Leadership
Release Management
Continuous Integration

October 9, 2013
Software Release Management

Software release management encompasses the identification, packaging, and delivery of the elements of a product. This includes executable program, documentation, release notes, and configuration data.

Benefits

- Helps make software builds simple, quick and reliable
- Helps take care of configuration management.
- Helps drive software quality.
- Helps optimize development and QA time.
- Speeds up time to market.
Release Mgmt in Repository
Software Release Management

Software release management encompasses the identification, packaging, and delivery of the elements of a product. This includes executable program, documentation, release notes, and configuration data.

Benefits

- Helps make software builds simple, quick and reliable.
  - This is achieved by employing the best tools for the job. This means understanding all the various build tools, seeing how they integrate with the systems that already exist in the workplace, and making an informed choice.
Software Release Management

Software release management encompasses the identification, packaging, and delivery of the elements of a product. This includes executable program, documentation, release notes, and configuration data.

Benefits

- Helps take care of configuration management.
  - Software configuration management (SCM) is the task of tracking and controlling changes in the software, part of the larger cross-discipline field of configuration management.”
  - SCM practices include revision control and the establishment of baselines. If something goes wrong, SCM can determine what was changed and who changed it.
  - If a configuration is working well, SCM can determine how to replicate it across many hosts.
Software Release Management

Software release management encompasses the identification, packaging, and delivery of the elements of a product. This includes executable program, documentation, release notes, and configuration data.

Benefits

- Helps optimize development and QA time.
  - By giving the dev team the feedback on the quality of their code and telling them where they’re going right and going wrong, we’re helping them target their efforts.
  - By providing these solutions for them, doing the builds, configurations and releases, the developers can get busy doing the stuff they’re skilled at doing.
  - For the QA team, we’re finding defects and failing releases before the releases even get to them.
Software Release Management

Software release management encompasses the identification, packaging, and delivery of the elements of a product. This includes executable program, documentation, release notes, and configuration data.

Benefits

- Speeds up time to market.
  - Making builds quicker, easier and more reliable, accelerates the process of fine tuning code quality.
  - This also helps identify defects before QA starts and the process of releasing the software into production quicker and simpler.
  - This saves significant development, QA and operations and so our new, higher quality software, can be released efficiently into production.
Software Release Management

Software release management encompasses the identification, packaging, and delivery of the elements of a product. This includes executable program, documentation, release notes, and configuration data.

Benefits

- Helps drive software quality.
- Thanks to the Continuous Integration process, and the tools that have been built around it, it’s now possible for us to build software every single time a piece of code is checked in, run a suite of unit tests, analyze the code for lazy programming and report on the amount of test coverage a project has.
- Jenkins is an open-source continuous integration software tool written in the Java programming language for testing and reporting of defects in an automated fashion.
Continuous Integration

Continuous integration (CI) is the practice, in software engineering, of merging all developer working copies with a shared mainline several times a day. Its primary purpose is to prevent integration problems.

Continuous Integration with automated test execution has seen broad adoption in recent years. The ideas behind Continuous Integration have changed how companies look at Build Management, Release Management, Deployment Automation, and Test Orchestration.

Jenkins is a popular industry CI tool which provides an easy-to-use continuous integration system, making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. The automated, continuous build increases the productivity.
Continuous Integration Example 1
Basic configuration.
Continuous Integration Example 2
Basic configuration plus automated testing.