



10

BEST PRACTICES FOR UPGRADING PEGA APPLICATIONS

1. PERFORM A FULL TECH-STACK UPGRADE



Details: When performing Pega Upgrade, check the platform support guide from Pegasystems and upgrade the database and application server along with it.



Benefits: While testing the application in the staging environment, it's easier to test the upgraded environment at the same time.

Also, upgrading the application server and database as per the support guide will improve stability and performance.

2. LEVERAGE THE ADDITIONAL SERVER IN PEGA CLOUD UPGRADE



Details: If your application is hosted in Pega cloud, Pegasystems clones the staging server and upgrades the platform on the cloned server.



Caution: Though the platform is upgraded, the customer's application must be verified and updated for the latest version. Test the application in the cloned server and apply all the necessary fixes. Once these fixes are applied, it's easy to upgrade all environments to the latest version.

3. TIME IT RIGHT



Details: Your upgrade process starts from the development environment, staging environment, and production environment. It may take a two-week sprint time or few months, depending on your application complexity.



Recommendation : Plan the upgrade during the slow time of the year.

4. LEVERAGE AUTOMATED TESTING



Details: If you do not have automated testing, working on this will help speed up the upgrade.



Reason: Upgrade demands more testing than other tasks. To reduce the testing timeline, implement automated testing before upgrade.

5. PLAN TO KEEP DOWN-TIME MINIMAL



Details: You can plan things like the out-of-place upgrade to reduce downtime.



Reason: Though upgrades are faster in lower environments, the production environment may have additional delays due to an enormous amount of data. Proper planning needs to be done for an efficient upgrade.

6. VALIDATE YOUR PRODUCTION UPGRADE PLAN



Details: Keep a well-documented plan for production to reduce impact. If you have multiple channels and listeners, plan properly to start one by one.



Reason: Listeners and Channels may create many cases in the background. The best bet is to validate the upgrade with manual case creation, followed by cases created by channels/listeners. This also ensures each channel and listener is verified on the upgraded environment.

7. VALIDATE AND RESTORE YOUR DATABASE BACKUP



Details: In a worst-case scenario, if you have to restore the backup in production – doing that in lower environments will help avoid any mishaps.



Caution: There are times backups may miss data and go useless.

8 . START WITH A THIN RULE BASE



Details: Continuous Development increases the number of ruleset versions. Skimming to a major ruleset will eradicate the unwanted rules and keeps your rule base slim.



Recommendation: Plan for major ruleset instead of minor for a faster compilation and removal of blocked rules.

9. SYNCH YOUR ENVIRONMENTS



Details: Active development makes rules different between environments. Also, there are production rulesets that may not be available in lower environments.



Caution: Your new upgraded environment should be similar to the current production environment. If you do not take proper care, you may introduce unwanted features in the upgraded environment.

10 . TRACK THE PERFORMANCE STATISTICS



Details: Every new version may have new settings to improve the performance to watch for.



Recommendation : Review the different logs available on the platform and app server for proper tuning. Tuning the environment accordingly will help for an efficient system.



HAVE QUESTIONS?

Contact our team. We'd love to help you.

4550 New Linden Hill Road, Wilmington, DE-19808 +1 844 EVONSY

innovation@evonsys.com