ACCEPTANCE TESTING

UAT INITIATION & DESIGN

**Testing in the Build environment**

Test a new application in the Build environment before migrating the application to a test or production environment. Testing in the Build environment enables you to verify that basic functionality and interfaces work correctly, and that performance is acceptable.

1. Run functional tests, to test specific features from the end-user perspective.
2. Use the Performance tool to measure the performance of the application. For more information, see [Track system utilization for a requestor session with Performance Analyzer](https://pega-prod.zoominsoftware.io/bundle/platform-88/page/platform/system-administration/performance-statistics-pal.html).
   * Prior to extending your site-specific Pega implementation, establish a performance benchmark (baseline) by using the Performance tool. This allows subsequent, iterative performance tests against the baseline to help identify any degradation in performance resulting from development efforts.
   * Use the Performance tool to check the performance of the following features:
     + Search
     + Account selection
     + Loading of components
     + Kickoff of all service intents. Automated scripts are recommended for this unit testing, but are not required.
   * Save the test results so that you can compare them to future test results to determine whether an application update has a performance impact.
3. Verify that the out-of-the-box reports and your custom reports run successfully, and that they show your implementation layer data, rather than the default demonstration data. This can be an automated test.
4. Test all integrations, both independently and with associated integrations.

Test integrations for any optional Pega Government Platform components and other applications that you plan to use. See the product documentation for the component or application to determine which product components to test.

1. Test security. Test the most common roles to ensure that the required access groups are configured and point to the correct software version.

**Testing in the test or production environments**

After you import your application to a test or production environment, test the application in the new environment to verify that it works correctly in that environment.

1. Verify that the source and destination files are the same.
2. Run functional tests to test specific features from the user perspective.
3. In the test or production environment, run the Application Guardrails Compliance Score to ensure that the application meets guardrails.
4. Verify that the out-of-the-box reports and your custom reports run successfully, and that they show your implementation layer data, rather than the default demonstration data. This can be an automated test.
5. Test all integrations, both independently and with associated integrations.

Test integrations for any optional Pega Government Platform components and other applications that you plan to use. See the product documentation for the component or application to determine which product components to test.

1. Verify that the integrations point to the correct system of record, and not to the system of record for the Build environment.
2. Test security. Test the most common roles to ensure that the required access groups are configured and point to the correct software version. Use these common roles in your smoke tests. See step [8](https://docs.pega.com/bundle/pega-government-platform/page/common/test/test-new-app.html#Testing_in_the_Test_or_Production_environments__step_rfb_5vy_tnb).
3. Run a smoke test to compare the source and destination environments. Verify that all tests that pass in the build environment also pass in the test or production environment. If anything fails, compare the environments to determine whether it is a difference in environment that causes the test to fail. If the environment causes a failure, either fix the issue that causes the failure or adjust the test as appropriate for the new environment.
4. Run performance tests to verify that performance meets expectations. Pega recommends automated performance testing. Save the results so that you can compare them to future performance test results to determine whether an application update has a performance impact.

**Testing in the UAT environment**

After you complete testing in a Test environment, it is common to perform User Acceptance Testing (UAT) in a designated UAT environment, which could be a preproduction environment. UAT ensures that users will be able to successfully complete their work and meet business objectives.

**Note:** Organizations that use Scrum for application development will complete less formal UAT as part of each sprint cycle.

1. Verify the integrity of the UAT environment.
2. Have the end-users (or business analysts acting as end-users) run scripts to test all scenarios, including boundary and exception testing. The end-users (trainers, managers, and directors), must then perform the following steps during UAT:
   1. Verify that there are no major issues.
   2. Review changes to understand the features.

**On this page**

* [**Testing in the Build environment**](https://docs.pega.com/en-US/bundle/pega-government-platform/page/common/test/test-new-app.html#ariaid-title2)
* [Testing in the test or production environments](https://docs.pega.com/en-US/bundle/pega-government-platform/page/common/test/test-new-app.html#ariaid-title3)
* [Testing in the UAT environment](https://docs.pega.com/en-US/bundle/pega-government-platform/page/common/test/test-new-app.html#ariaid-title4)

**About Pegasystems**

Pegasystems is the leader in cloud software for customer engagement and operational excellence. If you've driven a car, used a credit card, called a company for service, opened an account, flown on a plane, submitted a claim, or performed countless other everyday tasks, chances are you've to interact.

