White Paper

Application Delivery Management

How to Build a DevOps Toolchain That Scales

A connected and integrated DevOps toolset enables end-to-end communication and collaboration at enterprise scale.

MICRO FOCUS

This paper examines how connected lifecycle management tools enable effective communication at the enterprise level without changing how each team works.

Introduction

What started with one or two small teams doing continuous integration (CI), DevOps has been embraced by enterprises as a means to quickly deliver high-quality customer-centric innovation to customers.

However, organizations that are scaling up their DevOps implementation are finding that their toolsets are large, unwieldy— and disconnected. This bottleneck should be removed early on in your journey to an enterprise DevOps transformation.

Fortunately, you can ensure collaboration and cooperation between your teams and their toolsets without changing your entire tool ecosystem. The key is to connect your tools together in a way that allows them to share information and make work visible, but without changing the way people work with each tool.

Many tools have built-in integrations that let you share data directly with other tools. But if no such integrations exist, you can use connected lifecycle management tools to integrate with them in order to collate and share information.

This paper examines how connected lifecycle management tools enable effective communication at the enterprise level without changing how each team works.

Islands of DevOps

A key tenet of DevOps is to break down walls, remove silos, and form teams capable of everything required to deliver software: developing features, deploying them to production, and maintaining them to ensure high quality and high availability.

Due to the organic way in which DevOps has evolved in many organizations, and the ever-growing number of tools aimed at the DevOps market, it's not unusual to find teams working side-by-side, but using different tools.

Each team manages its own pipeline, maintains its own backlog, and uses its own testing tools, deployment tools and monitoring tools. For many teams, this works well; it makes no difference to them if other teams are using alternative products, as long as each uses the tools that they consider the most effective for their needs.

Why It Matters

While each team works perfectly well on its own, the team members often have to integrate with other teams. When one team delivers a service that is consumed by another, it's not always possible to create a cross-functional end-to-end team that can cover everything from the service's back-end to the user interface on the front-end. The two teams will need to synchronize their work, and be able to share assets and information.

For example, if the consumer team finds a defect in the service, and they can't fix it on the spot, they must log the defect. If the two teams are using different defect tracking systems, it's challenging to share these assets. For one thing, the defect needs to be logged in the provider's system, so that they're aware of it, and can fix it. If the consumer team doesn't have login rights to the provider's system, or even access to the server, there's no way to record the defect, and it won't be fixed. In this situation, the teams will continue to rely on email and other inefficient and ineffective collaboration tools that aren't suitable for this purpose.

While improving and evolving their DevOps practices, organizations sometimes encounter the situation where a single team uses more than one tool for the same thing. It is not uncommon for test engineers to use one tool for managing their backlog and tasks, while the developers use another. This is often a historical artifact of organizations that had a clear division between the QA and the development teams – a strong indicator of Waterfall development – and who are now combining their people within the same team. However, the various specialists continue to use the tools they've used in the past. This results in a situation where even communication within the team is stifled.

The Scale of the Problem

In the 2018 survey of 191 technology influencers and decision-makers, The State of Application Delivery Management in the Enterprise, BizTechInsights reported that 40% of respondents said that their current application delivery management solution does not integrate with other systems. This hampers DevOps, which strives to identify and remove bottlenecks in the delivery pipeline. These organizations have pointed to perhaps the biggest bottleneck of all. And of those organizations that have adopted an application delivery management solution, 38% are dissatisfied with it. Just 21% say their current system adequately addresses their needs.

A simple solution is to pick a single set of tools and enforce them across the organization. Everyone must use the same lifecycle management tool, the same functional testing, performance testing, and security testing tools, the same deployment automation tools, etc. But this is naïve, and the cost of replacing the existing tools with a new set of tools is prohibitive

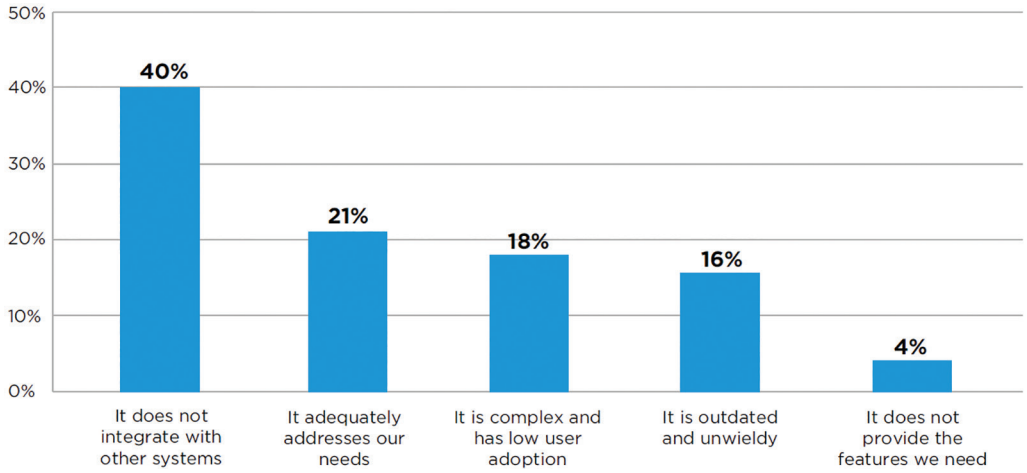

Figure 1. How would you describe your current application development/delivery management solution.

But the good news is that these organizations recognize that this is a problem and would like to do something about it.

So, what can you do?

Connect the Tool Sets

A Simple Approach

A simple solution is to pick a single set of tools and enforce them across the organization. Everyone must use the same lifecycle management tool, the same functional testing, performance testing, and security testing tools, the same deployment automation tools, etc. But this is naïve, and the cost of replacing the existing tools with a new set of tools is prohibitive, because:

- Many organizations are still paying for their existing tools, in terms of licenses and support. There's no guarantee that they'll get a refund if they end the contract early. They'll end up paying for both the existing software and the new software at the same time.
- The effort of disconnecting the existing tools from their DevOps pipelines, and adapting the new software to replace them, can be unexpectedly high in cost and time. Customers are more interested in getting their hands on software updates than on waiting for the team to re-engineer their pipeline.

- The DevOps team should be concentrating on delivering value to the customer. Introducing new tools often means introducing new defects and bottlenecks, as the team learns its way around the new ecosystem.
- Even if a single set of tools is chosen, there's no guarantee that the tools are able to share information and provide teams with the insights that they need in order to track their investments, manage their backlogs, and resolve issues.

If a new organization is forming from scratch, this approach may work (although keep in mind the last point). But how often does that happen? Most large organizations that are moving to DevOps have a long history and investment in tools, processes and procedures. These can only be changed gradually.

Create a Connected and Inclusive System

The better way is to let teams choose the tools that work best for them. If they choose to align with the organization, or even between two teams that work closely together, this is the right time to do it. But they should approach it carefully, keeping in mind the potential costs of re-engineering processes and procedures.

The better way is to let teams choose the tools that work best for them. If they choose to align with the organization, or even between two teams that work closely together, this is the right time to do it.

Ideally, your teams should choose tools that are already integrated and communicating with each other. But that rarely happens. Instead, introduce an integrated and connected, yet loosely-coupled DevOps toolset that adds value by providing the necessary end-to-end communication and collaboration channels, while integrating with the teams' chosen tools. This approach has several advantages:

- Integrated toolsets are typically offered as cloud-based services. There are no set-up costs, no infrastructure costs, and it's easy to register new team members with the tool and train them as the use of the system grows. The tools typically come pre-configured with all of the required integrations.
- The tools in the connected toolset can integrate with the tools that the teams are already using. They don't enforce changes in the way the organization's teams work with their tools. Connected toolsets are also extensible, so you can integrate them with any third party or even custom tools, through open APIs.
- The tools can also replace the team's existing tools if that works better for them. For example, if a team is not happy with its existing lifecycle management tool, or if it isn't using one at all, the integrated toolset will include a lifecycle management tool already designed, engineered and configured for enterprise DevOps teams. As the business evolves, you can swap tools in and out as necessary, which also protects you from vendor lock-in.
- They enable automation of the entire DevOps pipeline, while aggregating data and insights from the whole ecosystem, presenting them in easy-to-use dashboards for all stakeholders to view. Developers can see the status of their pipeline, testers can see what tests are running, executives can track the overall status of the portfolio, and so on — without interrupting the existing processes.

Teams need to have the autonomy to make decisions about the toolsets that work best for them. At the same time, teams need to work with each other, and if they choose tools that don't integrate closely, communication channels within and between teams are hampered.

- There are frameworks available to help organizations as they make their transition from legacy software delivery methods, to agile and continuous delivery through DevOps, at enterprise scale. The Scaled Agile Framework, or SAFe, is probably the best-known. Connected and integrated toolsets often support one or more frameworks out of the box.
  - Note that it is always best to choose the framework, and then your tools, rather than choosing a framework based on your tools. Choose a connected and integrated toolset that can support the framework that will underpin your organization for years to come.

Conclusion

Teams need to have the autonomy to make decisions about the toolsets that work best for them. At the same time, teams need to work with each other, and if they choose tools that don't integrate closely, communication channels within and between teams are hampered.

The most effective solution is to let the teams choose their own tools, and in parallel, adopt a connected DevOps tool set that provides the necessary communication and collaboration channels end-to-end—and that integrates with the tools the teams already use. Each team decides how it prefers to work, while the information it generates is automatically shared with the other teams and the senior managers. The organization can ensure that each team is giving the most critical and strategic investments the correct priority, and is gaining visibility into the status and progress of each item, regardless of what tools it is using.

The Micro Focus DevOps Suite

The Micro Focus DevOps Suite is an integrated set of products that help you govern, test, and manage your software, however you choose to develop it. Supporting modern agile and DevOps methodologies, as well as traditional software development practices, it includes Micro Focus ALM Octane for lifecycle and pipeline management; Micro Focus StormRunner Functional and Micro Focus StormRunner Load for functional and performance testing of web and mobile applications; and the Micro Focus AppPulse suite to monitor and adapt your products' user experience in production.

To learn more about the DevOps Suite, and to sign up for a free trial or demo, go to **http://microfocus.com/devopssuite**

Contact us at:
www.microfocus.com

About Micro Focus

Micro Focus is a global software company with 40 years of experience in delivering and supporting enterprise software solutions that help customers innovate faster with lower risk. By applying proven expertise in software and security, we enable customers to utilize new technology solutions while maximizing the value of their investments in critical IT infrastructure and business applications. As a result, they can build, operate, and secure the IT systems that bring together existing business logic and applications with emerging technologies—in essence, bridging the old and the new—to meet their increasingly complex business demands.

To learn more about Micro Focus, watch this video

Learn More At
www.microfocus.com/devops