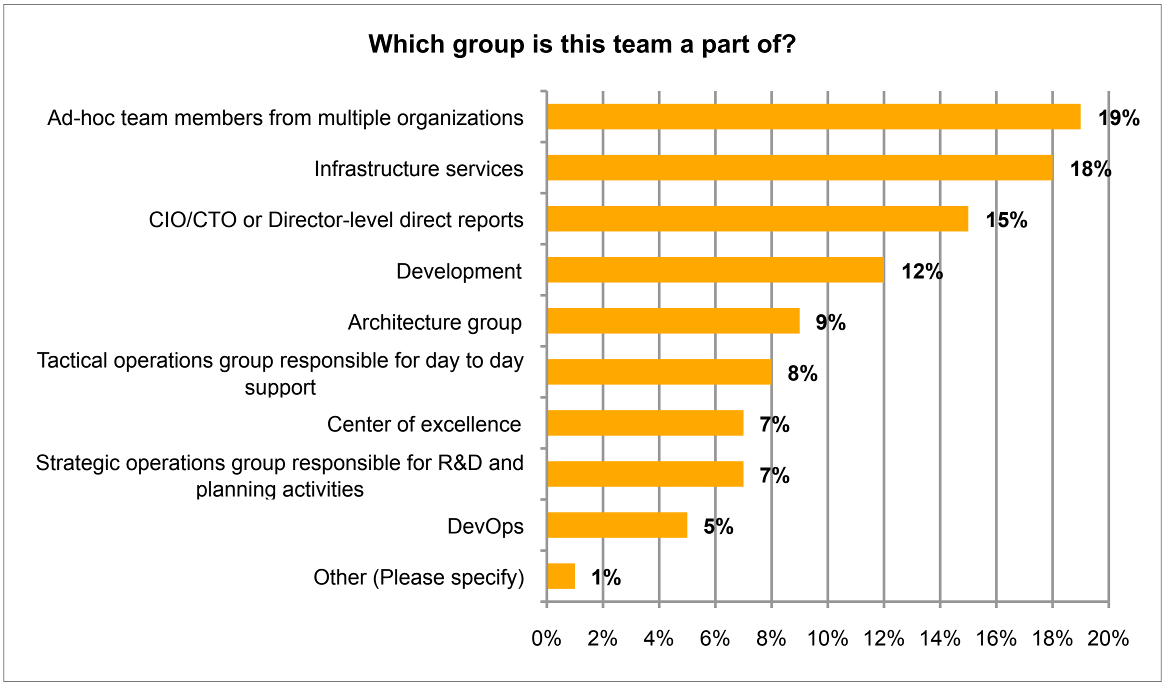

Figure 8

Finally, Figure 9 shows that this team is typically funded at the CIO or Director level. This underlines the fact that IT executives are directly involved in overseeing the quality of application delivery. It also highlights the fact that they see cross-functional application support practices as being essential to delivering quality services to the business.

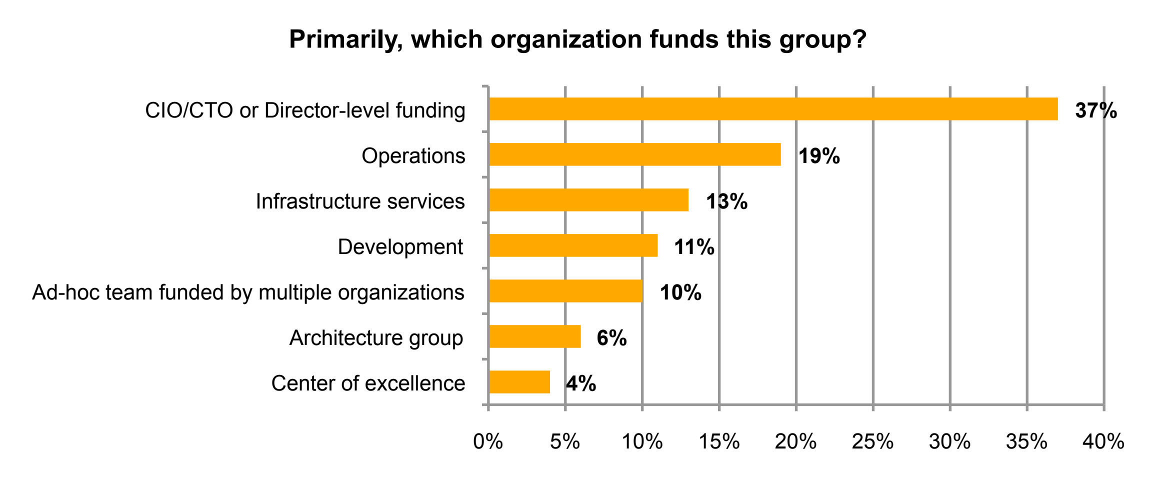

Figure 9

In short, the research suggests that "real-world" collaborations across Development and Operations are more often an ongoing effort than a point-in-time handover. It also shows that while only about 5% of companies are investing in "point-in-time" DevOps, almost 90% are investing in cross-functional, cross-lifecycle teams. From this perspective, the tooling aspects of application support become critical as tools become a key factor supporting a unified lifecycle approach.

Financial Services Company uses IBM Software to Build Agility and Continuous Integration into DevOps Lifecycle

Vendor and Product Description

IBM Software

"IBM is a pioneer, practice leader, and world-class platform provider in software and systems development, with decades of development expertise. Whether your team needs to improve the profitability of a software intensive product or service, or improve your IT productivity and reduce overhead costs, the tools and know-how from IBM Rational can help you succeed."[4]

Interviewee

Director, Application Development
Insurance and Financial Services Vertical

Problem Scenario

This company was challenged to adapt to an industry that is both in constant flux and highly regulated. "Our industry has been changing so rapidly that we realized three or four years ago that we needed to execute the development lifecycle much faster. At the same time, our industry is highly regulated. To address both of these concerns, we made a concerted effort to adopt Agile development practices, which in turn drove new requirements for continuous integration.

"We are not releasing software daily and probably never will. Our industry is just too regulated for that. Instead, our releases are divided into six to twelve two-week sprints or iterations. In general, our teams release new software every few months.

"As we began to promote Agile as a core practice, the idea of having environments that were highly governed, managed, and reliable was very compelling. However we also wanted teams to be able to own and manage the environments on a self-service basis and to deploy them via standard configurations."

Building a Tools Framework to support Agile Processes

The company had a mix of tools with some IBM products already in-house. When they started down the Agile path, they sat down and constructed a plan of how the processes would look and what types of tools would be needed to support Agile practices.

They started with IBM Rational Team Concert (RTC) which they used to introduce the idea of work item management into their processes. They leveraged IBM Rational Requirements Composer for the requirements side. They then began to focus on how their existing tools integrated versus allowing each team (or stakeholder) to select tools on an individual basis.

"This turned out to be a strength for us, as we took a step back and looked at our tools architecture from an end-to-end perspective. We were able to use both our IBM and third-party solutions, and IBM assisted us with the integrations. With a shared vision of our constraints and requirements, we solved our tooling issues as a team."

[4] Adapted from IBM website on 4/9/2013: http://www.ibm.com/software/rational/

They adopted IBM Rational Automation Framework (www.ibm.com/software/products/us/en/ratiautofram) to deliver the ownership and self-service capabilities the teams needed, and set up Java Web hosting along with standard configurations for continuous integration. "Teams can now request and provision environments ten times faster than was previously possible."

"Teams can now request and provision environments ten times faster than was previously possible."

Once Agile began working well and they were seeing improvements in productivity and quality, they expanded the processes enterprise-wide. The next challenge was to replicate Agile practices for the mainframe. They decided that they did not want to have multiple sets of processes depending on the technology and extended their Agile processes across the technology ecosystem. "From an engineering practices standpoint, we decided to see how far we could push the envelope. We now use IBM Rational Development and Test Environment for System z (RD&T) to bring mainframe into the lifecycle.

"We are not yet as automated with the mainframe as we are with Java, since we use Java Web hosting and can spin up environments automatically. With Java, we also do continuous integration with Maven and Hudson. However, the same concepts of continuous integration are still applicable and we would like to get to the point where we also have standard deployment configurations for RD&T."

DevOps and Line of Business Interact as Part of One Team

Since Agile, continuous integration, and continuous deployment were drivers for change, they were already using these practices prior to setting up their DevOps groups. "We set up cross-functional teams with developers, testers, Database Administrators (DBAs) and other specialists. Our teams are composed of roughly twelve to twenty personnel. We also have an infrastructure role that works closely with each area to help facilitate issues such as procuring environments and troubleshooting. Although we still do have separate Infrastructure and Ops organization, the groups now work much more efficiently together."

In terms of LOB involvement, each development line has a product owner that sits with the team and answers any questions. The product owner is also responsible for prioritization of story cards and show and tell activities. Each line has at least one person dedicated to providing the business perspective.

The teams physically sit together in pods versus cubes and are very collaborative. Including a product owner from the business side means that the entire lifecycle is transparent from the business perspective. "We can change direction very quickly based on what's best for the business. This is helped by the fact that when you have a development/test environment where you aren't breaking things all the time, you can become much more nimble."

They also found that dedicated versus ad-hoc DevOps teams produced the best results. "Initially, multiple projects touched a given set of assets, and teams were constantly being broken apart and re-formed to support each step in the lifecycle. Just as a team started to perform well, it was disbanded. We found that this wasn't sustainable. So we set up our cross-functional groups around sets of software assets. Our teams are now dedicated versus ad-hoc, and we have found this greatly facilitates the effectiveness of the teams."

Our teams are now dedicated versus ad-hoc, and we have found this greatly facilitates the effectiveness of the teams."

Outcomes

- *Self-service environments*: "Standing up environments is now much faster. Team members have administrative permissions and this helps make self-service a reality. If systems are 'trashed' they can be rapidly reset from a standard configuration. Developers can deploy automatically, and generally do deploy to test environments once or twice a day."

Automated testing: Every time a developer checks in code, a set of unit tests automatically runs through IBM Rational Application Developer (www-ibm.com/software/products/us/en/application/). Each developer has his/her own version of RAD on the desktop, and some regression testing is performed every time code is checked in. Once or twice a day, they deploy whatever code was changed that day to the development environment where regression testing is done to make sure "nothing is broken by the new code."

- Testing also runs overnight, and in the morning the technical lead checks to make sure that "nothing is broken in the build." If it is, the first thing they do that day is fix the build. In this way, code is tested multiple times based on daily checkins.
- *"Zero bugs" within reach:* "While there is still a role for system testing, that role is much leaner now. It isn't at all unusual to deliver zero bugs for system testing. This is a result of the fact that we are doing continuous development, integration, deployment, and testing."
- *Monthly releases and continuous improvement:* As software proceeds through the lifecycle, it is eventually deployed to system and performance testing environments. Production software is deployed from these environments via monthly releases. "At the enterprise level, we deploy two releases per month, one for applications and one for infrastructure. We also focus on continuous improvement with each release."
- *Flexible use of resources*: "We want to become more agile still with development testing and deployment, to be able to standardize environments, provision quickly, and de-provision equally quickly. It used to be painful to reclaim environments because they were so difficult to build and track. Today, it is easy to provision for a team that needs an environment for only a few months. We continue to incrementally improve the process to see what makes sense and we have data to show that we are much more productive since we started down the Agile and DevOps path."

Final Takeaways

"We have worked closely with IBM to get an understanding of how best to utilize/integrate tools and to continuously improve. Automating via tools is an important skill and we have continually asked ourselves what problem we were trying to solve and how could we make our existing tools solve it. But regardless of whether Tool A is better than Tool B, a big part of the puzzle is working with a vendor which understands and is responsive to what is important."

"We kept this in mind when considering toolsets. As part of the tools selection process, one incumbent tools provider ultimately lost out to IBM because of their attitude towards us and the fact they didn't listen to us. Because of the effort IBM made to collaborate and partner with us, they didn't even have to sell us – the relationship made all the difference."

Quote

"Change is necessary, but you need a process that is adaptive. We are comfortable being adaptive because we know change won't wreak havoc with the quality of our software."

Cross-Lifecycle Tooling: Bridging Diverse Groups, Skills

Approaching DevOps as a lifecycle has significant implications in terms of tools architectures, options, and selections. As DevOps progresses from being a point-in-time handoff to an ongoing process of creating, deploying, and supporting business software assets, tools become a primary factor unifying lifecycle stages, roles, and leadership changes. The implications are significant and center on several key areas:

As DevOps progresses from being a point-in-time handoff to an ongoing process of creating, deploying, and supporting business software assets, tools become a primary factor unifying lifecycle stages, roles, and leadership changes.

- ***Integration:*** Modern applications don't exist as discrete entities. Instead, they exist as distributed ecosystems that must integrate and interoperate for an "end-to-end" application to become a reality. In the same way, tools must support the fact that technology silos don't live in a stand-alone world. Instead, they are part of a multi-stage continuum. While each stage brings with it specialized toolsets to support its own internal lifecycle, tools must interoperate to support seamless collaboration across stages.

  For example, Application Lifecycle Management (ALM) tools are commonly used to manage requirements, code modifications, testing outcomes, and similar development-related tasks. Release Management tools address factors such as release workflows, change approvals, scheduling, deployment, and automated validation testing. Application Management tools monitor and manage the application in production and often during pre-deployment testing as well. Approaching DevOps as a lifecycle requires information sharing across discrete toolsets. Interoperability is a unifying force across diverse teams, skills, technology languages, and methodologies.

- ***Workflow:*** As applications become increasingly complex, building, deploying, and managing them becomes equally complex. Workflows are essential to governance and traceability, particularly since most IT processes combine human and automated tasks. One of the biggest challenges facing today's technology teams is one of managing complex, multi-step processes in context to manual tasks, automated steps, and reviews/approvals. This is particularly true in cases where multiple projects or tasks have dependencies on others, and where multiple projects (and deployments) are being completed in parallel. In such environments, the sheer effort of "keeping track" of bottlenecks, work queues, point-in-time responsibilities, and similar factors can be gargantuan. This, combined with the touch points across stages that lifecycle DevOps entails, makes workflow a key capability for companies embracing this approach.

- ***Cross-lifecycle service quality supported by tooling:*** Research studies have repeatedly found that the best way to reduce the cost of supporting applications is to find and fix flaws early. A famous National Institute of Standards and Technology (NIST) study reported that if an application problem fixed during the design stage costs X, the same problem fixed during production costs between 470 and 880X. If the change requires an engineering change, the cost could be as high as 2,900X.[5]

  The clear implication is that testing early, often, and iteratively is a far more cost-effective strategy than waiting to find problems once applications are released. In terms of DevOps, this means moving performance and availability testing to a point far earlier in the lifecycle, a task that can be accomplished by investing in testing tools capable of spanning both Development and Production.

[5] The Economic Impacts of Inadequate Infrastructure for Software Testing, http://www.nist.gov/director/planning/upload/report02-3.pdf

- ***Budget, staffing, and quality efficiencies:*** Tools that enable data sharing across the lifecycle not only extend the value proposition of tools investments, they also empower technical personnel to move beyond the boundaries of their own skillsets. It is unreasonable to expect that developers be Operations experts, that Operations personnel be security experts, or that Security experts understand how to install an Operating System on a mainframe. However, it is reasonable to expect that applications are delivered at the highest levels of quality possible, and that quality is built in across the lifecycle.

  There are a number of tools-related capabilities that contribute to this positive outcome. One is that IT specialists be able to see the impact of their own silo on the quality of the application as a whole. Another is that IT as a department is capable of rapidly solving application-related problems in complex ecosystems. Finally, and potentially the most important, is for tools to be unifying forces versus roadblocks to cross-functional collaboration.

Vendor Solutions and the Lifecycle Approach

Many of today's enterprise management vendors are focusing on DevOps and some of the leading vendors in the space provided input into this report. The following table provides information regarding the stages of the lifecycle each vendor supports as of this writing. The Appendix contains detailed information related to the ways in which each vendor's products support the stages checked in the table below. This information is intended to provide readers with a starting point for developing a short list of vendors and products supporting specific DevOps stages and automation requirements.

Vendor	Assess	Design	Develop	Test	Deploy/ Release	Manage
AppFirst	✓			✓		✓
Boundary	✓			✓	✓	✓
CollabNet	✓	✓	✓	✓	✓	
dbMaestro			✓	✓	✓	
ExtraHop	✓	✓	✓	✓	✓	✓
IBM	✓	✓	✓	✓	✓	✓
Nolio				✓	✓	✓
OpTier	✓	✓	✓	✓	✓	✓
OutSystems	✓	✓	✓		✓	✓
Serena	✓	✓	✓	✓	✓	✓
Splunk	✓			✓	✓	✓
Xebia Labs				✓	✓	✓

Table 1: "DevOps" vendors and the lifecycle stages supported

Summary

The rise of DevOps was driven by necessity. Businesses forced to "do more with less" found themselves struggling with the need to introduce applications into production far more frequently than they were used to. As requirements for more agile deployment methodologies collided with the realities of resource availability, IT organizations found themselves in a nearly perfect storm. They were forced to embrace change, a factor that has traditionally been the enemy of technology environments as it is the primary source of production failures.

Perhaps coincidentally, this evolution coincided with an exponential increase in technology complexity. Cloud, virtualization, hybrid Cloud, mobile and wireless introduced networks of loosely coupled, hardware-abstracted services that are complex to engineer and impossible to manage without collaboration and commercial enterprise management solutions.

In short, rapid change, limited resources, and the chaos of evolution drove a need for more effective collaboration and cross-functional orchestration. Initially introduced as a panacea for reining in these pain points, DevOps has morphed from being a methodology for transitioning applications from development to production into a far more lifecycle-focused process. And while "breaking down silos" has also become an ongoing industry buzzword, leading IT organizations have understood the need for deep, specialized skills and quietly built teams capable of pooling skills and "know how" to solve the industry's most difficult problems.

The role of tools in these environments cannot be underestimated. Tools are the key to reining in complexity and facilitating collaboration via a common language across diverse skills. Tools enable team members with specialized skill sets to view the application ecosystem from the perspective of their own areas of expertise. Problems which can't be solved by a single person or silo can be solved with the combination of human and tools-based expertise.

Undoubtedly, the combined effects of an unfavorable economic climate and rapid technology evolution accelerated the evolution of DevOps. With all the negativity associated with "the downturn," one beneficial outcome cannot be ignored. Businesses have learned to work far more efficiently, and lifecycle DevOps is one positive result.

Appendix: Vendor Descriptions[6] of Product Functionality at Each Stage of the Lifecycle

AppFirst

Test: To understand the resources required for the application running in production. Load test to see what happens from a resource utilization perspective to the application as load is increased. You can also see things like memory leaks, network abuse, etc.

Manage: Ensure the application is performing in production. Be alerted when things start to trend poorly so Ops can be proactive and fix something before the end customer is affected. Correlate all the sources of data we collect to troubleshoot issues.

Assess: Using StatsD you can get any metric out of an application. Therefore the business can understand things like what features are being used, what aren't, how often. The power of StatsD is endless and can provide amazing visibility.

Boundary

Load Testing: Customers use Boundary in their testing environments to understand how dynamic load patterns impact performance. This is particularly useful when testing massively distributed applications operating in Cloud environments. In real-time, Boundary can visualize the performance impact of a change across different components of your application. It gives you real-time host-to-host communication, latency, network retransmits, volumes and helps you determine if there are any performance bottlenecks in your application and allows you to quickly identify those culprits. It also helps ascertain whether the components are talking to appropriate application tiers, and whether there are any unexpected configuration or networking errors within your application.

Release: Customers integrate Boundary with their release automation tools to help understand the impact of new releases or changes on application performance. This is particularly interesting in shops where continuous build methodologies are in use. Boundary gives you ability to visualize the change events (e.g. new patch deployment) in context of the real-time application performance. Boundary easily integrates with third-party systems like Chef and Puppet to capture those events and process them as part of the overall application context. If a new release slows down your application, or creates unexpected application flow, Boundary will help pinpoint the change and impact.

Manage: Customers deploy Boundary in operational environments to monitor in real-time the overall performance of the application flow and the flows between various tiers of the application. For each host, Boundary presents the volume of traffic flowing in/out, the ports involved, latency, network retransmits, etc. Hosts can be grouped into Applications for easier administration. Boundary can watch for specific conversations between hosts or all traffic flowing through your application. Alerting capabilities help you proactively detect patterns of interest early and remedy the situation before your application fails. The one-second resolution of application metrics ensures that you do not overlook the smallest of symptoms of application failure. Boundary also determines which third-party networks your application is connecting to, thereby allowing you to monitor these interactions and measure the impact of third-party performance on the overall health of your system. Boundary automatically discovers the topology of your application network. Armed with this information, you can further drive impact analysis for operational or business needs.

[6] Descriptions provided by the respective vendors, minimally edited by EMA

Assess: Customers use Boundary to optimize their application architecture over time. As the application traffic grows, Boundary helps customers determine whether the current topology can deliver the desired performance. Based on this feedback, customers can experiment with different deployment architectures (say, relocating application components to the different availability zones or same availability zone inside AWS as the service endpoint being called) and determine the topology that best meets their needs. Boundary also helps customers assess their overall bandwidth needs and ensures they are not overpaying for capacity they don't use. Boundary also helps answers such questions as: Are two nodes excessively chatty? Are there traffic paths that are making too many hops? Is one cluster consuming too much bandwidth? Is the cloud provider meeting SLA promises?

CollabNet

Design: TeamForge enables agile planning, design and task management for distributed teams.

TeamForge manages and streamlines the agile design process. Product and business owners collaborate using a common platform within integrated design tools and documents, requirement trackers and project workspaces. Requirements and technical specifications from IT Ops and engineering are codified early on into the backlog. The platform facilitates requirement prioritization into the design process with interactive team task boards. Design documents are securely stored and versioned, and approval workflows enforce content review and sign-offs by technical and business stakeholders. Design progress is measured and visualized via graphical team dashboards.

Develop: TeamForge lets organizations streamline and govern development at scale.

TeamForge provides the collaboration and productivity tools to simplify development and improve code quality. From their IDE (Eclipse, Visual Studio) or browser, developers access the context-rich TeamForge environment: Code repositories (Git, Subversion), tracker artifacts, documents and integrated tools. Search and discovery tools provide rapid access to code and artifacts, driving reuse.

TeamForge scales to the world's largest Development teams, supporting geographically distributed development, and supports multiple methodologies from Agile to waterfall. TeamForge enforces access controls to code and artifacts, protecting IP compliance and ensuring development process standards.

Test: TeamForge manages test processes, it elastically provisions test systems, and connects test tools.

TeamForge lets organizations define, standardize, launch and control build and test activities. The platform automates continuous integration, through pre-packaged build/integration servers such as Hudson/Jenkins. Test scripts are documented and versioned, in application context. CollabNet integrates test tools such as HP QM (Quality Manager), TestMaker and Janova.

TeamForge Lab Management lets users provision build- and test-servers elastically and on-the-fly. Servers are pre-configured based on IT Ops approved system/configuration templates, and can reside in CloudForge, Amazon EC2, or any data center.

Deploy/Release: CollabNet Deploy with UC4 provides application release automation (ARA) at scale.

In combination with CollabNet TeamForge, CollabNet Deploy automates and orchestrates all activities involved in application release and deployment. All deployments are governed by permission management, to ensure segregation of duties. CollabNet Deploy is flexible and allows for different environments, owners and processes. It handles upgrades, changes, patches and first-time deployments—from development through multiple intermediate stages to production. CollabNet Deploy provides visibility and control across all application releases, enterprise-wide, and provides full control of the process (pause, continue, stop, roll back) and integrates with enterprise ticketing and management applications.

Manage: CollabNet solutions do support IT Operations teams; however, are not intended to provide a full-fledge IT Operations management solutions.

IT Operations and support staff use TeamForge to investigate applications and patches for feature/scope coverage and code quality. TeamForge provides lifecycle traceability, associating apps and patches with underlying code commits, builds and tests, and the individuals involved in the code release. These associations provide context and insight, so teams better understand the rationale for project decisions. TeamForge also provides an efficient means to capture production feedback and communicate back to product and engineering staff, using trackers and wikis.

Assess: TeamForge provides the agile capabilities to rapidly feed business requirements back to engineering.

Business owners and technical users communicate objectives and review application statuses using shared project spaces. This helps teams identify early-on opportunities for improvement and risk reduction. Stakeholders can view projects resources at a high level or drill into specific releases, iterations or milestone views. Aggregated metrics and reports provide program-wide visibility. Pre-packaged analytics, based on best industry practices for application lifecycle management provide application processes oversight across business stakeholders, including LOB managers, engineering management, IT management and compliance management.

dbMaestro

TeamWork™ allows organizations to complete their DevOps automation managing and controlling the database changes in the following steps:

Develop: As developers perform their database changes, each change is documented and linked to the business requirement (the key to be able to promote changes selectively). dbMaestro TeamWork has a unique Change Policy Enforcement that enforces Check-Out/Check-In operations on the database objects regardless of the database IDE.

Test: The business performs application testing, while also testing the deployable package. Every test can result in additional fixes to the application. This results in another iteration(s) of promoting changes from development to testing. Each iterative deployment may consist of application executable(s) which override the existing one(s) with different database script(s) (as some of the changes were promoted in the previous iterations). dbMaestro TeamWork automatically generates the database deploy script of the current iteration taking into account what had been promoted with no need to re-execute (sometimes re-executing can cause damage, for example: switch data between columns, the second execution is wrong) and changes made in this iteration and needs to be promoted

Release: Operations deploy all database changes to production. The database deploy script must be automatically generated upon the current state of the production database in order to have a success deploy.

"While every tier is complex and has its challenges, the database tier acts as the heart of the organization; all key applications eventually access and modify data existing in the database. Every change needs to be reviewed and tested. DbMaestro TeamWork helps to complete a DevOps solution by professionally, accurately and efficiently covering the database tier."

ExtraHop

The ExtraHop platform provides stakeholders across these phases with a single view into real-time application performance that is not dependent on application instrumentation or agents, which require maintenance and configuration.

Design: In the design phase, application architects use ExtraHop to compare the performance of proposed changes in the application design against historical performance baselines. Operations teams can identify performance bottlenecks and other problems that must be fixed in application design. Additionally, Operations teams can validate whether applications can be migrated from dedicated hardware to less expensive virtual environments while still maintaining the same performance characteristics. ExtraHop provides application response times, which can serve as KPIs and baselines when performing these physical-to-virtual migrations.

Develop: During development, engineers use ExtraHop to see how their code performs in staging environments. "We set up the ExtraHop system in our staging environment so that the Engineering team can see the impact of new code against our baseline performance. With visibility across all tiers of the environment, we can determine whether a performance problem is due to infrastructure, misconfigurations, or possibly a code-level issue."

Test: ExtraHop provides application owners and developers the opportunity to catch potential misconfigurations and other issues during the testing phase. One customer used the solution to identify and fix database performance issues related to indexing and poorly written application queries before deploying its new warehouse management system to production, for example. The same company also identified DNS issues that were caused by servers misconfigured to use IPv6 instead of IPv4. Another customer uses the solution to test applications pre-production, and in one case spotted a particular slow call was generating a 30MB response that ultimately the application was not using.

Release: ExtraHop enables Operations teams to mitigate the risk of rollouts by quickly detecting problems and pointing to their source, such as missing indices, new access patterns, and storage contention. After an application rollout, a financial services customer used ExtraHop to identify the cause of unexpected poor performance for certain functions. ExtraHop showed a large increase in database errors, including the error messages that indicated a database schema change as the root cause. Typically, obtaining this type of information would require running database profilers, which are not usually run in production.

Manage: ExtraHop is particularly important in production because it can monitor the performance of all tiers of the application without imposing overhead. Alaska Airlines uses ExtraHop to continuously monitor the performance of its Informix database, helping to ensure that its aircraft will not be stuck waiting on the tarmac. "Our DBAs could either run SQL traces on each database, which would be like shining a spotlight on a small section of a highway, or they could use the ExtraHop system, which is like lighting the entire highway," says Drew Garner, Director of Architecture Services.

Assess: ExtraHop provides the historical performance baselines that are important when trying to identify areas for improvement. MedSolutions uses ExtraHop to support its continual service improvement efforts. This is what Joseph Steele, VP of IT Infrastructure at MedSolutions says: "We were looking for a solution that would provide real-time visibility into performance across tiers as well as historical measurements that would enable us to track our progress. ExtraHop was the only product we evaluated that could easily capture application traffic as it crossed through our converged network, load balancers, and vast numbers of application and database servers."

IBM

IBM provides an open lifecycle and service management integration platform called Jazz that supports the DevOps lifecycle. The Jazz platform enables collaboration, information sharing, and is based on Open Services for Lifecycle Collaboration (OSLC) open standards that enable information sharing with non-IBM vendor-based and open source tools.

Assess: IBM Rational Focal Point: Helps delivery teams with business planning to make the right decisions to deliver business, customer and market value. Captures, compares, contrasts, and visualizes key stakeholder input such as cost, strategic alignment, revenue potential, ability to execute, etc. to analyze impacts of various decision scenarios to optimize product and portfolio plans. Also supports the governance of development outsourcing.

IBM Rational Requirements Composer [RRC]: Empowers teams to define, manage and report on requirements in a lifecycle development project. This Web-based application supports iterative, waterfall and agile-at-scale development methodologies using lightweight requirements processes. (IBM Rational DOORS for product engineering scenarios.)

Design: IBM Rational Software Architect [RSA]: Provides visual service modeling and deployment planning and modeling allowing teams to plan, collaborate, and validate the operational characteristics of their application and services. RSA includes extensions for model-driven generation of deployment workflows that automate the deployment of the desired state topology. The RSA Design Manager feature enables collaboration between Development and Operations on the overall architecture and design of the application and its deployment via the Web by connecting to the Jazz-based Collaborative Lifecycle Management solution (IBM Rational Rhapsody for product engineering scenarios.).

Develop: IBM Rational Team Concert [RTC]: Integrated agile team collaboration and management including work item management, development planning, build management, source code management (SCM), dashboarding, and reporting. RTC provides tooling that enables the shared planning and tracking of tasks across Development and Operations as well as a single source of truth in the source code management system for application and infrastructure code. RTC improves auditability of authorized changes to each and every environment for an application.

IBM Rational Developer for the Enterprise: A family of Integrated Development Environments (IDEs) based on the Eclipse workbench. Provides productivity enhancing tools to quickly design, construct, and validate code changes for multi-tier, multi-channel applications that support a wide variety of client devices and influence a wide variety of back-end systems [JEE, Mobile, Mainframe, IBM i, etc.]

IBM SmartCloud Continuous Delivery [SCD]: Defines automated delivery processes to quickly build, deploy, and test incremental changes to ensure operational characteristics are validated and tested incrementally throughout the development cycle. SCD includes an artifact library to track and govern the deliverable artifacts to ensure that only the right deliverables are deployed into the correct environments. Developers can easily deploy private environments with their isolated changes promoting more frequent exploratory efforts. SCD provides integration with existing open-source tools such as Chef, Jenkins/Hudson, JUnit, Cucumber, and Selenium.

Test: IBM Rational Quality Manager [RQM]: Allows quality professionals to manage test cases while enabling them to plan, track, share, and report on quality results in real-time across automated and manual testing tasks.

IBM Rational Test Workbench [RTW] (with IBM Rational Test Virtualization Server [RTVS]): Records and replays test scripts spanning functional, performance, and integration test scenarios. RTW also provides a test virtualization capability via integration with the Rational Test Virtualization server. RTW recorded tests can be converted to virtual service models, then shared in the RTVS environment, allowing teams to leverage simulated application dependencies to construct test environments in less time and with lower test lab infrastructure costs.

IBM SmartCloud Continuous Delivery [SCD]: Continuously test application changes using reliable and repeatable processes to quickly provision and configure new test environments on virtual platforms such as VMware or KVM. SCD has out-of-the-box integration with RQM and RTW to automate the execution of a test suite and its test cases to incrementally validate the function and operational characteristics of application changes and present the feedback to the development, test, and operations teams.

Release: IBM Rational Automation Framework: Automates middleware environment configuration, administration, and deployment of application and related artifacts for WebSphere middleware. Reduces cost, increases productivity, and improves application delivery velocity through repeatable, consistent, and auditable automation processes.

IBM SmartCloud Continuous Delivery [SCD]: Enables operations teams to save time and cost of provisioning and configuring pre-production environments using virtualization and cloud pattern technology. SCD governs and enforces environment configuration consistency across development, test, and staging environments. Includes deployment automation capabilities to quickly update configurations across multiple environment deployments from a single point of control. Includes real-time dashboarding of quality information across development and test deliverables to facilitate readiness monitoring for production release of application changes.

Manage: IBM Smart-Cloud Application Performance Management: Manages and monitors application and infrastructure performance trends across traditional IT, virtualized, cloud, and hybrid environments. Gives operators the ability to quickly determine the root cause of critical cross-platform application incidents, while meeting demands to develop new applications and modify existing ones. Improves the visibility, control, and automation operators need to manage complex and interdependent applications throughout the enterprise using embedded instrumentation and analytics capabilities.

IBM SmartCloud Control Desk: Unified asset and service management software provides a common control center for managing business processes for both digital and physical assets. IT Infrastructure Library® (ITIL®)-compliant software is accessible through mobile devices and integrates with social media and development tools. Integrates service request management, change/configuration/release management, asset lifecycle management, and service catalog capabilities.

IBM Tealeaf: Supports continuous customer feedback and optimization by providing visual evidence and full context to analyze customer behavior, pinpoint customer struggles and understand customer experience using Web or mobile applications.

Nolio

Test: As organizations embrace agile development and continuous integration, the need for greater testing throughput increases. Nolio AROS isn't a testing tool, but it is used to create test environments (including parallel and dynamic cloud test environments), invoke testing scripts (creating the environments and choosing the appropriate tests based on the specific components that make up the release candidate) and collect test results. Nolio can also be used for automated promotion of deployments from environment to environment based on test results.

Release: Nolio AROS is an operational solution for delivering application releases to users quickly and flawlessly. Nolio is the operational hub tying together tools and processes (e.g. CI tools, ALM tools, CM tools) used for a release while automating the release deployment processes themselves. Nolio's graphical deployment language makes it easy to define reusable deployment processes that are based on the application's model and its release data model. Nolio defined release processes are abstract enough to be reusable across environments and releases, but specific enough to effectively and efficiently automate application deployment. Nolio's dashboard provides release and application level reporting and tracking.

Manage: Nolio is an operational, not management tool; however, there are many cases where emergency procedures or hot fixes are applied directly to production that need to be operationalized. These corrective and preventative operations (COPO) are critical to the daily maintenance of production environments, but are not a part of a standard release. Nolio supports these COPO releases by enabling them to be defined using the application model (as opposed to tied directly to a specific topology) and providing a release manifest that provides real world production information about the COPO release back to Development and QA for release optimization.

OpTier

Design: OpTier APM & BDA value propositions: Out-of-the-box solution to collect end-to-end transaction data from production/real usage data. Simplify the time to intelligently answer design questions from several months to just weeks. Advanced customer segmentation and analytics; understand who your users are, what they do, which features key/important users use, etc., IT investment/project analytics; understand the projected use, adoption, or ROI of planned features. Monitor business transactions and conversion rates (order to cash), understand factors that affect or may improve conversion.

Develop: OpTier APM value propositions: Develop with an eye for production by understanding end-to-end transaction performance and 100% of all real user usages from the production environment (OpTier runs in high scale production environments at 1%–2% overhead). Understand the topology/dependencies between the different tiers of your application, and how those tiers interact on behalf of a user or business transaction. A "chatty" transaction (one that calls the database, for example, a thousand times) may not be a problem from the database or developer perspective, but when rolled into production the exponential use of the transaction by real users could cause the database to fail. Understand the application code– OpTier provides method level diagnostic information for both production and testing environments (at low overhead 1%–2%).

Test: OpTier APM value propositions: End-to-end transaction visibility: Traditional testing with legacy load testing solutions can be significantly improved by determining the bottlenecks associated with transaction performance issues. Which server, what part of the code, which SQL statements, Web service calls, etc. Testing teams often have limited visibility and expertise, and spend significant amounts of time diagnosing the root cause of performance issues during test. Code and SQL in context: Understanding the performance of the application code and SQL database calls in the context of load testing and the business processes being tested provides more granular and powerful context for problem isolation and improvement. Architecture and transaction design/diagrams: OpTier automatically generates a topology diagram of the tiers and transactions traced. This can lead to powerful visibility into poorly designed transactions. In their rush to release a component of functionality, a developer may write an inefficient piece of code. Individually that may not be a problem, but at a higher level, the frequency of its use in business transactions (or in production load) could cause significant strain on back-end resources. OpTier identifies poorly performing transactions as part of the out-of-the-box Service Optimization features. Customers have shown significant improvements in scalability and performance (and even saved hundreds of thousands of dollars of hardware investments) by identifying and tuning key transactions in the application. Visibility to real-world production usage: One of the big challenges with testing is making sure the scenarios you test are an accurate representation of the production load. Because OpTier is lightweight and can be run in production, it can tell you exactly what the production mix of transactions are.

Deploy/Release: OpTier APM value propositions: Production monitoring at low overhead: The ability to collect end-to-end transaction data. Real/production comparisons: OpTier can be used to compare the performance of different versions of code in production. Is the Product Search performing better or worse with the latest release? Track resource utilization/CPU: In addition to tracking the performance of the application, OpTier also shows CPU usage of real users/transactions. Is the performance and CPU utilization better or worse? What problems can we identify and tune to increase performance and capacity?

Manage: OpTier APM & BDA (for advanced real time dashboards) value propositions: Horizontal and vertical APM: OpTier is known for the ability to track end-to-end business transactions to more quickly and effectively identify the root cause of issues. Unlike with vertical/silo monitoring, OpTier traces all real-user transactions and explicitly tells you where the bottlenecks are. Drilldown into the problematic tiers, KPIs, code hot spots, and SQL statements are also available in the tier and transaction context.

Improve operational efficiency: One significant challenge with application management is a lack of visibility and communication between the different teams and tools used to run the production application. As a horizontal solution that provides insight into the user and transaction performance, OpTier helps reduce the time and number of people involved in troubleshooting a problem. Being able to quickly identify the location/tier associated with a problem saves time and resources. Business/user focus and impact analysis: As a user/business transaction focused solution, OpTier provides a context for troubleshooting problems affecting specific users or transactions. Operations deals with hundreds of alerts at any given moment, so being able to quantify the business impact of problems provides a context for prioritizing resolution efforts. Real-time transaction and business monitoring: OpTier BDA provides the ability to capture, visualize, and report on the real time performance of both transactions and business processes. A "quote-to-cash" process can be identified as the individual steps of a process, and fall-out or conversions issue can be detected, alerted, and dashboarded.

Assess: OpTier APM value propositions: OpTier provides valuable information for each phase of the DevOps cycle and for each stakeholder. This data represents a powerful, 360-degree view of the application. The ability to improve communication between teams (within a single tool), focus on users and business transactions, dashboard business performance and transaction conversions, and report/trend/compare data over time provides a powerful solution to assess the entire application.

OutSystems

Design: We provide support for this phase in multiple ways. First we provide a sizing tool that takes "epic" user stories and generates accurate project estimates. While fixed projects are not agile in nature, it is still how most corporate IT teams operate so we help automate this effort. Second, we provide a working method that leverages strong design concepts at the beginning of a project. An example is the delivery of a vision document that has proven to be critical to deliver the right application the first time.

Develop: Here we provide our IDE, Service Studio to develop your applications very fast. We cover all application layers: Process, Logic, User Interface and Data. In addition, as we are a model-driven development platform, you work at a much higher level of abstraction allowing the team to focus on addressing the business problem vs. writing low-level code. Our code generators have been optimized and generate native C# or Java, plus the needed HTML, CSS, Java Script, DDL and implementation scripts to deploy working applications with all nonfunctional requirements built in. Every application has security, scalability, audit trials, documentation automatically handled.

Test: We don't provide any built-in testing capabilities as there are very good products on the market and our applications are standard apps (no interpreters, etc.) so they work with existing tools. We do have several testing components delivered via our Community Forge for specific testing tasks like Unit Testing or Regression Testing. Where we do help from a testing perspective with our own technology is with our Embedded Change Technology (ECT). ECT provides a mechanism for application users to submit feedback directly from the running application. The platform automatically captures the screen image, the specified change/error log from the user and application heuristics (browser type, window size, date, time, user information). ECT makes it easy to submit feedback and removes ambiguity.

Deploy/Release: Here we simplify the complex handoff between Dev and Ops. Our LifeTime console makes staging from one environment to the next (Dev to QA to Production) simple, low risk and very fast. Our impact analysis engine looks across environments and assesses the impact of planned changes making sure you never break related apps when making changes. The ability for Dev to set up the deployment and Ops to actually run the process is key to supporting most complex DevOps team processes.

Manage: Our Service Center console lets you understand your application configuration per environment and manage everything from batch processes to application performance.

Assess: By supporting an Agile process we make it easier for the business to assess the working application, suggest changes and improve the application in a matter of days, changing the way IT responds to the business.

Serena

Design: Serena Business Manager (SBM)
Serena Dimensions RM: Included in Serena Requirements Manager
Serena Demand Manager
Serena Request Center
Serena Requirements Manager

Develop: Serena Business Manager (SBM)
Serena ChangeMan ZMF
Serena ChangeMan ZMF Client Pack
Serena Development Manager
Serena Dimensions CM (Included in Serena Development Manager)
Serena PVCS

Test: Serena Business Manager (SBM)

Release: Serena Business Manager (SBM)
Serena ChangeMan ZMF
Serena ChangeMan ZMF ERO
Serena Dimensions CM
Serena Release /automation: Included in Serena Release Manager
Serena Release Control: Included in Serena Release Manager
Serena Release Manager
Serena Release Vault: Included in Serena Release Manager

Manage: Serena Business Manager (SBM)
Serena ChangeMan SSM
Serena Service Manager (SSM)

Assess: Serena Business Manager (SBM)
Serena Dashboard

Xebia Labs

Test: With Deployit, users can trigger tests.

Release: Deployit fully automates the release of application deliverables (binaries and configurations) to target environments. Deployit allows you to transition/promote applications through your "delivery pipeline" (Dev, Test, Acc, Prod etc.) with role-based access control governing which users are allowed to deploy to which environments. Out-of-the-box content packs for all common enterprise middleware stacks mean no more custom scripting or workflow creation. Integration with build and Continuous Integration tools on the dev side, and Provisioning and Cloud Management tools on the Ops side gives users a complete Dev-to-Ops automated delivery platform composed of industry-leading best of breed solutions.

Manage: Through Deployit's Release Dashboard, Development and release managers gain an instant "live view" of which applications are running in which environments and can track the progress of each version through the delivery pipeline on the way to Production. Release conditions linked to environments act as gates that prevent deployments from being carried out until all relevant requirements have been met. Integration with Change Management tools such as ServiceNow, Remedy, and JIRA allows these conditions to be automatically verified (Deployit checking in the CM system) and marked as completed (CM system updating Deployit).

About Enterprise Management Associates, Inc.

Founded in 1996, Enterprise Management Associates (EMA) is a leading industry analyst firm that provides deep insight across the full spectrum of IT and data management technologies. EMA analysts leverage a unique combination of practical experience, insight into industry best practices, and in-depth knowledge of current and planned vendor solutions to help its clients achieve their goals. Learn more about EMA research, analysis, and consulting services for enterprise line of business users, IT professionals and IT vendors at www.enterprisemanagement.com or blogs.enterprisemanagement.com. You can also follow EMA on Twitter or Facebook.

Corporate Headquarters:
1995 North 57th Court, Suite 120
Boulder, CO 80301
Phone: +1 303.543.9500
Fax: +1 303.543.7687
www.enterprisemanagement.com
2650-Summary.042213