5 steps to getting started with DevOps in Pega

Adopting DevOps in your organization can be a daunting task – with so many excellent tools to help your team deliver quality software quickly. In fact, there are a plethora of surveys and forecasts that show the benefits of DevOps adoption. However, adopting a DevOps toolchain isn't a guarantee of success. In a 2019 blog, Gartner's Chris Little noted that "By aligning your application architecture team's working practices with those of your DevOps teams, you can ensure the success of your DevOps initiative and overcome resistance to architecture requirements."

Pega Infinity™ comes with DevOps best practices in mind, allowing low-code application development teams to apply DevOps best practices across the application lifecycle.

Here are five steps to address DevOps methodology across the application lifecycle.

PEGA

STEP 1: START PLANNING WITH AGILE WORKBENCH

Agile Workbench enables app development teams to close the gap between business and IT by driving continuous feedback and collaboration throughout the development process. It all starts by creating a work item, which may be a user story, bug, or feedback item, in Agile Workbench. With this tool, stakeholders can provide feedback in multiple formats, including video, annotated screen captures, file attachments, or URLs to drive clarity in collaboration.

Once feedback is posted to Agile Workbench, project stakeholders can view, update, and triage the work items. When creating features, associate each feature with a work item to establish traceability from requirement to implementation. You can also configure Agile Workbench to integrate with Pega Agile Studio, Jira, or Rally – so bugs and stories are synchronized between the two systems. Within each work item, you can communicate with other team members and stakeholders by using Pulse for transparent and efficient application development.

STEP 2: BRANCH AND MERGE IN PEGA INFINITY

Branches are a critical part of delivering updated applications quickly. A branch allows multiple teams to work on features or bug fixes simultaneously in a version-controlled environment. By providing teams with separate branches on the same base ruleset, you minimize the risk of errors and work overwrites that might arise when both teams work on the same ruleset simultaneously.

To develop applications using branches in Pega, start in Dev Studio by saving a new version of your base application. Next, you can add development branches in the new version of your application, making sure to add a Branch ID and a Branch Name to identify your branched versions. Your team can then “check in” and “check out” rulesets as you make your changes.

STEP 3: BUILD YOUR PIPELINE WITH PEGA DEPLOYMENT MANAGER

Pega Deployment Manager is the pipeline tool included in your Pega Infinity system. It is a model-driven solution with built-in best practices to deploy Pega applications using DevOps best practices. Deployment Manager requires its own environment to operate. In Pega Cloud®, your subscription entitles you to a Deployment Manager environment; while for clients operating on-premises or in their own cloud environment, an environment must be configured.

Deployment Manager supports multiple types of pipelines to optimize your operations. Plus, Pega provides useful pipeline templates for each type of pipeline to speed your adoption.

PIPELINE TYPE	DESCRIPTION
Deployment	Deployment pipelines are an application pipeline type that generates artifacts from the system of record and delivers them through the various stages of the workflow. When you add a pipeline, you define all the stages and tasks that you want to do on each system. If you are using a QA system, you can run test tasks to validate application data. Pipelines are the first step in setting up a process to deploy a change through the release.
Merge	Merge pipelines enable developers to submit rule branches for a merge after validating the changes against specific quality gates, such as guardrails, branch reviews, and automated tests. For example, if you are using branches, you can start a build when a branch is merged. Pipelines are the first step in setting up a process to deploy a change through the release.
Business change	Business change pipelines are used to support every day, business-as-usual changes to your application. This pipeline type allows users to respond to changing requirements by modifying and deploying application rules in a controlled manner. When you add a pipeline, you define all the stages and tasks that you want to do on each system. Pipelines are the first step in setting up a process to deploy a business change outside of an enterprise release.
Data migration	Create a pipeline by defining the production and simulation environments and the application details for the pipeline. By using a data migration pipeline, you can export and import simulation data automatically.
Deploy artifact	Use deploy artifact pipelines to deploy an existing artifact that Deployment Manager validates. For example, use deploy artifact pipelines to deploy an existing artifact to a production or pre-production environment. You can also promote an artifact from one environment to another to update those environments to their most recent stage.
Update	Update pipelines automate and simplify near-zero-downtime updates by automating the basic update process. By using Deployment Manager as a standard DevOps approach during updates, you can promote the update fixes with ease to production and non-production environments – all through a single pipeline.

STEP 4: AUTOMATING YOUR TESTS

Test automation is an important area to focus on when building a DevOps practice. Testing is both critical to your development process and likely the most time-consuming step in the lifecycle. Automated testing not only helps improve the quality of delivery, but also can both reduce throughput time for releases and maximize your pipeline capacity.

The testing pyramid is often used to depict how to invest in test automation. Unit testing typically has the highest test case count because they are the most cost effective to quickly develop. Next are function and API tests, and finally user experience tests at the peak.

Pega provides support for all levels of the testing pyramid. We make it easy for the citizen developer to create simple and powerful tests right from Agile Studio. Using Gherkin-style test authoring, development teams can create and manage test plans for their applications at the user-story or epic level.

To execute automated unit tests, Pega provides PegaUnit with Pega Infinity subscriptions. PegaUnit supports grouping unit tests into suites where you can prioritize test case order. At the end of the test cycle, PegaUnit provides testing reports to enable release managers to view unit test compliance.

At the functional level of the test pyramid, Pega ADK Simul8r is the new, low-code approach to simulating your integrations. It's provided as a component which can be added to Pega Infinity versions 8.3 and above. The component introduces a new "Simulation" rule type, available from the "Integration-Simulation" rule category. This new rule type gives you a simple UI for capturing all the API responses you would like to simulate.

Pega Digital Experience (DX) API is a set of model-driven API endpoints that enable developers to view, create, and update Pega cases and assignments remotely. More than this, the DX API powers Pega's out-of-the-box user experiences, and these same endpoints allow developers to build custom front-end experiences. This means that you can build consistent user experiences for Pega applications without encoding business logic into each and every channel.

Finally, for user experience testing, Pega Scenario Testing allows you to run scenario tests against a user interface to verify that the end-to-end scenarios are functioning correctly. The UI-based scenario testing tool allows you to focus on creating functional and useful tests, rather than writing complex code.

WHAT IS GHERKIN?

Gherkin is a programming language developed by Cucumber that allows developers to define tests in a format that drives collaboration across your development team and business stakeholders.

Gherkin uses a strict but simple syntax where each line in the file begins with a Gherkin keyword and specifies a different component of the feature. The goal is to develop clear definitions for each characteristic, which can subsequently be rigorously evaluated. Gherkin is meant to be human-readable, but, like other formats such as XML, it is not intended to be machine-readable.

STEP 5: MONITORING QUALITY WITH PEGA PREDICTIVE DIAGNOSTIC CLOUD

Pega Predictive Diagnostic Cloud monitors your applications at run-time and gathers performance, quality, and diagnostic data for Pega Platform™ components. Predictive Diagnostic Cloud organizes monitoring insights into several views, based on the use case you are addressing.

For DevOps teams, the Predictive Diagnostic Cloud improvement plan provides real-time insights and recommendations on how to improve application performance. In the default view, teams can see the top five most impactful cases to improve the quality of your application, making it easy to identify improvements that will deliver the highest return on investment.

Predictive Diagnostic Cloud monitoring insights

Event Viewer provides the means by which you can investigate issues by accessing the diagnostic data from all nodes in your system. The Event Viewer provides a live feed of log data for events that occurred in every node of your system in near real time and displays all events that PDC recognizes.

Notifications can be delivered as a daily digest or be event-based – to help you address any issues that might occur.

Cases organize the alert messages, exceptions, and other data that the Pega Platform provides and Predictive Diagnostic Cloud gathers. To help you track specific issues, each case in Predictive Diagnostic Cloud groups events that have a similar origin and cause. Cases represent unique problems and require individual solutions

Improvement Plan allows you to view detailed information about issues that affect the performance of your monitored system. Cases are divided into the following categories: database, Pega Platform tuning, connectors, decisioning, exceptions, operations, application logic, and custom cases.

Summing it up

Having the right tools doesn't always guarantee success, but when combined with an organizational commitment to agile best practices, you can deliver applications with higher quality faster. Pega provides low-code application development teams with the tools they need to apply DevOps best practices at each stage of the application lifecycle. As your practice grows, Pega provides the flexibility to integrate with third-party tools using the DevOps API.

About Pegasystems

Pega is a powerful low-code platform that builds agility into the world's leading organizations so they can adapt to change. Clients use our AI-powered decisioning and workflow automation to solve their most pressing business challenges – from personalizing engagement to automating service to streamlining operations. Since 1983, we've built our scalable and flexible architecture to help people focus on what matters most, so they can meet today's customer demands while continuously transforming for tomorrow.

For more information, please visit us at pega.com