Software updates deployment and release procedures

Deployment is a shift of software from one controlled environment to another. On the other hand, releases are a collection of changes delivered to the Production environment for users to experience. Applications need several updates, security patches, and code changes. Deploying them across platforms and environments requires proper management of releases.

The following process should be followed to deploy software changes:

Preparation

During this stage, developers test if the code is running smoothly and ready to deploy. Then, they must gather all the code along with all the configuration files, libraries, and resources needed for the seamless functioning of the application. These items combined create a single package to be released. Important activities to carry out during preparation stage:

- Notify all the stakeholders about the upcoming deployment.
- Identify and inform all the collaborators to reduce friction between the teams involved in the software development lifecycle (SDLC) process.
- Check third-party software requirements to ensure everyone knows their uses to minimize any issues in the future.
- Map out the deployment process to ensure everyone is on the same page.
- Have a rollback plan ready in case any critical issues occur during deployment.
- Establish key performance indicators (KPIs) to measure the deployment’s effectiveness and determine whether it succeeds or fails.

Testing

The objective of the testing stage is to validate the software update or change before it reaches the deployment stage. During this phase, the update is deployed to a test environment and subjected to testing. Developers then review the results of the testing to fix any errors before deploying the update to a live environment. Key activities carried out include:

- Conducting unit tests on components of the software affected by the change to verify that the results meet user requirements.
- Integrating unit tests with the continuous integration (CI) tools where available to automatically build and verify each portion and remove bugs easily.
- Creating a staging environment similar to target the production environment and run tests to ensure the code works as intended.
- Run end-to-end tests to check the application’s workflow and its compatibility with other components like hardware and network connectivity.

- Seek feedback from stakeholders including requesting end users to undertake functional testing to determine whether the code is ready for production.
- Use smoke tests after deployment to make sure the software is running smoothly.

Deployment

Once the update is thoroughly tested, you can deploy it to the live environment. At this stage, the main task is to look out for any bugs and errors that might occur on the live server and fix those to deliver the best experience possible. Key activities carried out include:

- Deploy to the production environment to let users interact with the software.
- Monitor performance closely by matching it with the predetermined KPIs and checking for errors.
- Automate rollbacks so you can go to the previous version if the release is unsuccessful.
- Check environment health to identify potential issues with the operating system (OS) or database.
- Keep track of logs to gain insight into how the software is running and identify any security threats.
- Document everything to maintain consistency when you want to change the new version.

The software deployment process can be manual or automated, but it should be done at the least disruptive time in an organization’s work routine.

DevOps tools such as continuous delivery software can enable developers to generate deployment-ready code quickly. This means software releases remain in a deployable state, enabling automatic deployment to production within seconds.

Revision #1
Created 8 October 2025 10:03:37 by RISA
Updated 8 October 2025 10:06:34 by RISA