

A close-up, low-key photograph of a person's face, focusing on their eyes and nose. They are wearing clear safety glasses. Overlaid on the right lens and the bridge of the glasses is a vibrant, glowing digital circuit board pattern in shades of blue, cyan, and magenta. The person's left eye is visible through the left lens. The background is dark, and the lighting is dramatic, highlighting the contours of the face and the glowing elements.

opentext™

Fit ALM/Quality Center into Your Toolchain

Why do you need a toolchain?

We've seen digital transformation projects accelerate in the past couple of years, leading to an explosion of software. It's no secret that businesses feel the pressure to adapt and respond rapidly to sudden change. That's why DevOps has become a standard practice across industries.

DevOps emphasizes collaboration between development and operations teams. When it's done right, software release cycles take only weeks or days. A DevOps toolchain plays a crucial role in delivering quality software fast. Building one enables you to:

- Automate and orchestrate processes.
- Rely on teams that make fewer mistakes and work faster.
- Release apps and deliver value faster.

As a OpenText ALM/Quality Center customer, you can add the solution to your toolchain and leverage your existing test assets.

Toolchain Considerations

More than anything, DevOps is a cultural transformation. So you need to consider people, processes, and tools when designing your toolchain.

Here are the top five things to consider:

1

End-to-end system

Treat the toolchain as a system. It should connect people working across phases of the application lifecycle: Plan, build, test, release, and operate. It breaks all the silos that once separated these people.

2

Existing tools

You already have some great tools. So building a toolchain without them doesn't make sense. You want to integrate and orchestrate them based on your processes and automate the workflow.

3

Easy to build with low cost

To use your existing tools, you need a custom toolchain for your organization. The cost and time involved are key factors that will affect how you build it.

4

Visibility and analysis

By using a toolchain, you can gather data in real time for analysis. The ability to assess status and identify improvements is indispensable.

5

Adaptive to potential changes

Your toolchain should be able to cope with changing needs, scales, and uncertainties.

Let's see exactly how
ALM/Quality Center works
in your DevOps toolchain.



Connect

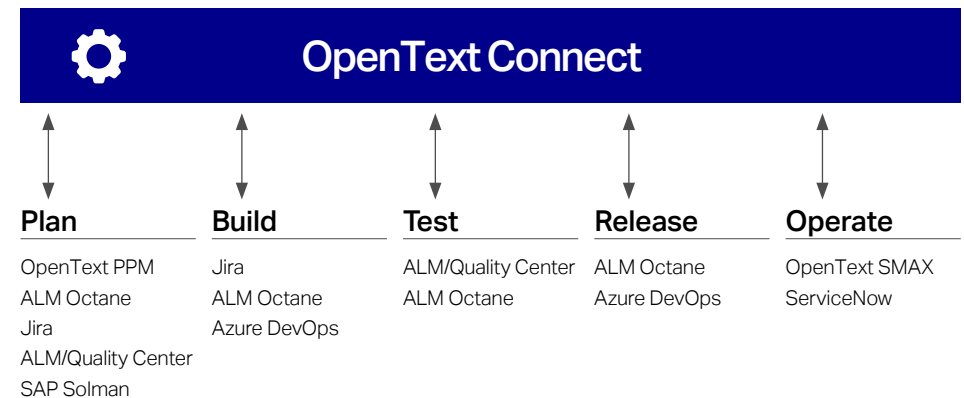
First things first, you need to connect your tools. Each team in the software delivery cycle uses a different tool for their work. Ultimately, connecting your tools connects your people.



But building a custom toolchain with existing tools can be costly and time-consuming. Fortunately, [OpenText Connect](#) can help.

It gives you out-of-the-box connections for the most popular third-party and OpenText tools to form your toolchain. Its low cost and fast implementation are just two more reasons to check it out.

ALM/Quality Center is among the [15+ tools connectable](#) by Micro Focus Connect and fits the test management part well.



Case Study

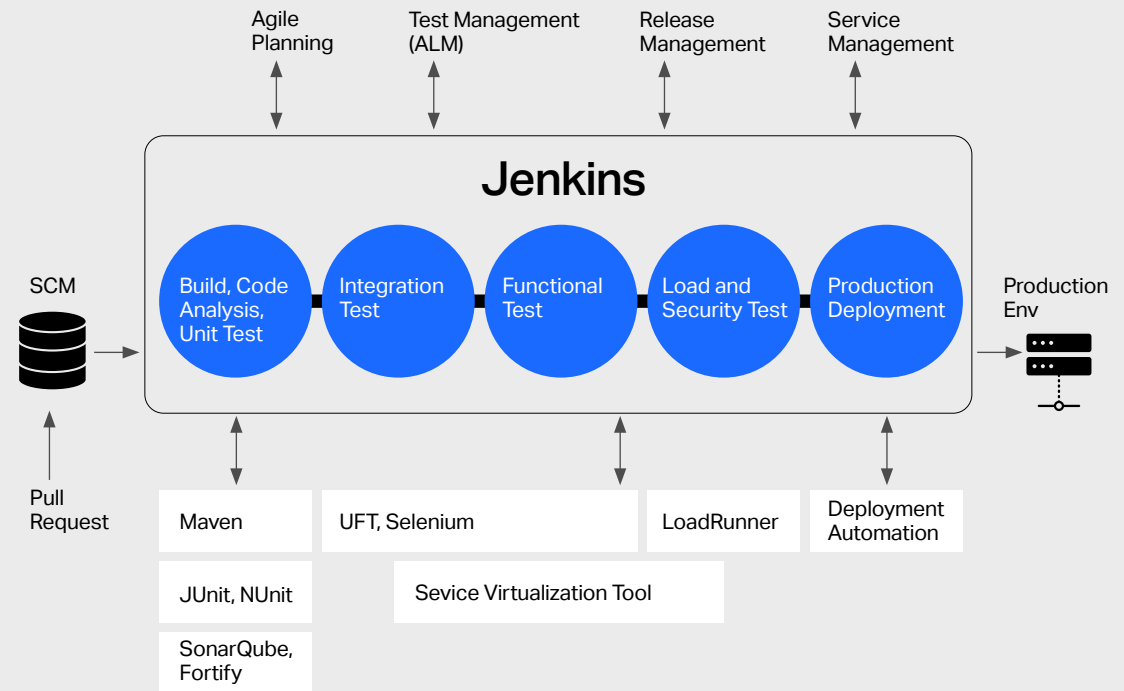
See how [connecting ALM/Quality Center with third party solutions via OpenText Connect enables visibility and traceability in heterogeneous application testing eco-system.](#)

Orchestrate

Next, you'll need an orchestration tool (also called CI/CD tool) to automate the whole process and coordinate tasks.

Jenkins, Bamboo, and Azure Pipelines are popular in this category. These tools are like the engine that powers all the automated parts. Each component you add to the toolchain must integrate with the orchestration tool.

Jenkins is the most widely adopted orchestration tool. ALM/Quality Center—along with many other Micro Focus products—can join Jenkins-driven toolchains through the [Application Automation Tools plugin](#). With its large integration ecosystem, Jenkins can integrate ALM/Quality Center with a wide range of great tools from third-party vendors and OpenText.



Test Management in DevOps

It's tempting to think that test management is no longer important with automated DevOps toolchains. Why use ALM/Quality Center when there are ready-to-use toolsets and all-in-one DevOps products?

In reality, speedy automatic flows make rigorous and efficient quality management even more critical. When testing mission-critical products or services, you need the ability to manage complexity, scale, and regulatory compliance.

Nowadays, major cloud service providers include DevOps toolsets in their cloud environment. Leveraging them—especially the cloud-based pipelines—is very convenient. Meanwhile, you can use ALM/Quality Center to conduct full-blown QA. Take Microsoft Azure DevOps as an example. By synchronizing its bugs and tasks with ALM/Quality Center, you take advantage of both solutions to achieve optimal enterprise-level QA.

Use-case: Syncing ALM and Azure DevOps

[See documentation.](#)

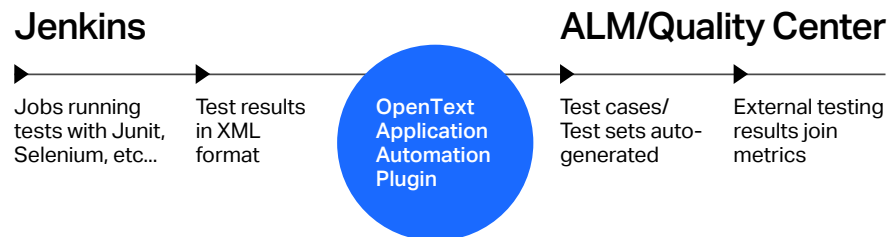


Third-Party Testing Tools

Test automation is key in DevOps. You want to automate large portions of your testing activities and let the CI/CD tool run them according to a trigger or schedule. But automation shouldn't stop with testing. You can also use it to set up your service virtualization environment.

Meanwhile, you need to strike a balance between cost and performance with a mixture of the best open source and proprietary testing tools.

But automation doesn't mean you can sit back and relax. Running automated tests without keeping track of the results and connecting the dots between tests, requirements, and defects is meaningless. These test results must be visible and traceable.



But how?

The Application Automation Tools plugin also processes the results of automated tests run by Jenkins jobs. When the results are sent to ALM/Quality Center, test cases or sets are automatically created. Once set up, you can achieve a high degree of automation. More importantly, these tests become an integral part of your quality metrics.

Upload external test result to ALM

[For OpenText Application Automation Tools Plugin 6.4 and later](#)

Align Dev and QA

Do you use ALM/Quality Center and Atlassian Jira? You're not alone. Many organizations use Jira or another vendor's Agile planning tool. Integrating them using Micro Focus Connect eliminates the communication barriers between development and QA.

You can integrate multiple Agile projects with ALM so that the QA team gets a holistic view of the overall product quality.

Plus, the [ALM Test Management Jira Plugin](#) offers even more convenience. It gives Jira users visibility into the test status and coverage relevant to their development items. Within the Jira user interface, they get immediate feedback on how well their work was tested.

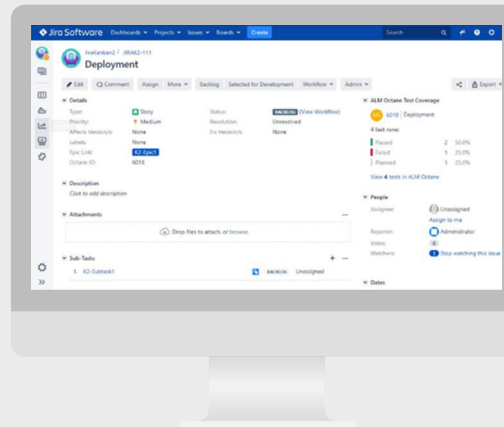
Video

[How to use ALM Test Management Jira Plugin](#)

SEAMLESS DEV AND QA COLLABORATION

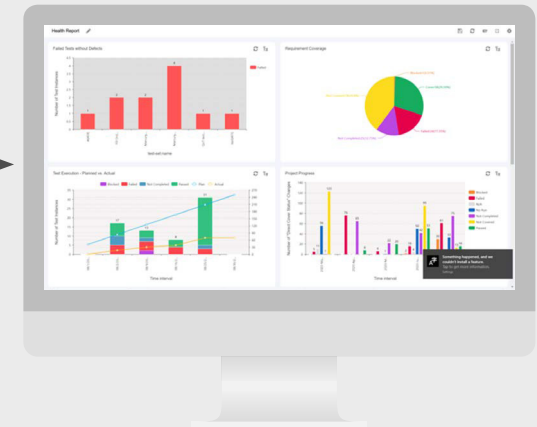
Popular software development tool used by agile teams

Atlassian Jira



Popular test management tool used by quality assurance

OpenText ALM



Eliminate communications barriers



Unite Dev and QA on common goals



Stay with the tool best suited



Keep Ops in the Loop

When your software reaches users during production, it enters the operation phase. You probably use a service management tool as a way for users to report issues or get assistance.

A toolchain links operations, development, and testing. You can map incidents or cases in your service management tool to defects or requirements in your planning and test management tools (ALM/Quality Center, for example). Then feedback from the production environment goes back to the development team. While development fixes the issue, the operations team gets visibility into the progress.

Now, it's time to combine the data from Dev and Ops for analysis.

Analyze and Improve

With your tools connected, data from multiple sources can offer useful insight into where you can improve. You can leverage the analytics available in ALM/Quality Center immediately with the built-in anomalies graph, cycle time graph, composite graph, and trend graph.

Here are two use cases when you integrate ALM/Quality Center with a service management tool:

USE CASE 1

Mean time required to resolve (MTTR)

You can measure the average end-to-end resolution time for customer-reported defects. You can also find out how long defects stayed in which state using the cycle time graph. For example, if they stay in the "Pending Dev" state very long, you may have a capacity problem in your development team. This capability shows you how Dev impacts the customer experience.

USE CASE 2

Defect escape rate—the number of quality issues that make it to production

Gain an understanding of how many issues are detected before release and how many come from the production environment via service management tickets. A higher rate indicates you need to improve your testing processes.





Collaborate

Good DevOps practices tear down silos by fostering a culture of collaboration. With a toolchain, hand-offs between teams are very efficient.

But collaboration also includes sharing information and tackling tasks together. That's why a collaboration tool is important in a toolchain.

Microsoft Teams and Atlassian Confluence belong to this category. They enable collaboration on projects and use chat or mention features to notify colleagues about relevant events.

ALM/Quality Center contributes to collaboration with out-of-the-box integrations with Teams and Confluence.

Moving forward, you'll find ways to finetune your toolchain—ALM Octane can help.

Prevent Defects from Code

Testing against executables helps you detect issues in your software. But modern quality practice is shifting more to defect prevention.

Many defects are avoidable by reviewing and scanning things like architecture and source code. The earlier you find defects, the less it costs to fix them.

When you want defect prevention, it's time to adopt ALM Octane. With this solution, you can use tools like SonarQube and Fortify Static Code Analyzer to analyze code quality and security vulnerabilities. You can track how committed changes, branches, and pull requests relate to your backlog items and defects. You can also identify hotspots in your code.

Health & Life Sciences Case Study

See how [Secure agile testing processes with ALM Octane and Fortify help deliver a superior patient experience.](#)

Strengthen Release Processes

The purpose of release management is to improve the success rate of releases. For mission-critical releases, setting up quality gates is an effective way to ensure that you ship high-quality software into production.

However, it can be dangerous if quality gates don't control automated flow in a toolchain. Even a small increase of production defects can cost millions of dollars.

ALM Octane helps you set up solid release processes with quality gates. It gives you the control and visibility you need to handle quality issues that happen far before the release date.

Gain Pipeline Visibility

Your toolchain enables faster delivery of software through CI/CD pipelines. However, the pipelines can be highly complex, especially for enterprise applications.

Under Agile and DevOps practices, build and test frequently happen, which creates massive pipeline run data. But what does all this data mean?

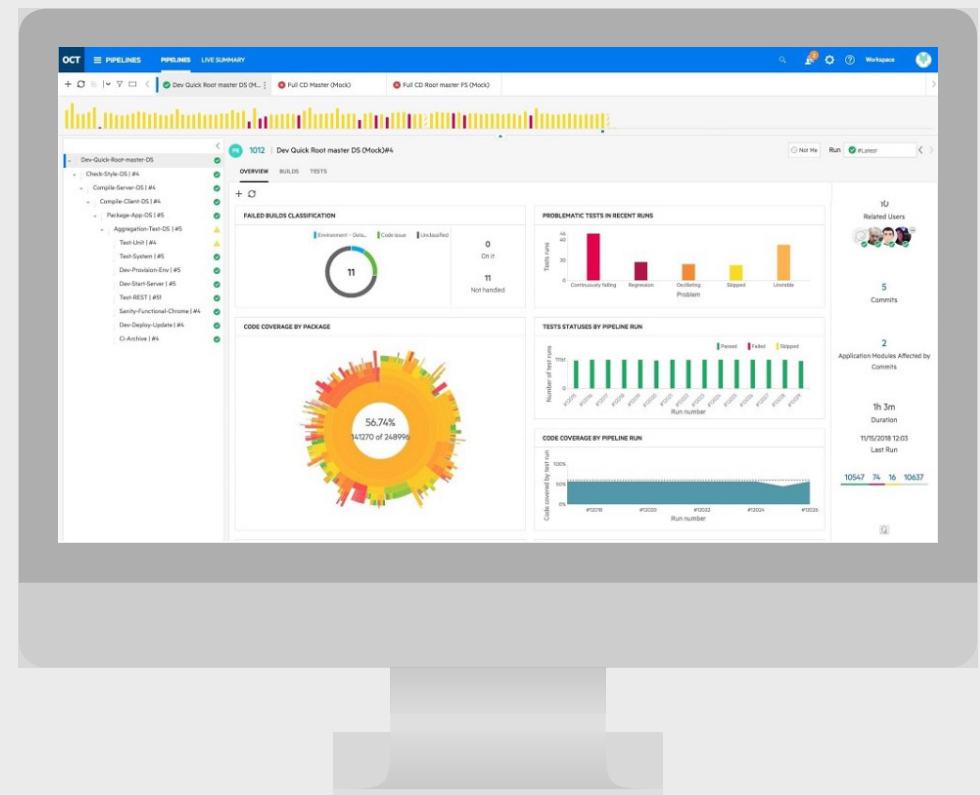
Build failures and automated test failures are also challenges. They slow you down and affect pipeline performance. But troubleshooting them isn't easy.

ALM Octane provides two critical capabilities to keep delivery moving fast:

Pipeline metrics—By visualizing the status of jobs and tests together, you get the big picture and can see trends in real time: coverage, test success rate, job duration, and more. You'll be able to see problem areas and drill down to address them.

Failure analysis—By automatically classifying failures through error message patterns, you can focus on the most common reasons first. Then you can efficiently fix those failures.

The popularity of CI/CD has resulted in more CI/CD server tools. With ALM Octane, you choose the one you want. It supports Jenkins, Bamboo, TeamCity, GoCD, GitLab, Azure DevOps, and OpenText PulseUno.



More to Come

With its rich set of out-of-the-box integrations, ALM/Quality Center is a good fit for test management in your DevOps toolchains. After all, you have the advantage of using the testing assets you've already created.

Your DevOps toolchain will evolve as you strive to further boost the efficiency of your software delivery. By using Micro Focus ALM Octane, you can achieve more with your toolchain and prepare for the next level—[value stream management](#).

OpenText continues to innovate application delivery management. Stay tuned for new capabilities from future versions of our ALM.

Learn more from:

[ALM/Quality Center Integrations Overview ›](#)

[ALM Octane Integrations Overview ›](#)

Bookmark product homepages:

[ALM/Quality Center ›](#)

[ALM Octane ›](#)

Free Trial and Demo:

[Try ALM Octane Free ›](#)

[Request a Demo of ALM/Quality Center ›](#)

Contact us:

[Let us know how we can help ›](#)

