



Brochure

Application Delivery Management

Lifecycle and Portfolio Management: Accelerating Delivery Pipeline with Atlassian

Better Together: Discover how Atlassian customers get more value from their Agile/DevOps processes by teaming up with Micro Focus Lifecycle and Portfolio Management (LPM) software.

Introduction

- Atlassian customers are using Micro Focus Lifecycle and Portfolio Management (LPM) software to improve their Agile/DevOps processes and results. With this combination teams can make better and faster decisions
- Enhance collaboration between R&D, QA, and Project Management Office (PMO) teams
- Gain real time visibility into their quality parameters and test results

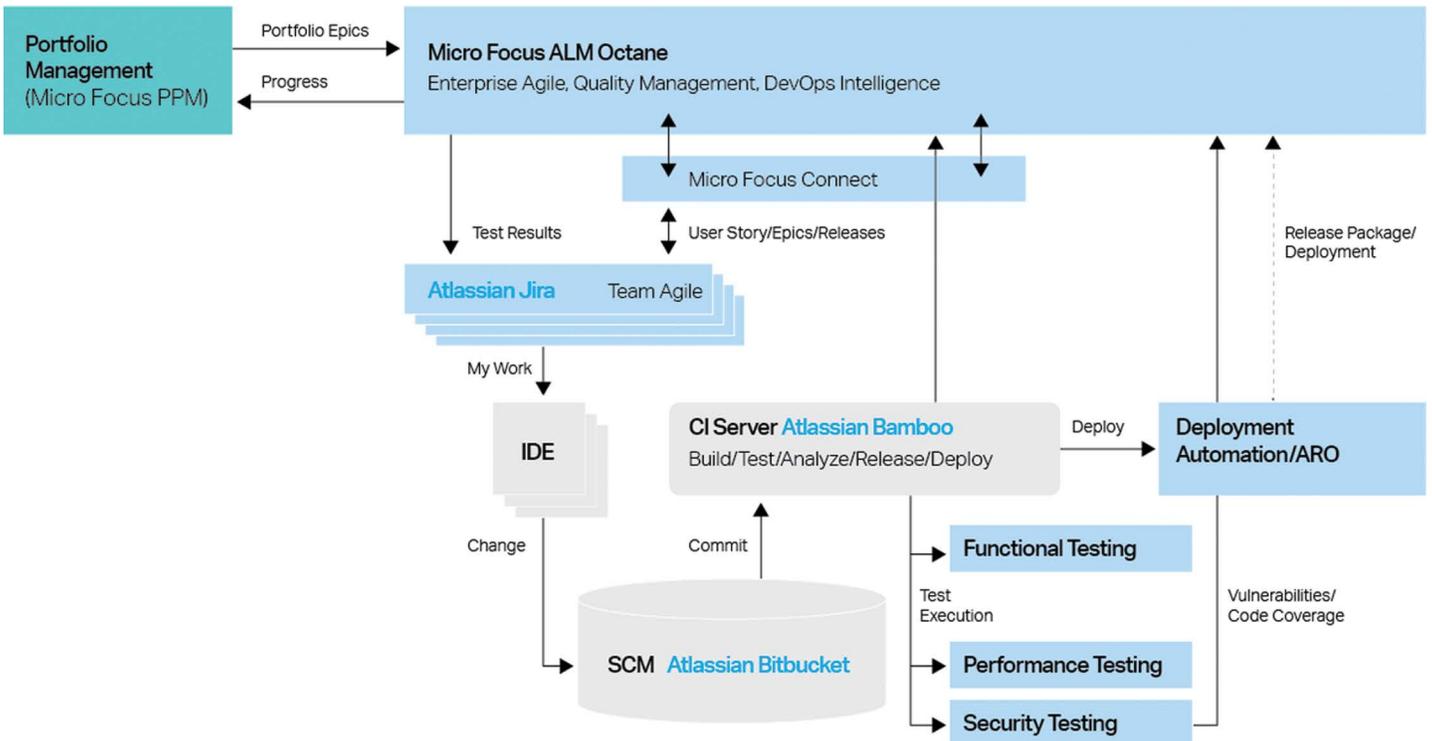
In their eyes, it proves the two solutions can also work side-by-side, rather than as an either/or option.

There are other strategic and operational rewards when you deploy Micro Focus LPM with Atlassian. Full hands-on portfolio control, while working at both code and development cycle level, helps ensure a quality, on-time delivery. Customers who adopt the 'better together' principle release and deploy better software, on time, and leverage the maximum functionality and collaboration from their delivery ecosystems.

Together, Micro Focus LPM and Atlassian can be a single source of truth for application lifecycle management. Top 10 tangible benefits include:

- Project and Portfolio Management (PPM)
- Enterprise-scaled Agile
- Quality management
- Test management
- Team collaboration
- Application delivery with Jira, Bitbucket and Bamboo
- Source control management with Atlassian Bitbucket
- Release control
- Deployment automation
- REST APIs for Jira Service Desk

How Micro Focus Lifecycle and Portfolio Management Complement Atlassian



Top 10 Areas Where Micro Focus LPM Management Complements Atlassian

Project and Portfolio Management (PPM)

Micro Focus PPM complements Atlassian by providing portfolio, financial, project and resource management, planning for epics, user stories, and defect management—all required for Scaled Agile Framework (SAFe).

DETAILED FEATURES INCLUDE

- **Work Plan:** Import Jira issues as tasks in a PPM work plan. Content import formats include a single epic, a specific board, or the entire contents of a Jira Project, and enjoy a more flexible hierarchy structure. In the post-import scenario, the costing engine can provide a cost estimate, or generate 'actuals' based on the Jira issues story points.
- **Timesheets:** When completing a PPM timesheet, users avoid double entries by importing time already captured in Jira. The user can manually relate the imported timesheet lines to PPM entities (Tasks, Projects, Program Milestones, etc.), or automatically relate by PPM when there's no possible confusion.
- **PPM Governance:** Configure a request type to be synced to a Jira Issue, created by PPM, at any point of the workflow. Then bi-directionally synchronize the fields to bring any changes to mapped fields done in Jira into PPM, and vice versa. Use this integration when leveraging the PPM governance capability to design, review and approve large Agile initiatives in PPM for later execution in Jira.
- **Sync PPM Portfolio Epics:** Use this integration to define portfolio epics in a PPM portfolio, and push it to multiple Jira projects for execution by distributed teams. PPM then regularly pulls the values of story points of Jira epics into PPM, and aggregate them to mark the progress of the portfolio epic. This is useful for large organizations, because it works even when using different Jira instances. Alternatively, push the same PPM portfolio epic to teams working with different tools, bring back story point data, and aggregate it back into PPM.

Enterprise-Scaled Agile

Because teams owning Jira must robustly customize their software, there is no easy way to generate overall status progress across multiple teams, and manual reporting is costly throughout the enterprise. Synchronizing Jira software projects and instances with Micro Focus ALM Octane generates real-time graphs, dashboards and reports for insights into the status and progress across multiple teams. Alternatively, organizations leveraging Open Data Protocol (OData) to report into Business Intelligence (BI) solutions reduce time, effort, and the cost of manual report consolidations.

Quality Management

Whether organizations are moving from Waterfall to Agile or accelerating their AI and machine learning automation initiatives, robust product quality is essential. Delivering that quality depends on being able to analyze and track software quality with organization-wide end-to-end testing, cross-feature functionality testing, and regression testing. In addition, support for Behavior-driven Development (BDD) testing, manual testing, regression suites, and auto-discovery of automated functional/performance tests and runs is a pre-requisite. By associating features, pre-production and production defects, tests and test runs with application modules, you get a completed solution that enables solution trains to measure not simply release quality, but holistic product quality from every angle.

ALM Octane adds value to Jira by consolidating quality status and progress across multiple teams, without multiple plug-ins. This reduces manual effort and the time involved in preparing reports. With ALM Octane, users gain insight into holistic product quality across multiple teams and projects, saving time, effort and cost by leveraging real-time graphs, dashboards and reports.

Jira can prioritize defect fixing based on quality criteria and business criticality. ALM Octane users easily track quality with application modules, defining one or more quality criteria, such as test coverage, problematic tests and percentages of automation, risky commits and failed last runs.

An additional value Jira highlights is in tracking the contribution of dev and test teams to quality, to inform decisions around investments and work prioritization. Jira teams using ALM Octane track dev and test team quality side-by-side in a single dashboard. Criteria include risky commits, test strategy, planned effort and progress within automation, defects, test coverage, results, and effectiveness. This empowers teams to make good decisions around investments and prioritizing work.

ALM Octane also determines the quality of the release by providing Jira traceability reporting across multiple entities. This cross-entity traceability aligns with different personas and means you can deploy applications with confidence.

Test Management

Because individual plug-ins do not always meet the needs of QA, many organizations use separate plug-ins for manual test cases, Gherkin tests, functional test automation, and other automation test tools. ALM Octane manages your manual and Gherkin tests, and executes automated functional, security and performance tests using one or more popular CI servers.

Managing many plug-ins across multiple instances is challenging, and can be costly. ALM Octane reduces this complexity and these costs, and enables the updating of backlog items synchronized from Jira.

Tracking the testing status and progress for both manual and automation is also crucial to reduce or even eliminate testing cycle delays. ALM Octane plus Jira use test suites and pipeline for execution, and tracks testing status and progress for manual and automates tests. The combination delivers function, security and performance.

Team Collaboration

Development teams need visibility into the coverage of all test types, including manual, functional, security and performance. They need to avoid spending unnecessary time seeking complete coverage of a given backlog item. Development teams using ALM Octane with Jira can see the test coverage of a backlog item within Jira, a more efficient use of resources.

It can be difficult for QA teams to work with multiple teams and projects. Logging into multiple instances is always challenging, and duplicated efforts and error-prone processing can affect application quality. ALM Octane easily synchronizes Jira projects/instances using comments and notifications.

Application Delivery

Combining your Atlassian investments with Micro Focus tools creates an overarching framework of control and insight. A lack of data and insights can increase Mean Time to Repair (MTTR) and slows down application delivery. Using ALM Octane side-by-side with Atlassian (Jira, Bitbucket and Bamboo), empowers roles across the application management lifecycle to leverage pipeline and dashboard module data to minimize MTTR and accelerate delivery.

Simplifying navigation through multiple test tool reports, CI logs, error messages, and stack trace, to find the root cause of problems reduces development team time, effort, and cost. Module graphs and tables provide both summary-level information of pipeline stability, classification of build failures and tests, so ALM Octane pipeline complements Atlassian Jira, Bitbucket and Bamboo. By drilling down from the summary, teams source information for each test report, error message, and/or stack trace from a single view.

The ability to identify failed tests reduces the time taken to fix defects. ALM Octane helps Jira enable teams to quickly identify newly failed tests

and classify them using the business context. It does this with application modules, prioritizing and reducing the time to fix critical defects.

Teams must identify related test failures and group those tests with similar underlying issues. They need to avoid spending too much time navigating every failed test. The ALM Octane test clustering capability reduces MTTR and accelerates delivery by scaling down the number of failures, and groups the tests that share test-failure characteristics.

Even with every plug-in in place, and with thousands of automated test runs as part of the pipeline, many organizations cannot classify failed tests. Analyzing failed tests wastes time, effort, and increases cost. With ALM Octane, Jira automatically identifies problematic tests across pipeline builds and classifies them for quicker resolution. In addition, it tracks any time trends in failure recovery and identifies those tests with a long duration, saving analysis time, effort, and cost.

A fast and easy method of tracking both the failure age and first-failed build of each continuously failing test, would negate the time and effort taken in tracking the build commits. Using ALM Octane, Jira teams get the status of each test in recent builds, and navigate quickly to the first build where the test started failing. Users can understand the status of multiple tests across recent builds in a single view, identify the root cause using the commits, and identify the build developers in place when the test first started failing.

Identifying the possible related users behind each failed test wastes significant time and effort, and requires multiple resources to track each related commit and developer information. ALM Octane provides users with Jira insights to identify each failed test, assign the defects to related users, and quickly fix the problem.

“ALM Octane supports our digital transformation objectives by addressing various challenges. We’re producing releases faster and having a single platform for all automated testing makes for easier management and better collaboration between the development and testing teams.”

YANN HELLEBOID
Testing Community Manager
Orange Labs

Source Control Management (SCM) using Atlassian Bitbucket

Using ALM Octane, teams can track pull requests in the SCM system via Bitbucket and access details around impact release and build quality. ALM Octane tracks committed changes and their effects to Bitbucket in a pipeline run. With this integration, ALM Octane provides insights into multiple elements:

- List of backlog items and their associated commits
- List of commits associated with an epic or feature
- Number of committers and commits
- Commits related to failed automated test runs

In addition, ALM Octane displays [a] risk indicator near commits or features that indicate any changes committed in sensitive areas of code. QA teams use these insights to identify areas of significant change and plan the test strategy, while the development team can identify those areas too risky to change, while the Project Manager or Project Management Office may increase the level of testing, or postpone the release.

Release Control

The integration between Micro Focus Release Control and Atlassian Jira enables the development and support of Jira items as part of the release management process. The available plugin steps include associating Jira items with requests, creating Jira items, assigning Jira items to resource, and the transition of Jira backlog items.

Request providers associate Jira items to a release package/release train by selecting from existing Jira items, or creating new items from within Release Control. Execution providers assign team members Jira items and transition them to the next workflow stage, all from within Release Control.

By associating Jira items to release packages, Micro Focus Release Control provides visibility into what is being deployed to a specific release package, and ensures full traceability of the release activity back to the initiating Jira request, or request.

Deployment Automation

Plug-in for Jira Software

Micro Focus Deployment Automation and Atlassian Jira integration also enables the development and support of Jira items as part of the deployment process. The deployment automation plug-in enables the creation of Jira items, status checks, the publishing of issue reports,

updates, and the addition of comments as part of the deployment pipeline. It provides automation, visibility and traceability between deployment automation activities, and the issue tracking system.

Plug-in for Bamboo

The integration between Micro Focus Deployment Automation and Atlassian Bamboo ensure tight integration with Bamboo projects as part of the deployment process.

Micro Focus Deployment Automation works seamlessly with the continuous integration tool within Bamboo. Its delivery tools streamline the deployment pipeline automation by replacing homegrown scripts and manual processes. Configured at the design and runtime stages, the plug-in supports most systems through its extensible architecture. The Micro Focus Deployment Automation plug-in includes Atlassian Bamboo as part of the deployment process. It includes the following steps/capabilities: Add Build Comment, Get all plans, Get all projects, Get build comments, Get queried builds, and Make build.

Jira Service Desk

Integrating with ALM Octane via REST APIs, Jira Service Desk ensures rapid service delivery by supporting IT Service Management processes such as Request, Incident, Change and Problem Management.

Delays in resolving production defects result in a poor customer experience and decreased application adoption, increasing MTTR. The main causes of delays in resolving production defects include lack of collaboration, poor visibility, and a time-consuming, error-prone manual process. By integrating Jira Service Desk with ALM Octane, either through a REST API or by using Jira software, customers increase accuracy, improve production defect visibility, and foster collaboration among teams.

Application-related problems entered in Jira Service Desk become production defects in ALM Octane. As the development team(s) begin working on the defects, ALM Octane notifies the support team on the progress for onward communication to the end user. Once defects are fixed, Jira Service Desk updates, the fix is validated, and the customer notified. This process improves feedback between Dev and Ops, enables faster defect resolution, thus improves the customer experience and ultimately retention.

Learn more at
www.microfocus.com/en-us/products/alm-octane/overview

Contact us at:

www.microfocus.com

Like what you read? Share it.

