BEST PRACTICES

for Continuous Integration with Bonita BPM

Already considered a mainstream approach in software development projects, Continuous Integration is just as important when developing BPM applications with Bonita BPM.

Here's what we recommend.

1. Maintain a single, "version controlled" source repository

Keeping process design changes locally on developer's laptops is risky, hinders collaboration and leads to high tension during the integration phase – when it is not certain that all the fragments will come together successfully.

Leverage the Subversion integration available in Bonita BPM Studio to connect and synchronize your processes being developed with a centralized version controlled repository.

2. Frequently commit process design changes

The longer a developer keeps changes on a local machine, the higher the chance of running into a conflict with other team members' work. The key to resolving problems quickly is to find them quickly; undetected conflicts become harder to solve with time.

With version control in place, introduce the discipline to commit changes to the central repository frequently, multiple times a day.

3. Integrate new changes with an automated build

Even a centralized repository is no guarantee that new changes won't break the existing design. Detection of conflicts is still dependent on the individual developers running the latest version of the process on their machine.

Bonitasoft

Be Efficient

Bonita BPM provides the Workspace API to automatically compile your process models. Set up an automated integration environment with a continuous integration system (like Jenkins) and the Bonita Workspace API, to monitor the version control system for changes and trigger process compilation thereby helping detect integration issues early.

4. Own the process end to end

Discourage "throwing things over the wall" by encouraging more ownership within the team. Each team member should own the process end-to-end, ensuring that any enhancements made to the process are frequently committed and successfully integrated by the automated build.

5. Validate each integration with sanity tests

Process compilation during integration validates basic integrity but it does not ensure that the process application as a whole works as expected.

Explore the features available in your BPMS to develop tests which can validate process behavior. With Bonita BPM, use the best practice of abstracting business/integration logic using the Bonita BPM connectors which can then be tested. You can also use web testing frameworks (like Selenium) to automate end to end. This helps build more confidence in the development process and detection of defects.

6. Create stable and fast builds

Ensure the build process is stable and fast. Random build failures or long running builds reduce confidence, affect ownership and the discipline to frequently commit and keep the build "green."

7. Enable easy access to the deployment artifacts

Traditional methods of integration do not support early testing and business acceptance as they push them to the end of the development cycle. With continuous integration, you can involve your stakeholders early in the process by providing easy access to the successful builds, readily deployable to test environments.

8. Automate deployment of good builds

Teams often maintain several test and production environments and it is a highly repetitive and manual task to deploy builds to these environments.

Bonita BPM allows you to maintain multi-environment process configuration. Utilize this feature and automate the deployment of successful builds to various environments. This not only reduces the manual burden but also optimizes the development to delivery cycle.

For more information, visit our web site www.bonitasoft.com.