Ovum Decision Matrix: Selecting a DevOps Release Management Solution, 2016–17

How enterprises can improve their software application delivery
Summary

Catalyst

DevOps has grown to become a common item on the agendas of CTOs and CIOs since starting in 2008 as a grassroots movement. There are several reasons for this. To many organizations, DevOps is the continuation of agile in development to agile in IT operations. For others, DevOps is an opportunity to refresh an agile transformation that has gone stale, or to attempt an agile transformation that never even started. To yet others, DevOps is an opportunity to improve application delivery throughout the organization, irrespective of the development processes in place, traditional, agile, hybrid, or custom (a multi-speed project environment). DevOps originated in IT operations and was very quickly embraced by all IT stakeholders and has reached the mainstream far quicker than the agile movement did, of course helped by agile paving the way.

This report focuses on the automation aspects of DevOps, DevOps Release Management, by which we mean release management, orchestration, and automation. Vendors differ in what they offer in one product/box or split between multiple products. This report compares solutions from the leading release management vendors in the market. We also explain what DevOps means for those unfamiliar with the concept. Finally, we look at the evolution of DevOps and the technologies it is pushing forward, principally microservices architecture and the use of virtual containers.

Ovum view

Ovum’s definition of DevOps

There is no accepted definition of DevOps. In this report we define DevOps as an IT movement that encompasses agile principles (the agile manifesto) and encourages automation in application delivery as a way of reducing errors due to manual steps, as well as breaking down silo walls and improving communication and collaboration between all IT stakeholders within a business or organization. This last aspect, the cultural dimension to DevOps, is possibly the one given the least attention but it is often the most important. Without addressing the need to transform existing culture into a DevOps environment, the rest of DevOps may become an aspiration with no realistic chance of success.

The DevOps culture encourages staff engaged on a project but who reside in separate organizational silos (or who are in the same silo but not co-located) to meet face-to-face in workshops, meetings, and one-to-one discussions, as well as participate in agile retrospectives. The aim is to facilitate collaboration and avoid the type of impediments to progress that exist in traditional organizations, such as passing around heavy documents that might never get read, can be error laden, and become quickly out of date.

DevOps automation

This report is focused on the part of the application lifecycle management (ALM) called release management and automation. With every IT wave, and DevOps is no exception, there is a rush of startups to provide tooling, as well as existing specialist vendors adapting to the new trend. With DevOps, when in 2010/11 it looked like it had a good future ahead of it, the large IT players made acquisitions. The result is that some of the established ALM vendors offer release management solutions, but there are also vendors dedicated to this activity, including new players in the market.
The DevOps release management solution landscape is shown in Figure 1. Note that some vendors provide a single solution for the layers in bold in Figure 1, whereas others split “release management and orchestration” and “release automation” into two solutions.

DevOps release management solutions sit within an ecosystem of additional tooling with which they typically integrate (see Figure 1). Continuous integration is dominated by the open source server Jenkins but there are many others on the market. Build tools sit below this layer (popular Java tools include Ant, Maven, and Gradle). Test tools are too numerous to mention. Finally, popular provisioning tools include Red Hat Ansible, Chef, and Puppet.

**Figure 1: DevOps Release Management Landscape**

<table>
<thead>
<tr>
<th>Development and Testing</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALM</td>
<td>ITSM / ITIL</td>
</tr>
<tr>
<td>Release Management &amp; Orchestration</td>
<td></td>
</tr>
<tr>
<td>Continuous Delivery</td>
<td></td>
</tr>
<tr>
<td>Continuous Integration &amp; Testing</td>
<td>Release Automation</td>
</tr>
<tr>
<td>Performance Testing</td>
<td>Performance Monitoring</td>
</tr>
<tr>
<td>Integrations</td>
<td></td>
</tr>
<tr>
<td>Build Tools</td>
<td>Test Tools</td>
</tr>
</tbody>
</table>

Source: Ovum
Continuous delivery (CD) is central to DevOps Release Management, and involves creating a workflow pipeline that spans development, QA/ testing, and IT operations (Figure 2 shows a schematic of the CD pipeline). The idea behind CD is to allow developers to change some source code, and if the business so desires it, to be able to push a button and see this change implemented in production. Organizations are typically on a journey of reducing the cycle time it takes to complete such an action on their CD setup. If they can reduce a cycle time that used to take six months and reduce it to one week, this would be a major achievement. Internet companies such as Netflix and Amazon make many hundreds of changes per hour. The reason the most advanced implementations of CD are so rapid is that these companies build a software architecture that is designed for change.

Operations staff are able to delegate responsibility of application delivery to the release management tools, which have all the checks and audit trails required for compliance and IT governance, instead of relying on scripts that require manual steps and are error prone. This frees up operators and sysadmin to perform other duties, while providing them with monitoring information and alerts on application releases. Some vendors promote fully automated testing within CD without suggestion of any manual or exploratory testing. Ovum advises caution on this because there is still a role for professional testers.

Finally, note Figure 1 shows testing and operations as being two separate environments, but this is an “old school” view. Visionary and early adopter organizations pushing the boundaries of microservices architecture have extended testing into production (see Chaos Monkey, Simian Army, and related concepts).
The future of DevOps

Release management is part of ALM, and Ovum expects a new generation of ALM products to appear that offer a single, holistic, and integrated box, comprising an ALM solution with a full DevOps Release Management capability.

There is also potential for DevOps release management solutions to integrate with a number of tools used in the IT operations space, such as capacity planning. While systems administrators can liaise with developers and synchronize application releases with planned infrastructure changes, the DevOps approach is to automate where possible, and this creates an opportunity for release management tools to integrate with capacity planning tools. For vendors that have both types of tooling, this is an opportunity to differentiate their offerings on the market.

Agile and DevOps practices are also influencing software architecture with microservices architecture and the use of containers (lightweight virtual environments such as Docker), and this is creating a new IT wave that is starting to grow and we believe will dominate software architecture thinking in the decade ahead. In a nutshell, the primary reason is that for DevOps-style continuous delivery to work as intended matching software architecture is required that allows changes to be made to a live system without having to bring it down. Virtual containers such as Docker are becoming popular for microservices because they allow for good separation of dependencies (a key aspect of microservices), and they are lightweight so many (hundreds, thousands) can be generated in an environment and tend to be used as runtime objects. Unlike virtual machines that persist as part of the infrastructure, containers pop up and down mostly during runtime and therefore conserve resources.

The question of application security is uppermost in IT due to the responsibility for security passing from the network firewall to the application itself, because these commonly traverse networks, from web and mobile to data centers. Developers have historically not given application security sufficient attention despite the software security development lifecycle (SSDL) concept being a decade old. We therefore see a need for ALM tools, including DevOps release management solutions, doing more to support the SSDL. In this context, DevOps practitioners have introduced the concept of rugged software and rugged DevOps. Where DevOps is about bringing down silo walls, two sets of teams that still need to better engage are developers and security experts. Another concept is that testing for security weaknesses should be part of the continuous testing in the delivery pipeline. Expect to hear more about Rugged DevOps in 2016-2017.

Key findings

- DevOps is on CIOs’ and CTOs’ agendas and has reached mainstream awareness.
- Agile and DevOps are complementary but DevOps is beginning to be understood as a set of ideas and principles that can be implemented across the whole enterprise.
- Organizations typically have projects that deliver applications at varying cadences, depending on the use of traditional, agile/lean, custom, or hybrid processes, leading to what is called a multi-speed project environment.
- DevOps is also being applied to multi-speed project environments, not only agile ones.
- DevOps release management solutions typically comprise features that cover release management, orchestration, and automation, and continuous delivery.
• Many DevOps release management solutions rely on integration with continuous integration servers, build tools, and provisioning tools, particularly free open source tools.

• The most influential automation tool to arise from DevOps is continuous delivery (CD). CD is a core part of a DevOps release management solution.

• The adoption of agile development and DevOps and the desire to achieve rapid changes in production (when the business requests it), and is driving the use of microservices architecture (MSA) and containers.

• As MSA and container adoption grows, it will disrupt the release management solution market, but vendors are generally ready for this, with support for Docker already quite common.

• Rugged DevOps is being given more attention, and is about raising awareness of the security issues that exist in application releases and the tooling.

Market and solution analysis


The definition of what goes into a DevOps release management solution is depicted in Figure 1. Ovum favors single holistic boxes that perform release management, orchestration, and automation, including continuous delivery, and out-of-the-box integrations with the rest of the release tools ecosystem, including development, test, build, provisioning, and more. Some vendors prefer to split their solution into two parts (release management and orchestration, and release automation).
This ODM has been closely contested. As can be seen in Figure 3, the vendor solutions, apart from two challengers, are all clustered together with technology scores over 80%. We therefore decided to award all vendors in the cluster leader ranking. The solutions do differ in how they approach release management so some may be more suitable for some organizations than others.
Ovum ratings

The DevOps Release Management vendors are ranked into three types as follows:

- **Market Leader**: This category represents the leading solutions that we believe are worthy of a place on most technology selection shortlists. The vendor has established a commanding market position with a product that is widely accepted as best-of-breed.

- **Market Challenger**: The solutions in this category have a good market positioning and are selling and marketing the product well. The products offer competitive functionality and good price-performance proposition, and should be considered as part of the technology selection.

- **Market Follower**: Solutions in this category are typically aimed at meeting the requirements of a particular kind of customer. As a tier-one offering, they should be explored as part of the technology selection.

Vendor analysis

Serena Software (Ovum recommendation: leader)

**Products**

The primary products are Serena Deployment Automation, and Serena Release Control.
Ovum SWOT assessment

Strengths

Serena has evolved its enterprise application delivery solution to the needs of DevOps

DevOps is transforming how enterprises are delivering their applications into production. DevOps thinking has spread from visionaries/early adopters to the early majority without facing any chasm in-between. This has created a large market for agile and DevOps release management tools and Serena has produced a rounded solution to meet these needs, performing outstandingly well in the ODM and positioned in the leader category.

Serena is able to leverage its large enterprise presence in the IT market

Serena is a trusted name in the large enterprise space and is able to leverage this presence for the benefit of its release and deploy management solutions. Highly regulated industry sectors such as financial services, insurance, healthcare, aerospace, defense, and government/public sector need IT solutions that are compliant. In addition, these organizations have a mix of development work styles for building new systems and maintaining legacy applications. The typical large enterprise works at multiple delivery speeds and may blend traditional waterfall with agile/lean and regulated development processes as necessary. Serena’s tools support DevOps across the whole organization at the pace and degree of agility a project requires.

Weaknesses

Serena’s solutions could improve the out-of-box integrations

Serena did perform well in the integration section of the ODM technology dimension but there were a few areas that could be improved. Connectors to application security tools, such as security code scanners (static and dynamic) are lacking. There are no connectors to IT infrastructure capacity-planning tools and this omission is common across all the vendors in this ODM review, so correcting it will put Serena at an advantage. There is also no connector to mobile application development platforms.

Opportunities

Agile development and DevOps are pushing the adoption of microservices and containerization

As enterprise development becomes more agile and releases frequently to production, assisted by DevOps style continuous delivery, the next bottleneck is the software architecture itself. This is where microservices architecture (MSA) steps in. An evolution of service-oriented architecture, MSA allows a change to production while the system is live. Application delivery now means swapping a microservice, which can be done quickly and with minimum down time. There is a huge opportunity for DevOps release management vendors to support MSA and also container technology, such as Docker and Rocket, which is often deployed with MSA. At time of writing, Serena announced updated Docker support with a plugin for Serena Deployment Automation, so this is a move that will open up opportunities for the company.

Threats

As a private company Serena is able to reinvent itself and address new market opportunities
Serena was acquired by private equity firm HGGC in 2014 in a bid to redefine the company direction and be sheltered in this attempt without the intrusion of typically short-term views on Wall Street. In 2016 at the time of writing the company was acquired by Micro Focus. DevOps is one major new direction for the company and this was a shrewd move as we see DevOps hitting the mainstream in 2016-17. The growth opportunities for DevOps are also exciting, Ovum sees plenty of evolution potential in DevOps. The current growth area is in MSA and containerization, but there is also potential for improved integration with ALM on the dev side and improved integration with operations and IT business services. Serena is in a highly competitive market and has a large enterprise focus. However, large enterprises are now more accepting of the open source software movement and are always looking to reduce costs. Serena may well need to reduce its margins to compete effectively, and also address the larger SME market.

Vendor solution selection

Inclusion criteria

Ovum drew up a shortlist of vendors to invite participation of the ODM project based on its experience of the market. The shortlist was designed with the objective of capturing the largest players in the market as well as the most innovative and fastest growing companies. The general inclusion criteria are given below but discretion was made to ensure the above objective was met.

Ovum assessed vendor inclusion on:

- A minimum revenue threshold based on licenses, or, combined licenses and consulting threshold, or threshold number of customers.
- Vendor does not focus on DevOps in a single vertical industry.
- Vendor solution should include support for at least four core functional areas in release management and automation, listed in the technology methodology section below.

Methodology

This research is based on the considerable depth of research Ovum analysts have accumulated in the software lifecycle management space. Participating vendors were asked to complete a comprehensive spreadsheet for the three dimensions of the ODM. Briefings and product demos followed and the various parts of the research were reviewed with the vendors before signing off. The three dimensions of the ODM are explained in turn.

Technology assessment

In this assessment dimension, Ovum analysts develop a series of features and functionality that would provide differentiation between the leading solutions in the marketplace. The criteria groups identified for DevOps release management are given below, with section weight percentage. Note: the main sections are listed with the sub-row headings, but the sub-sub-rows are omitted.

General features, 18%

- Is the product available as a cloud service?
- Is the product available on-premise?
• Is the solution accessible over mobile for dashboards and operations (like approvals/transitions etc)?
• Control solution by
• Solution repository
• Reporting
• Dashboard
• Deployment targets
• Deep ALM integration allowing ALM traceability from requirements to deployment
• Release management process agnostic
• Capture (deployed application) end user feedback
• Build management

Application Manifest, 6%
• Define applications
• Add unpackaged software
• Add packaged software
• Define application configuration requirements
• Advanced integrations with databases and application servers
• Define dependencies between applications

Workflow/process design, 10%
• There is an embedded workflow engine/system
• Visual design and modeling of workflow/process
• Workflow/process design
• Parametrized workflow templates
• Design by scripts
• Content library of prebuilt workflows/processes for common processes
• Select actions, for example, for installation, verification, and so on
• Script-less interface for defining actions
• DSL with visual interface (drag and drop)
• DSL with command line interface
• Design workflows/processes that are environment and application independent
• Testing of desired configurations and pipeline based on model
• Deployment job duration estimation
• Deployment success rate prediction

Workflow/process execution, 10%
• Create parallel workflows/processes
• Define multi-tier processes, including versions of tiers
• Create programmatic loops and conditional actions
Real-time status visibility of process steps
- Manual start, pause, and stop processes
- Automated actionable rules based on real-time analytics
- Can change the planned release workflow/process during a live release
- Process rollback
- In-built scheduling
- Intelligent deployment: actions that do not need to be executed are not
- Context-sensitive deployments: automatically tailor actions based on current state of target environment
- Automatically ripple changes to applications in a deployment change

**Topology, 8%**
- Discover existing application topologies
- Detects the differences between what is about to be deployed and what is already deployed
- Visualization of the target environments
- Integrates with a CMDB to learn about assets to automate
- Pulls versioned deployable artifacts from an asset database (an ITIL Definitive Media Library).
- There is integration between the deployment engine and the artifact repository
- Includes an integrated artifact repository
- Define configurations for database, system, network etc
- Define private/public/hybrid cloud, virtual environments
- Define configuration property mappings/dependencies between tiers
- Infrastructure as code capabilities (managing, testing, and deploying infrastructure as code from dev through production)

**Solution management, 14%**
- Deployment agent
- Workflow/process and topology plans
- Define dependencies between releases
- Pass information between releases
- Version control of plans/workflow/topology/processes
- Capacity planning
- Permissions management/role-based access control compliance
- Fine grained workflow/process steps control with approvals and gates
- Visibility of all approval controls and gates
- Reporting
- Reporting API
- Collaboration: IM, wiki/knowledge base, automatic email alerts
- Logging, scheduling, and notifications
Dashboard/console for managing deployment processes
- Covers Windows, Linux, Unix, Mac machines/cross-platform neutrality
- Covers mainframes
- Covers mobile

**Analytics, Governance, Security, 10%**
- Register deployment changes in CMDB
- Audit trail of all actions, changes to releases
- Assign risk warnings or level to a release
- Detect out-of-process changes to operational environment
- Event inspection with drill down to detailed information, e.g. for failure analysis
- Traceability between application and operations assets
- Preliminary validation
- Security
- Real-time data analysis: analytics, logging, and error reporting
- Service level agreement management and service target management

**Development/Operations Integration, 12%**
- Automated continuous integration and continuous deployment launch
- All development and operational assets in a common database
- Audit operational configurations for changes to operational environments
- Continuous deployment (CD)
- Integrations to third party ALM solutions and technologies directly within a deployment workflow
- Production incident/defect tracking & remediation
- Coordination of defect root cause analysis between development and operations
- Shared information on known defects, workarounds, and so on
- Incident/defect service level and performance tracking/reporting across the lifecycle and organizations

**Integrations, 12%**
- Extensible secure web service interface
- Please list tools for which integration exists out-of-the-box

**Execution**

In this dimension, Ovum analysts review the capability of the solution around the following key areas:
- Maturity: The stage that the product/service is currently at in the maturity lifecycle is assessed here, relating to the maturity of the overall technology/service area.
- Interoperability: In this element we assess how easily the solution/service can be integrated into the organization’s operations, relative to the demand for integration for the project.
Innovation: Innovation can be a key differentiator in the value that an enterprise achieves from a software or services implementation, and this is assessed in this criteria.

Product support and deployment: Referring to a combination of assessed criteria and points of information, Ovum analysts provide detail on various deployment issues, including time, industries, services, and support.

Licensing and enterprise fit: Ovum analysts assess the product from a licensing perspective, for example, is it a single product with a common console for all modules. Can the solution be easily downloaded and purchased. Types of licenses available. The alignment of the solution is assessed in this dimension, and the potential ROI period identified.

Market impact

The global market impact of a solution is assessed in this dimension. Market Impact is measured by a formula based on revenue and revenue growth; the final score has a maximum score of 10 and is represented on the ODM by the bubble size. Note revenue is filtered by a ceiling cutoff. The ODM radar diagrams provide additional information on market impact as follows:

- Revenues: Each solution’s global DevOps Release Management revenues are calculated as a percentage of the market leader’s. This percentage is then multiplied by a market maturity value and rounded to the nearest integer. Overall global revenue carries the highest weighting in the market impact dimension.
- Revenue growth: Each solution’s revenue growth estimate for the next 12 months is calculated as a percentage of the growth rate of the fastest-growing solution in the market. The percentage is then multiplied by 10 and rounded to the nearest integer.
- Global reach: Ovum determines local presence in the four regions of the world: North America; Europe, the Middle East, and Africa (EMEA); Asia-Pacific; and Latin America. A score is provided factored to a maximum of 10.
- Partnerships: Ovum estimates the size of each vendor’s partnership eco-system and then factors this to a maximum possible score of 10.

Appendix

Further reading


DevOps Self-Assessment Model, March 2016, IT0022-000600IT.

Assessing an organization’s DevOps maturity, March 2016, IT0022-000601.
Author

Michael Azoff, Principal Analyst, Software Solutions
michael.azoff@ovum.com

Ovum Consulting

We hope that this analysis will help you make informed and imaginative business decisions. If you have further requirements, Ovum’s consulting team may be able to help you. For more information about Ovum’s consulting capabilities, please contact us directly at consulting@ovum.com.

Copyright notice and disclaimer

The contents of this product are protected by international copyright laws, database rights and other intellectual property rights. The owner of these rights is Informa Telecoms and Media Limited, our affiliates or other third party licensors. All product and company names and logos contained within or appearing on this product are the trademarks, service marks or trading names of their respective owners, including Informa Telecoms and Media Limited. This product may not be copied, reproduced, distributed or transmitted in any form or by any means without the prior permission of Informa Telecoms and Media Limited.

Whilst reasonable efforts have been made to ensure that the information and content of this product was correct as at the date of first publication, neither Informa Telecoms and Media Limited nor any person engaged or employed by Informa Telecoms and Media Limited accepts any liability for any errors, omissions or other inaccuracies. Readers should independently verify any facts and figures as no liability can be accepted in this regard – readers assume full responsibility and risk accordingly for their use of such information and content.

Any views and/or opinions expressed in this product by individual authors or contributors are their personal views and/or opinions and do not necessarily reflect the views and/or opinions of Informa Telecoms and Media Limited.
CONTACT US
www.ovum.com
analystsupport@ovum.com

INTERNATIONAL OFFICES
Beijing
Dubai
Hong Kong
Hyderabad
Johannesburg
London
Melbourne
New York
San Francisco
Sao Paulo
Tokyo