

The Challenge: Balancing Change & Control of Continuous Delivery at Scale

Challenges of Continuous Delivery at Scale

- **Lack of visibility into specialized tools used across the CD pipeline**
- **Difficult to assess and reduce schedule risk across interdependent pipelines**
- **Hard to identify what changes new code contains to properly assign test cases**
- **Inefficient handoffs from dev to test impede progress**

DevOps bridges the gap between development and operations to deliver business value more frequently. DevOps practices break down traditional domain silos that inhibit communication and process flow, establishing a culture of shared responsibility for development and operations.

As the separation between dev and ops is reduced, release trains begin to operate as a Continuous Delivery pipeline to define, implement, and deliver innovation faster by minimizing time-consuming handoffs and reducing operations support. This journey to Continuous Delivery (CD) requires release teams to address inefficiencies and constraints that may have been plaguing the organization for years. Existing software practices may create technical debt and rework that inhibits adoption of new methods. Or the current deployment pipeline is not repeatable, requiring manual heroics that drain resources and restrict team velocity.

The application architecture also plays a big role in the transition to DevOps. When architecture is loosely coupled, release trains can achieve quality validation and deployment independently of one another – ideal when establishing a flow-based CD model. However, the reality for most large enterprises is a tightly coupled application, with many interdependencies across multiple release trains. Deployment pipelines may run in parallel initially, then eventually converge as the test environments become increasingly complex to reflect the final production state. If a single pipeline faces delays, the flow of the entire release is jeopardized.

Plutora Environments establishes a single command center to schedule, manage, and configure test environments to expedite the handoff of new code from dev to test. Whether test environments are on-prem or in the cloud, Plutora Environments creates a single source of truth for environment management. Integration with Jenkins build automation, Github version control and Plutora Release maintain visibility of fast moving CD pipelines at scale.

Expedite the Handoff of New Code from Dev to Test

Lean-Agile methods emphasize optimizing across the full value stream and wait states are often the largest inefficiency. Automating tests and honing agile methodologies will increase efficiency, but optimizing a component doesn't optimize the system. If test teams are kept waiting for a properly configured test environment, or waste time identifying changes, the delivery flow stops.

Reduce dev/test cycle time

Many organizations spend more time getting large, complex enterprise environments up and ready for testing than they do writing code. Faster QA team access to test environments improves cycle time, especially important when dealing with critical fixes for sev 1/showstopper issues. Plutora highlights when dev has a new build available, increasing test team responsiveness and eliminating the delay in finding out when new code is ready for testing.

Enforce code quality gates

Multiple interdependent release trains require process efficiencies that minimize the time different groups across the CD pipeline invest in defective code. At the same time, developers shouldn't waste time triaging defects that end up being environment or deployment issues. With Plutora, release teams can create a well-structured deployment plan that enforces code quality gates designed to limit multifaceted triage scenarios and facilitate identification of defects before moving code onto the next, more complex test environment.

Ensure test coverage

As dev team velocity increases, test teams may have a hard time tracking the change requests associated with a new build, making it difficult to accurately assign the test cases necessary to ensure accurate test coverage. With Plutora's Continuous Delivery Pipeline functionality, test teams automatically link the defect fix ID back to the original requirement to quickly associate new code commits with appropriate tests.

Automatically fetch version-controlled development artifacts

Test environment configurations continually change as code is validated along the delivery pipeline, and test teams run the risk of executing tests on incorrect configurations. On-prem, cloud, and hybrid architectures further complicate managing & planning. Plutora integrates with Jenkins to automatically pull version numbers from the Jenkins build server, creating a single source to track and update configuration settings for all test environments across the portfolio.

On-demand deployment to test environments

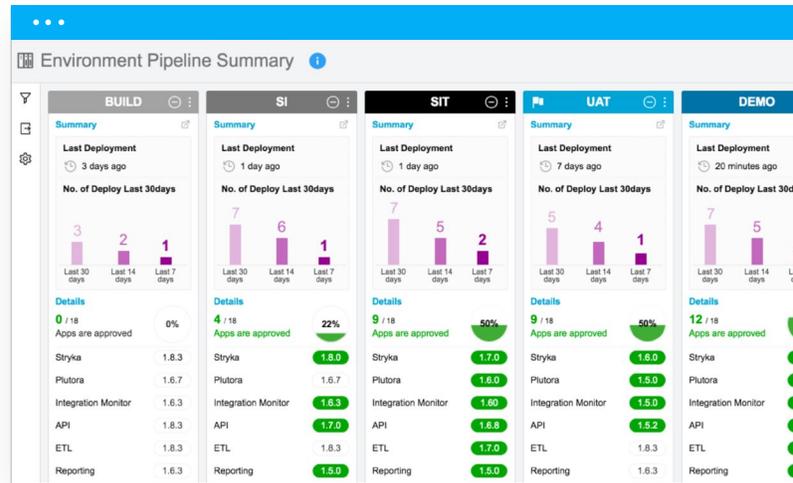
Don't make fast moving test teams wait for a test environment to become available. Plutora links Jenkins jobs to test environments, where a build is triggered on-demand to expedite the hand-off of new code from dev into test environments. Code commits newly assigned to the test environment are highlighted, and are easily compared across independent environments.

“For large tightly coupled systems, developers often don't understand the complexities of the production environments. Additionally, the people that understand the production environments don't understand well the impact of the changes that developers are making. There are also frequently different end points in different test environments at each stage of the deployment pipeline. No one person understands what needs to happen all the way down the deployment pipeline. Therefore, managing environments for complex systems requires close collaboration from every group between dev and ops.”

Gary Gruver,
from *Starting and Scaling DevOps*

Speed progression of code along the delivery pipeline:

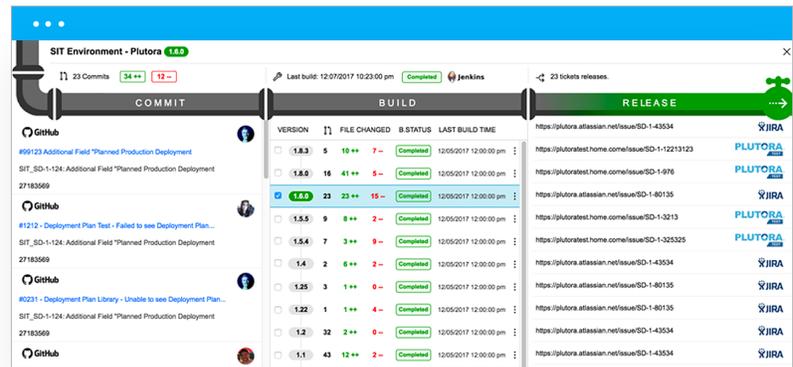
- 1 Assess new code quality as it progresses along the delivery pipeline
- 2 Identify impediments and process inefficiencies
- 3 Establish traceability from inception to production



Maintain visibility of fast moving CD pipelines at scale

Maintain Visibility of Fast Moving CD Pipelines at Scale

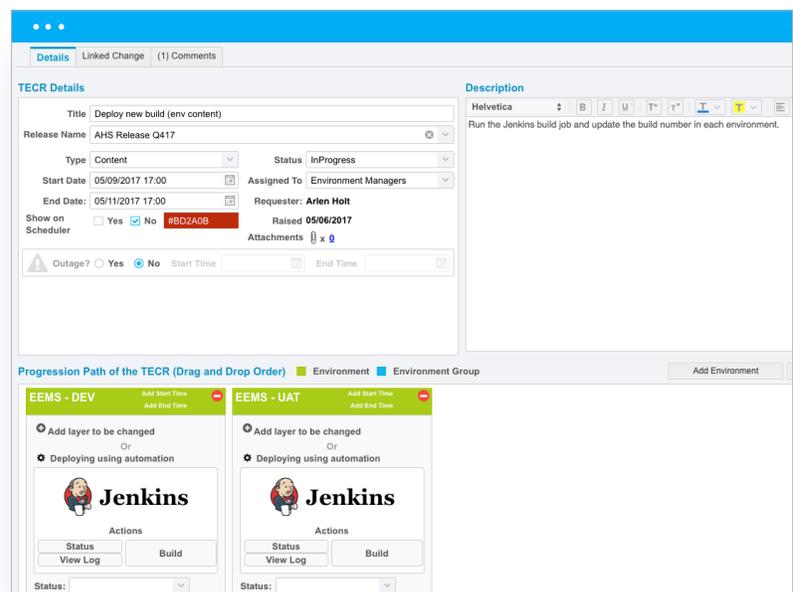
Release managers are tasked with assessing schedule risk and monitoring application quality. However, many activities in the CD pipeline (code commits, builds and their version numbers, test environment deployments, test plans created for business requirements or issues, test cases run against each build, status and results of these tests) are not visible, hidden within the specialized tools of dev and test teams. This results in a large amount of release activity with limited visibility and coordination.



Expedite the handoff of new code from dev to test

Monitor product quality of multi-pipeline releases

With Plutora, release managers can quickly view test status and results in real time to continually monitor product quality and evaluate schedule risk at each phase of the CD pipeline. Centralized dashboards provide visibility of multiple release trains across the portfolio. Easily drill down to gain deeper insights of automated and manual test results and defect rates.



Establish traceability from inception to production

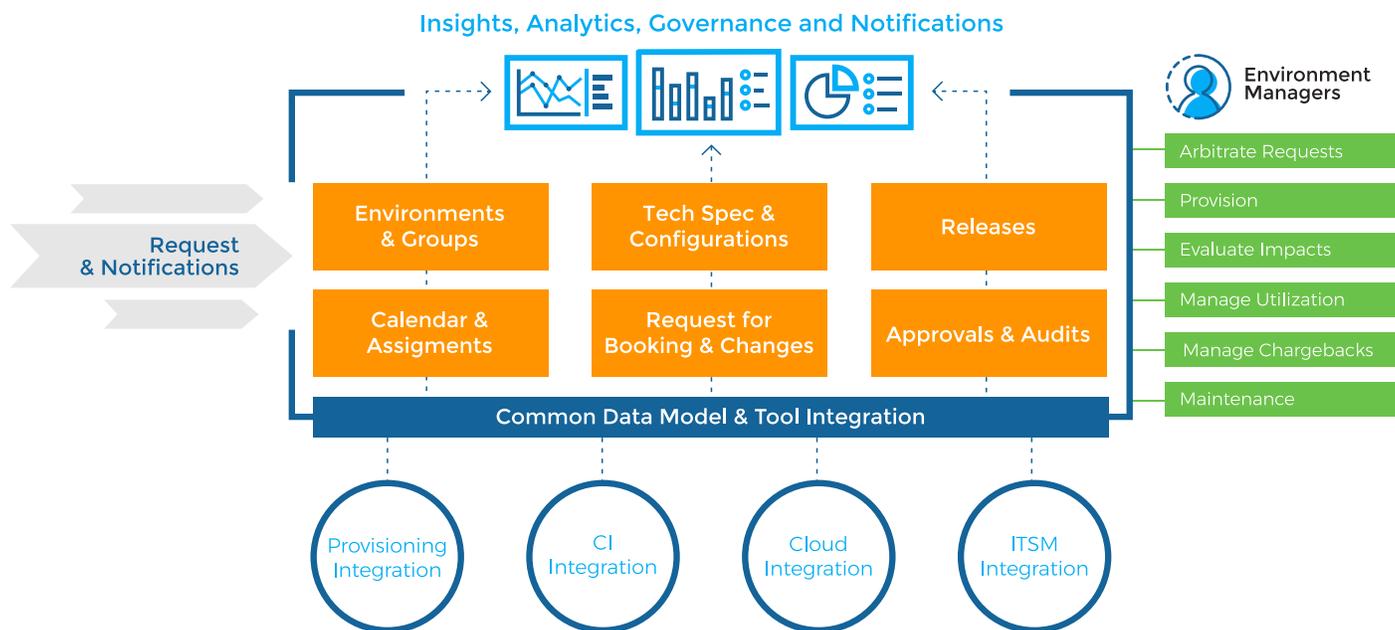
Plutora tracks CD workflows and change requests to ensure compliance, establishing traceability of where each change has been introduced, when versions were deployed, and what tests have been run. Release audit history is automatically updated to include Jenkins build numbers deployed to each test environment over the course of a release.

Build test environments on-demand

Simplify Test Environment Management

Plutora Environments gives internal and external environment teams one place to collaborate on and view environment bookings, allocations, configurations, and conflicts.

It supports intake requests, environment change requests, environment allocations, conflict detection, and impact analysis with complete notification options for stakeholders. Plutora Environments improves the quality, availability, and utilization of your environments by reducing costs and delays. The solution simplifies the management of requests by providing a single system for all tracking.



About Plutora

Plutora, the market leader of value stream management solutions for enterprise IT, improves the speed and quality of software creation by capturing, visualizing and analyzing critical indicators of every aspect of the delivery process. Plutora orchestrates release pipelines across a diverse ecosystem of development methodologies, manages hybrid test environments, correlates data from existing toolchains, and incorporates test metrics gathered at every step. The Plutora Platform ensures organizational alignment of software development with business strategy and provides visibility, analytics and a system of insights into the entire value stream, guiding continuous improvement through the measured outcomes of each effort.

PLUTORA

Learn more: www.plutora.com

Email: contact@plutora.com