Empower teams to support DevOps with Pega Infinity

How Pega sets your teams up for DevOps success

A PEGA WHITEPAPER

PEGA®

TABLE OF CONTENTS

Trademarks

For information about the third-party software that is delivered with the product, refer to the third-party license file on your installation media that is specific to your release.

Notices

This publication describes and/or represents products and services of Pegasystems Inc. It may contain trade secrets and proprietary information that are protected by various federal, state, and international laws, and distributed under licenses restricting their use, copying, modification, distribution, or transmittal in any form without prior written authorization of Pegasystems Inc.

This publication is current as of the date of publication only. Changes to the publication may be made from time to time at the discretion of Pegasystems Inc. This publication remains the property of Pegasystems Inc. and must be returned to it upon request. This publication does not imply any commitment to offer or deliver the products or services described herein.

This publication may include references to Pegasystems Inc. product features that have not been licensed by you or your company. If you have questions about whether a particular capability is included in your installation, please consult your Pegasystems Inc. services consultant.

Although Pegasystems Inc. strives for accuracy in its publications, any publication may contain in accuracies or typographical errors, as well as technical inaccuracies. Pegasystems Inc. shall not be liable for technical or editorial errors or omissions contained herein. Pegasystems Inc. may make improvements and/or changes to the publication at any time without notice.

Any references in this publication to non-Pegasystems websites are provided for convenience only and do not serve as an endorsement of these websites. The materials at these websites are not part of the material for Pegasystems products and use of those websites is at your own risk.

Information concerning non-Pegasystems products was obtained from the suppliers of those products, their publications, or other publicly available sources. Address questions about non-Pegasystems products to the suppliers of those products.

This publication may contain examples used in daily business operations that include the names of people, companies, products, and other third-party publications. Such examples are fictitious and any similarity to the names or other data used by an actual business enterprise or individual is coincidental.

This publication provides a technical overview of Pega's capabilities and architecture. The exact capabilities available to you vary based on your contract and license terms. Please contact your Pega account executive for more details.

This document is the property of:

Pegasystems Inc.
1 Main Street
Cambridge, MA, 02142, U.S.A.
Phone: (617) 374-9600 Fax: (617) 374-9620
www.pega.com

Document: Empower teams to support DevOps with Pega Infinity
Publication date: 2022

EXECUTIVE SUMMARY

DevOps has become the de facto best practice for software development.

Most enterprises are at some stage of their evolution to DevOps best practices. This has been driven by the speed, security, and quality that DevOps teams can deliver

Internal application users and customers alike demand seamless experiences. And the demand for enhancement to these experiences continues to increase. Software-as-a-Service and the move to the cloud results in user demands for ever-increasing functionality without compromising reliability. But simply moving workloads to the cloud doesn't make a team DevOps-ready. You need the right tools and mindset to be successful.

DevOps is fundamentally about increasing collaboration and speed of delivery, effectively reducing time to market (TTM). There is a perception that implementing DevOps is about finding the right tools, moving to the cloud, or using the latest, hottest technology. While these can be components of your technology strategy, at its very core, DevOps is about implementing fundamental practices, such as continuous integration (CI), continuous delivery (CD), and continuous deployment (CD), while fostering a collaborative culture between the teams involved.

The tools are there to empower teams and support these practices. Without this perspective and the resulting change in culture, adopting DevOps will only lead to automation of the application deployment from one environment to another – without improving TTM in any meaningful way. With Pega Infinity™, you'll find all the crucial DevOps capabilities and APIs built into a single, unique solution to significantly slash your TTM, boost productivity, and gain a competitive edge.

The adoption of DevOps best practices has become an imperative for IT teams to rapidly develop and enhance applications. According to the Gartner 2020 DevOps Survey, 82% of organizations have created platform teams to provide developers with self-service access to cloud, automation and infrastructure capabilities.

[1] Gartner's 2020 Achieve Business Agility With Automation, Continuous Quality and DevOps Survey

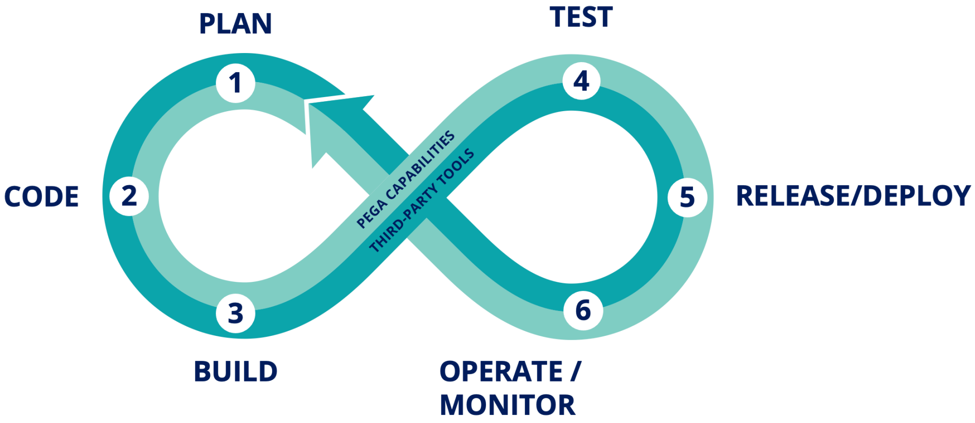
Your DevOps journey with Pega

DevOps is a mindset, not just one tool that will magically help deliver high-quality applications at speed. This mindset has a foundation of an application lifecycle that delivers continuous improvement and enhancements as your market demands them.

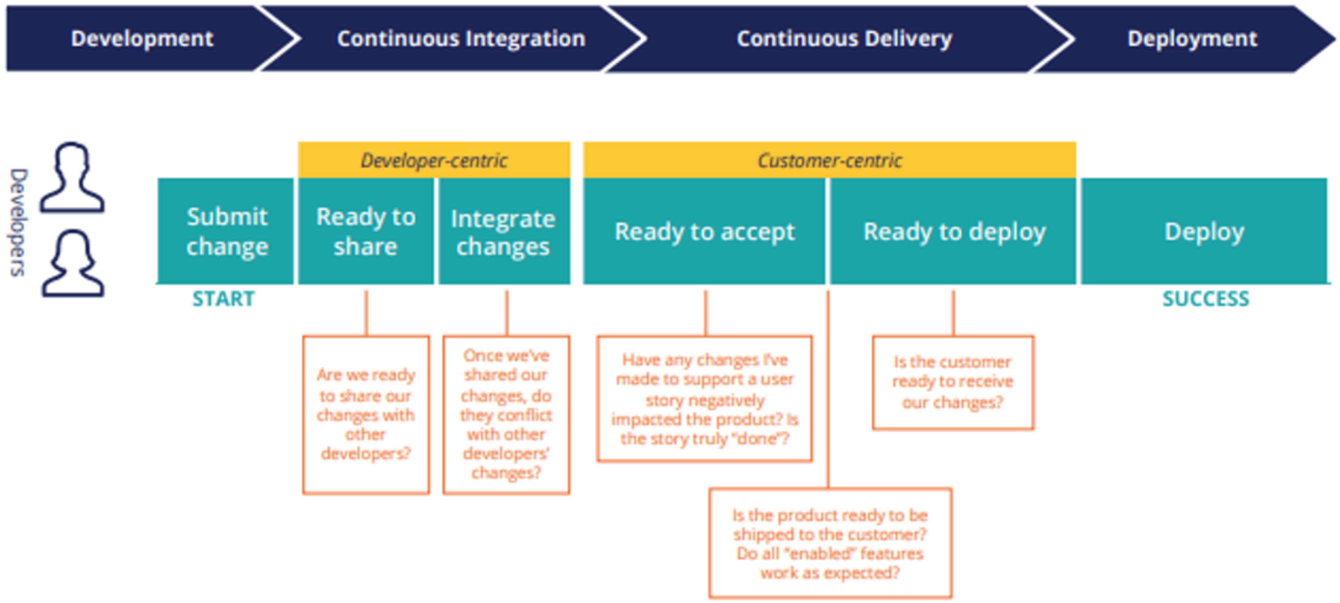
To implement DevOps in your organization, you will need a suite of tools to support you at each stage of the application lifecycle. With continuous improvement of your applications combined with strong monitoring tools, you can maximize business value in a changing market.

The application lifecycle is a repeatable process that accounts for the following:

- Plan: Capture business requirements.
- Code: Create a branch (or new version) of your application implementing these requirements .
- Build: Validate and integrate changes safely, merging with other incoming changes and existing functionality.
- Test: Functional, non-functional (security and performance), and user acceptance testing and validation occurs.
- Release: The build is validated and ready for promotion to the higher environment.
- Deploy: Packing and promoting into higher environments.
- Operate: Environments are monitored to handle scale required to support changes, feedback is collected from users on application behavior, performance, bugs, and more..
- Monitor: Analyze feedback from the application and review overall DevOps pipeline performance to make recommendations to improve overall productivity.



If along the way any issues should threaten the deployment, all relevant stakeholders should be notified so that the reported issue can be investigated and fixed. Tying all steps together into the fundamental DevOps practice is often represented as the continuous integration/continuous delivery (CI/CD) pipeline. This pipeline is designed to deliver application changes incrementally through short delivery cycles – gaining greater speed and frequency without compromising on quality.


Development
Continuous Integration
Continuous Delivery
Deployment
Developers
Developer-centric
Customer-centric
Submit change
Ready to share
Integrate changes
Ready to accept
Ready to deploy
Deploy
START
SUCCESS
Are we ready to share our changes with other developers?
Once we've shared our changes, do they conflict with other developers' changes?
Have any changes I've made to support a user story negatively impacted the product? Is the story truly "done"?
Is the customer ready to receive our changes?
Is the product ready to be shipped to the customer? Do all "enabled" features work as expected?

An effective approach when you are starting to design this delivery process is to make sure the right questions are being asked and that answers are implemented at every stage in the CI/CD pipeline. You should then look to speed up this process through automation – this is where the tools and technology come into the picture.

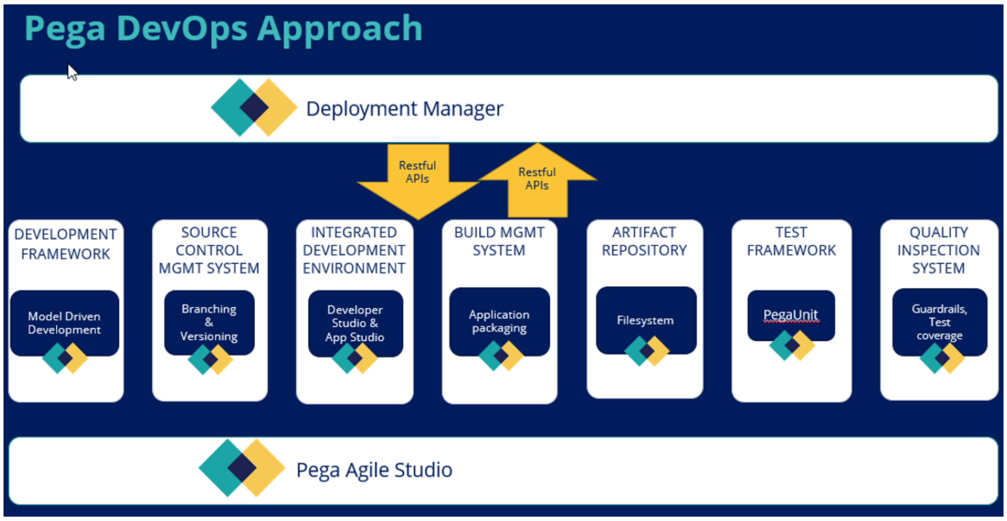
The challenges with collaborative development, especially at scale, are another factor to consider. The efficient transfer of development projects between teams is vital to ensure quality development and deployment. However, without the right tools, the easy transfer becomes one of the most difficult things to manage in a deployment project.

It's also important that the application development artifacts are not separated – even if your teams are. You need a single source of truth for your application and deployment history; if a problem arises, developers have all the information they need to rectify the problem.

Equipping your team for DevOps success

Once you've made the decision to adopt DevOps best practices for your Pega application development, you need to put the right tools in place to support them. Establishing a DevOps practice requires tools to support you throughout each phase of the DevOps cycle, whether you have an established DevOps center of excellence (COE) or are an application development team. The Pega Platform™ includes core DevOps capabilities to support Pega application development as well as a robust DevOps API to allow a mature DevOps team to integrate existing DevOps tools throughout the application lifecycle.

The figure below shows how Pega supports your DevOps lifecycle with both our provided capabilities and third-party tools using the Pega DevOps API.



Pega DevOps Approach

Deployment Manager

Restful APIs

Restful APIs

DEVELOPMENT FRAMEWORK: Model Driven Development

SOURCE CONTROL MGMT SYSTEM: Branching & Versioning

INTEGRATED DEVELOPMENT ENVIRONMENT: Developer Studio & App Studio

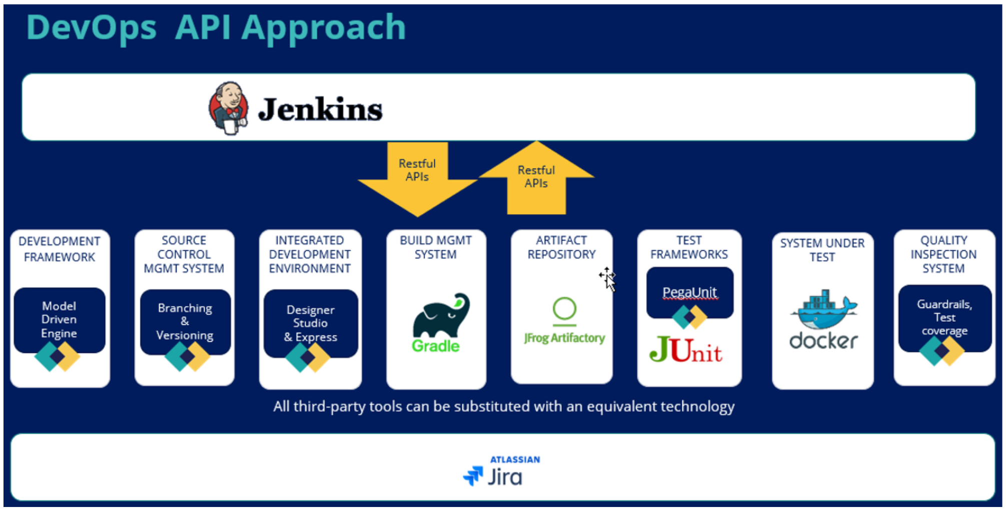
BUILD MGMT SYSTEM: Application packaging

ARTIFACT REPOSITORY: Filesystem

TEST FRAMEWORK: PegaUnit

QUALITY INSPECTION SYSTEM: Guardrails, Test coverage

Pega Agile Studio



DevOps API Approach

Jenkins

Restful APIs

Restful APIs

DEVELOPMENT FRAMEWORK: Model Driven Engine

SOURCE CONTROL MGMT SYSTEM: Branching & Versioning

INTEGRATED DEVELOPMENT ENVIRONMENT: Designer Studio & Express

BUILD MGMT SYSTEM: Gradle

ARTIFACT REPOSITORY: JFrog Artifactory

TEST FRAMEWORKS: PegaUnit, JUnit

SYSTEM UNDER TEST: docker

QUALITY INSPECTION SYSTEM: Guardrails, Test coverage

All third-party tools can be substituted with an equivalent technology

ATLASSIAN Jira

Discover Pega DevOps, our low-code approach

Pega effectively simplifies adoption of DevOps best practices for your organization, which helps teams without access to an established DevOps COE start their development off with DevOps in mind. Our approach focuses on empowering your team to employ continuous delivery, integration, and deployment best practices – across all your Pega applications.

Through each stage of the development cycle, Pega provides a built-in suite of DevOps capabilities designed to help you strengthen a culture of collaboration, drive success, and deliver high-quality applications.

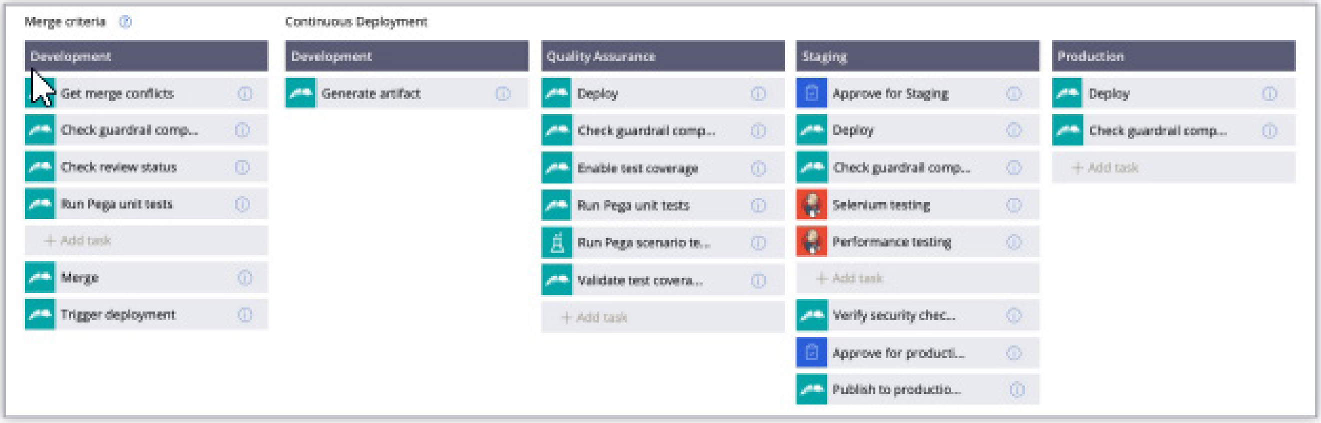
Automating with DevOps pipelines

Pega Deployment Manager is Pega's pipeline tool to automate your build and deployment process. It integrates seamlessly with Pega Infinity to quickly move from development to testing to deployment – all using Restful APIs. Deployment Manager coordinates your DevOps workflow throughout the application lifecycle.

It provides out-of-the-box, best-practice–based application deployment pipelines that are easily configured and customized through a low-code, model-driven experience. This enables you to get started quickly with a standardized, automated, and repeatable process, ensuring predictable, high-quality releases.

- These deployment pipelines come with built-in support for the following:
- Seamless branch validation and merge for developer changes
- Automated application packaging and distribution to higher environments
- Enforcement of quality gates such as guardrail scores and test coverage
- Ability to run Pega automated tests
- Adherence to security compliance via the application security checklist
- Power to roll back changes in an environment upon failure
- Built-in configurable notification for all pipeline events

Third-party pipelines, like Jenkins, are easily integrated into the pipeline process. You can use the same Restful API commands to integrate your pipeline with your Pega workflow. See how Pega Deployment Manager and Jenkins work together to deliver applications in a DevOps workflow in the graphic below.



Merge criteria

Development
- Get merge conflicts
- Check guardrail comp...
- Check review status
- Run Pega unit tests
- + Add task
- Merge
- Trigger deployment

Continuous Deployment

Development
- Generate artifact

Quality Assurance
- Deploy
- Check guardrail comp...
- Enable test coverage
- Run Pega unit tests
- Run Pega scenario te...
- Validate test covera...
- + Add task

Staging
- Approve for Staging
- Deploy
- Check guardrail comp...
- Selenium testing
- Performance testing
- + Add task
- Verify security chec...
- Approve for producti...
- Publish to productio...

Production
- Deploy
- Check guardrail comp...
- + Add task

Executing automated testing

Application testing is supported at three levels in the Pega Platform, through native features and integration with best-in-class tools. The most basic form of testing is unit testing, which supports continuous delivery of applications. Second is API-based integration testing that ensures applicable integrations function as expected. Finally, UI-based testing can be executed with Pega scenario testing, or through integration with leading UI testing tools such as Selenium.

Pega provides subscribers to the Pega Platform with access to testing tools plus integration with leading testing tools at each stage of testing.[2]

Pega testing tools

PegaUnit provides teams with the tools to quickly and easily create test cases to be incorporated into automated testing plans. Users may group tests into a test suite to run multiple cases – in a defined order that ensures high-quality releases.

ADK Simul8r is the new, low-code approach to simulating your integrations. It's provided as a component that can be added to Pega versions 8.3 and above. The component introduces a new "Simulation" rule type, available from the "Integration-Simulation" rule category. This new rule type provides you with a simple UI to capture 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. 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..

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.

Pega Selenium Starter Pack provides you with a Selenium-based UI test framework and sample UI tests that you can use to build a test automation suite for your Pega application. With this test framework, you can skip writing infrastructure code, take advantage of built-in diagnostics and testing capabilities, and focus on the most important thing, testing your Pega application! These test frameworks are built with maintenance and best practices in mind.

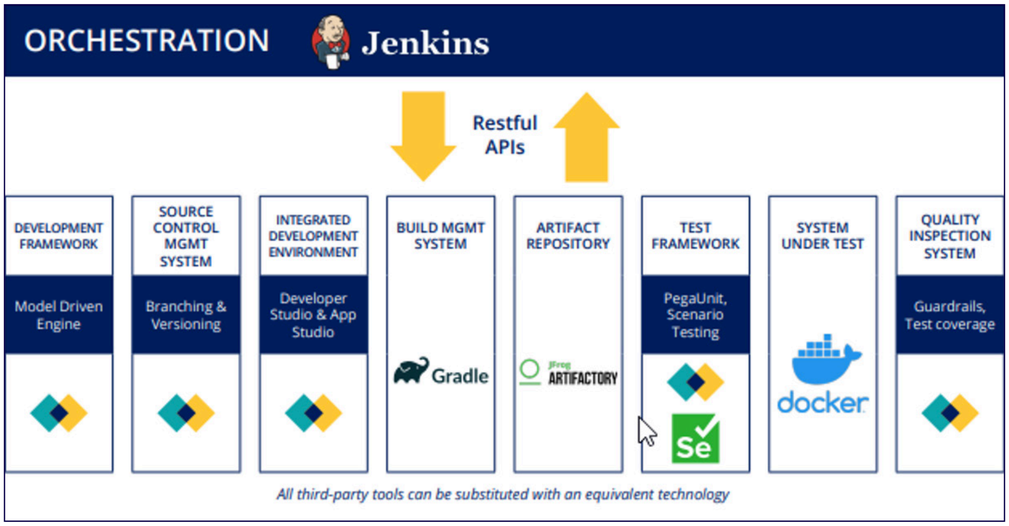
[2] See the Testing strategy section for additional details on automated testing.

Empowering the citizen developer

The power of Pega lies in democratizing application development by enabling citizen developers and business users to collaborate using our low-code Pega Platform™. Citizen developers are likely to use Pega App Studio to define a case lifecycle for addressing the latest user feedback without having to wait on IT – or a business user might use 1:1 Operations Manager to author strategies and offers for their marketing clients. This is where the real benefits of deep integration in Pega Infinity's Deployment Manager comes into play; it enables all stakeholders and contributors to be part of the DevOps process like no other products can.

Citizen developers can publish directly into an application deployment pipeline from App Studio as well as manage the process all the way to production with the appropriate Lead System Architect (LSA) review. All the while the deployment pipeline can be monitored by Deployment Manager's application administrator to handle any issues that arise. App Studio users are presented with a user-friendly pipeline representation from which they can monitor the progress of their changes and take any actions necessary. This integration empowers the citizens developers to participate in the deployment process directly, while application administrators or IT monitor and set the gates for deploying to production – tearing down silos across the board.

Using third-party tools

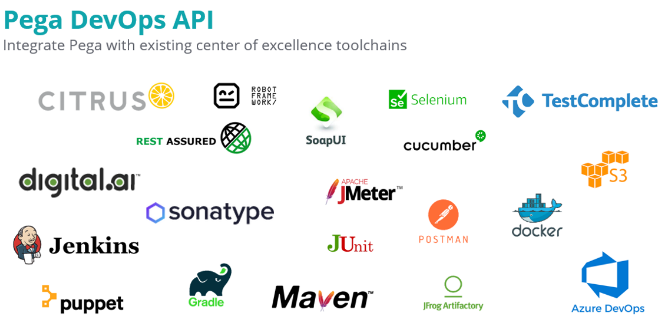


ORCHESTRATION Jenkins

Restful APIs

DEVELOPMENT FRAMEWORK	SOURCE CONTROL MGMT SYSTEM	INTEGRATED DEVELOPMENT ENVIRONMENT	BUILD MGMT SYSTEM	ARTIFACT REPOSITORY	TEST FRAMEWORK	SYSTEM UNDER TEST	QUALITY INSPECTION SYSTEM
Model Driven Engine	Branching & Versioning	Developer Studio & App Studio			PegaUnit, Scenario Testing		Guardrails, Test coverage
			Gradle	JFrog ARTIFACTORY	Se	docker	

All third-party tools can be substituted with an equivalent technology



Pega DevOps API

Integrate Pega with existing center of excellence toolchains

Pega's DevOps API allows integration for building a deployment pipeline using third-party tools to automate branch merging, application packaging and deployment, test execution, and quality metric enforcement. Pega Infinity provides an open platform with all the necessary hooks and service APIs for effective DevOps integration – by using popular third-party tools such as Jenkins and Microsoft Azure DevOps.

Pega's DevOps API enables you to use your existing tools to deploy Pega applications – so that your tried-and-true deployment processes will seamlessly work together. We make it easy for you to use your existing tools to deploy Pega applications faster and quickly realize the true value of our technology.

All third-party tools can be substituted with an equivalent technology.

Supporting multiteam parallel development

From its very inception, Pega was designed to support the needs of both small and large development projects, providing robust development capabilities – so you can gain the tools to facilitate process changes among multiple developers. The Pega design-time environment, called Dev Studio, includes a fully integrated version control system that supports check-in, check-out, and rollback at the individual model level.

The checked-out models stay on the server and are tested in context of other models that have been checked in – without affecting other users. During testing, there's no need to sync application branches to your local machine and then re-sync changes to a source control system. Additionally, Pega provides a change history for all application models so that users can restore previous versions of individual models.

How Pega supports your branch strategy

A good branch strategy is the basis of any good DevOps deployment. Branches allow your organization to develop software with multiple teams in parallel within a version-controlled environment. Pega supports this as part of a CI/CD pipeline, enabling you to use branches in development environments where multiple teams contribute to a single application. What's important to note is that the version control and branching functionality are built right into the Pega development environment. Your team can develop one feature in a branch while another team develops a feature in a different branch – even if they share the same base rulesets.

And you can still work with branches – even after you create branches and rules in them. For example, you can create reviews for branches to ensure that they are guardrail compliant. And after feature development is complete, you can merge the changes you made for all features into the base ruleset.

Pega supports check-in and check-out on rules within your Pega application. To avoid accidental rule changes or conflicts (which might result from multiple developers working on the same rule), perform a check-out so that you can lock and safely change the rule. By checking out a rule before editing, you avoid unwanted rule changes – saving you time so that you can maintain a better-quality application.

Ensuring quality with application testing

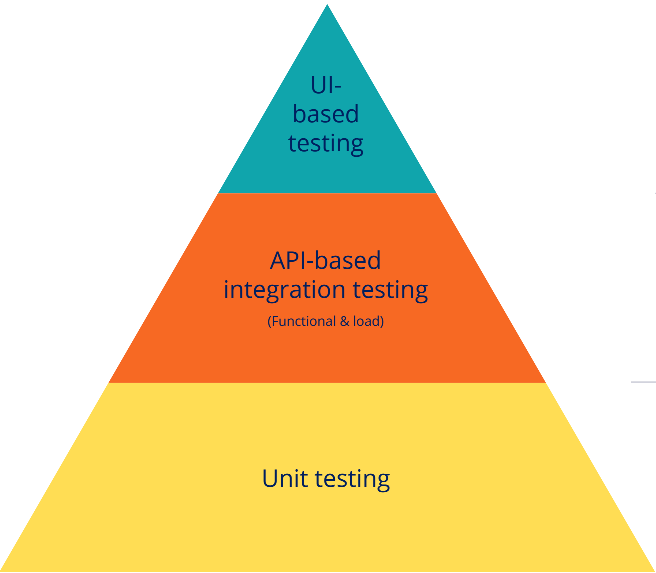
Having an effective automation test suite for your application in your continuous delivery DevOps pipeline is important. It ensures that the features and changes you deliver to your customers are of high quality and do not introduce regressions. Pega offers several capabilities to perform continuous testing and deliver high-quality applications as part of your CI/CD deployment pipelines. These application testing capabilities enable both developers and business users to launch and maintain automated tests that cater to various testing needs, all while continuously monitoring the application quality to take corrective action early in the cycle.

Testing strategy

The recommended test automation strategy for Pega applications is to:

- Develop your automation test suite based on industry best practices for test automation
- Create your test cases when you are developing your application and its rules – so when your application is complete, it is ready for automated testing
- Build up your automation test suite by using Pega Infinity capabilities and industry test solutions
- Run the right set of tests at different stages of your delivery pipeline
- Test early and test often

To make industry best practices for test automation easier to understand, they can be presented as a test pyramid.



What are you testing?	Tools
• Application functionality via user interface; real-world user experience • In Pega, this may be channel, portal, harness, or section	PEGA Scenario Test Selenium ROBOT FRAMEWORK cucumber PEGA Selenium Starter Pack TestComplete
• How a collection of integrated units function together • In Pega, this may be an API, case, or a specific process	PEGA Dx API CITRUS SoapUI REST ASSURED Simul8r APACHE JMeter POSTMAN
• The smallest piece of logic, code, or functionality that make up your application • In Pega, the smallest piece of functionality are Pega rules; unit testing in Pega means testing a rule in isolation	PEGA Pega Unit JUnit

Test types at the bottom of the pyramid are the least expensive to run, easiest to maintain, take the least amount of time to run, and should represent the greatest number of tests in the test suite. Test types at the top of the pyramid are the most expensive to run, hardest to maintain, take the most time to run, and should represent the least number of tests in the test suite. The higher up the pyramid you go, the higher the overall cost and the lower the benefits.

With all the tools and integrations that Pega provides, your organization can effectively and seamlessly perform automated testing and achieve benefits, including:

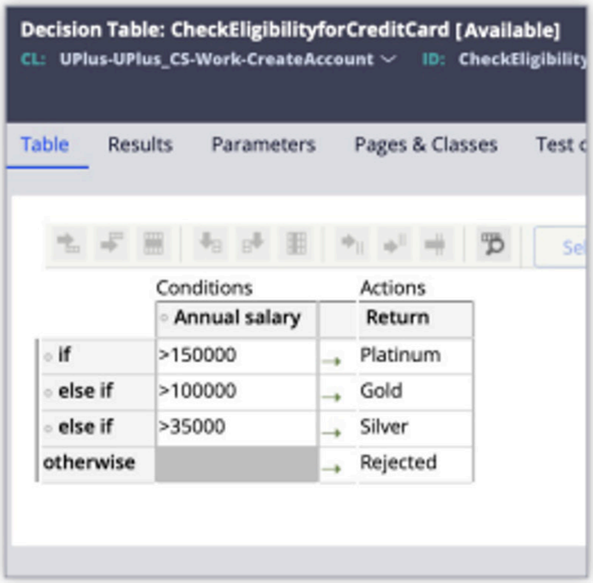
- Reducing costs as tests can be created, then executed again and again
- Eliminating the need for human intervention in the testing process
- Ensuring your tests are consistent from version to version
- Deploying faster since automated tests execute much faster than manual tests

Unit testing

For each created or updated rule, developers create fast unit test cases with support for setting up test data and mocking or stubbing external integrations. These unit tests run very fast and can quickly and easily be run on every check-in, increasing the confidence in the implementation and reducing the burden on regression testing. And as with all things Pega – there is never a need to write code.

Left: Decision table used for performing complex logic within a Pega case.

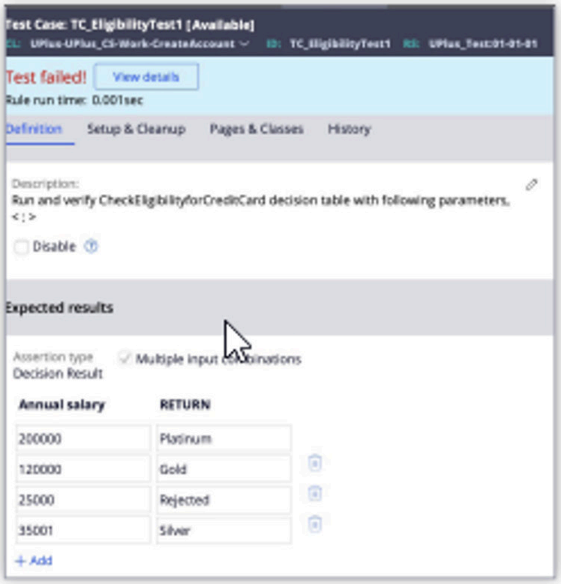
Right: Report on test case for the decision table.



Decision Table: CheckEligibilityforCreditCard [Available]

CL: UPlus-UPlus_CS-Work-CreateAccount ID: CheckEligibility

	Conditions	Actions
	Annual salary	Return
if	>150000	Platinum
else if	>100000	Gold
else if	>35000	Silver
otherwise		Rejected



Test Case: TC_EligibilityTest1 [Available]

CL: UPlus-UPlus_CS-Work-CreateAccount ID: TC_EligibilityTest1 RS: UPlus_Test:01-01-01

Test failed!

Rule run time: 0.001sec

Description:
Run and verify CheckEligibilityforCreditCard decision table with following parameters.
<;>

□ Disable

Expected results

Assertion type
Decision Result

☑ Multiple input combinations

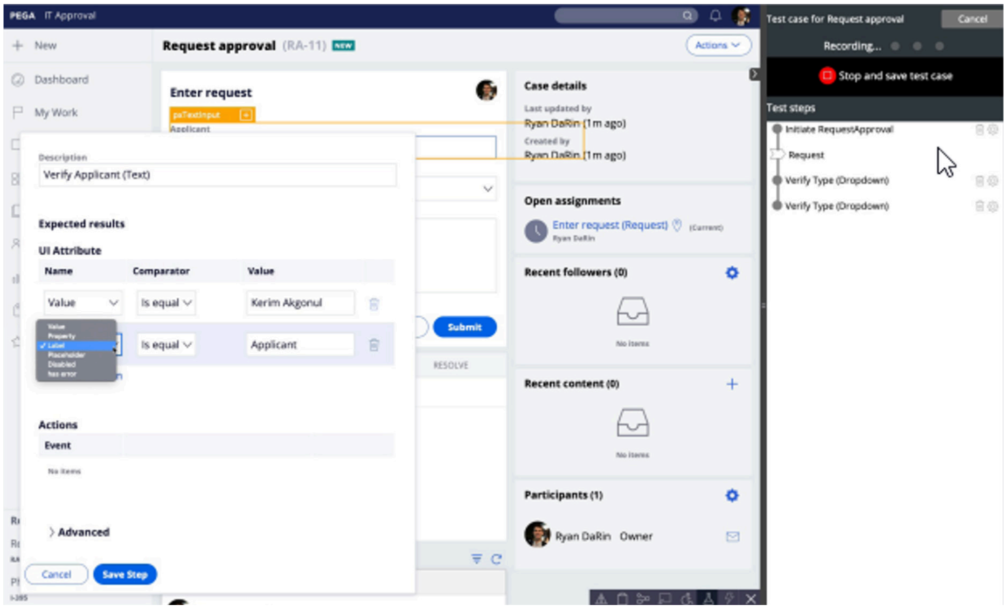
Annual salary	RETURN
200000	Platinum
120000	Gold
25000	Rejected
35001	Silver

Pega provides unit tests for all critical application logic and expression rules, including:

- Activities
- Collections
- Data transforms
- Decision trees
- Flows
- Report definitions
- When rules
- Case types
- Data pages
- Decision tables
- Declare expressions
- Map values
- Strategies

End-to-end scenario testing

Without writing a single line of code, business users can utilize the automated scenario testing capability to verify the end-user experience. The end user input is captured as the test data that will be executed on subsequent test runs for UI and portal testing. This means that your employees who will be using the application can be part of the process of creating automated regression test cases.



Description: Verify Applicant (Text)

Expected results

UI Attribute

Name	Comparator	Value
Value	Is equal	Kerim Akgonul
Label	Is equal	Applicant

Last updated by Ryan DaRin (1m ago)

Created by Ryan DaRin (1m ago)

Enter request (Request) (Current)

Ryan DaRin Owner

Test steps:
- Initiate RequestApproval
- Request
- Verify Type (Dropdown)
- Verify Type (Dropdown)

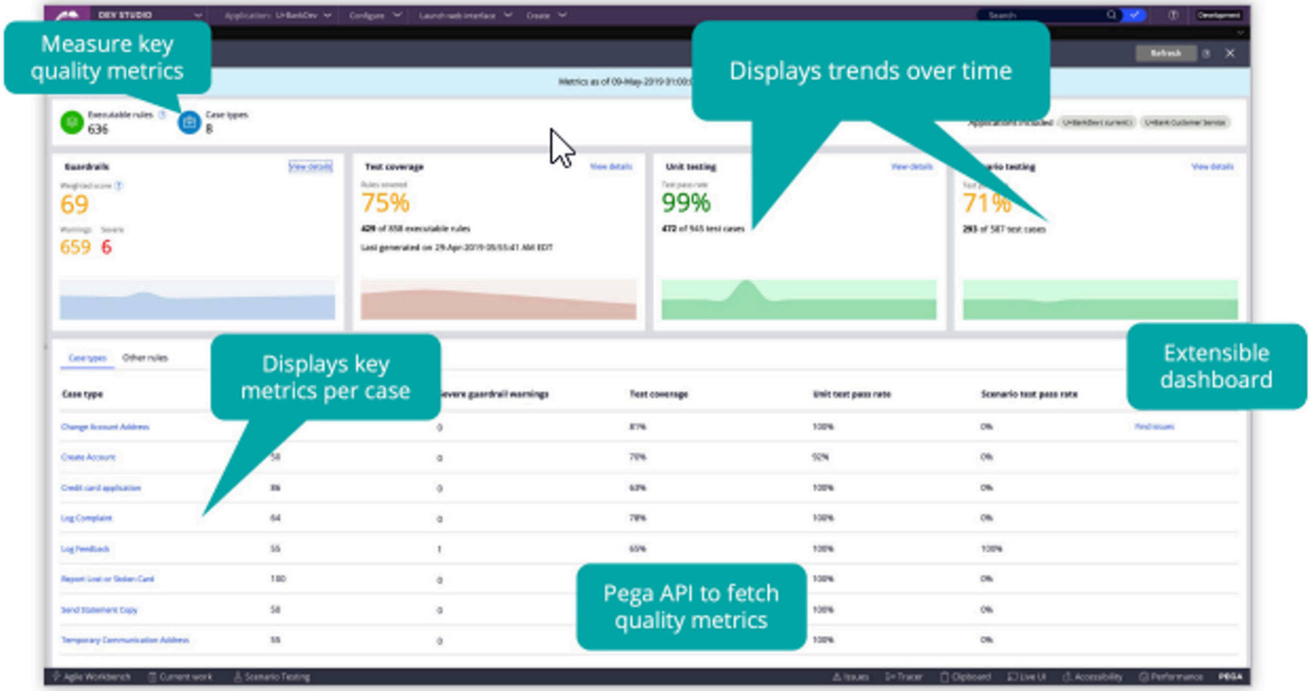
These business users can test either a specific case type or an entire portal by clicking "Scenario Testing" in the run-time toolbar to open the test recorder.

When you use the test recorder and hover over a testable element, an orange highlight indicates that the element can be tested. Interactions are recorded in a visual series of steps and the execution of a test step can include a delay.

These unit tests and scenario tests verifying the user experience can all be automatically run in the deployment pipeline to ensure that updates do not break your Pega application and lead to a production stoppage. Additionally, Pega testing provides details as to whether your testing succeeded for each test case. Deployment Manager supports executing these tests and gating the pipeline based on the test results, making it easy to incorporate this into your deployment process.

Understanding errors with the Application Quality dashboard

Many automated test applications provide a detailed text-based list of testing errors. Yet Pega makes it even easier to understand the error and take action – all by providing more details in a graphic format. Pega Infinity includes our Application Quality dashboard, which quickly identifies areas within your application that need improvement by displaying metrics related to your application's health.



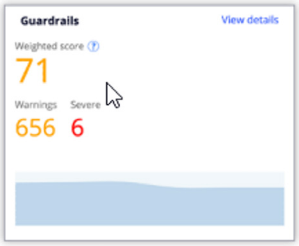
Case type		Severe guardrail warnings	Test coverage	Unit test pass rate	Scenario test pass rate	
Change Account Address		0	87%	100%	0%	Find issues
Create Account	58	0	79%	92%	0%	
Credit card application	86	0	63%	100%	0%	
Log Complaint	64	0	78%	100%	0%	
Log Feedback	55	1	65%	100%	100%	
Report Lost or Stolen Card	180	0		100%	0%	
Send Statement Copy	58	0		100%	0%	
Temporary Communication Address	55	0		100%	0%	

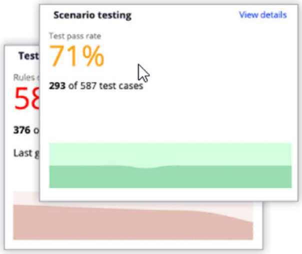
To better help you understand the health of your Pega applications, the dashboard includes four charts:

- Guardrails or compliance score
- Test coverage
- Unit testing
- Scenario testing

Guardrails

Guardrails are guidelines for achieving optimal performance, reusability, and maintainability in your application. A low guardrail score means that your application is not in compliance with Pega Infinity applications' best practices. This may introduce brittleness into your application and make it harder to upgrade. Using the dashboard, you may quickly view your application's guardrail compliance score and see the number and severity of guardrail violations that were found in your application. You can then improve your compliance score and overall quality by investigating and resolving the violations.



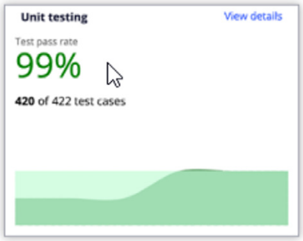


Test coverage

Test coverage is a chart that is used to improve the quality of your application and identify how many executable rules are covered by tests. As rules are executed in the testing, the coverage report indicates whether test cases exist. The user is provided a detailed list of rules that include a test case – so that a low score can be raised quickly. This coverage report also shows how the coverage metric changes over time – by running test coverage sessions, merging reports, and analyzing coverage trends.

Unit testing

Unit tests are typically developed by the application developers along with the new functionality. They validate the smallest meaningful units of functionality and logic that can be tested quickly and provide fast feedback. Getting a high pass rate, tracked by the unit testing metric, is critical and meaningful as it ensures that all core building blocks of the application are working as designed – simplifying the testing that is necessary later in the deployment process.



Scenario testing

Scenario testing executes tests against a user interface to verify that the end-to-end scenarios are functioning correctly. Pega supports built-in scenario testing and includes a built-in test recorder to create functional and useful acceptance tests – rather than writing complex code. This metric tracks the pass rate of these scenario acceptance tests over time and is a great way to get an understanding of how effectively the end-to-end functionality of the application is being tested over time.

Putting it all together

DevOps is imperative for the future of application development for both professional and citizen developers. Pega has built a flexible approach to applying DevOps to your Pega investment that allows both low-code, out-of-the-box capabilities and integration into your existing DevOps suite of tools.

We take this approach because DevOps is also about the people and collaboration, not just the tools and technology. You must also account for the cultural changes that embracing DevOps will bring. One of the biggest barriers to adopting these practices is figuring out where to start and which tools to use to implement these practices. Use the built-in Pega DevOps as well as test automation capabilities and tools like Deployment Manager to deploy Pega applications faster – reducing costs and increasing efficiency. Doing so will enable your organization to realize the value from your Pega applications in a matter of weeks – not months or years.

PEGA®

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