

Continuous Delivery

Summary

Getting software released to users is often a painful, risky, and time-consuming process. This tutorial sets out the principles and technical practices that enable rapid, incremental delivery of high quality, valuable new functionality to users. Through automation of the build, deployment, and testing process, and improved collaboration between developers, testers and operations, delivery teams can get changes released in a matter of hours—sometimes even minutes—no matter what the size of a project or the complexity of its code base.

In this tutorial we take the unique approach of moving from release back through testing to development practices, analyzing at each stage how to improve collaboration and increase feedback so as to make the delivery process as fast and efficient as possible. At the heart of the tutorial is a pattern called the deployment pipeline, which involves the creation of a living system that models your organization's value stream for delivering software. We spend the first half of the tutorial introducing this pattern, and discussing how to incrementally automate the build, test and deployment process, culminating in continuous deployment.

In the second half of the tutorial, we introduce agile infrastructure, including the use of Puppet to automate the management of testing and production environments. We'll discuss automating data management, including migrations. Development practices that enable incremental development and delivery will be covered at length, including a discussion of why branching is inimical to continuous delivery, and how practices such as branch by abstraction and componentization provide superior alternatives that enable large and distributed teams to deliver incrementally.

Course Outline

Audience: ✓ Software Developers
✓ Quality Analysts
✓ Software Testers
✓ Enterprise Architects
✓ Technical Managers
✓ Systems, Network and Database Administrators

Duration: 1 day

Pre-Requisite: None

- Learning Objectives:*
- ✓ Understand the importance of fast feedback on the production readiness of an application throughout the delivery process, and how to get it.
 - ✓ Understand how to improve collaboration between people involved in delivering software.
 - ✓ Understand the principles and practices that enable continuous deployment, including data management.
 - ✓ Experience automating testing and deployment.
 - ✓ Experience evolving an application without using version control branches.
 - ✓ Understand managing continuous delivery with large and distributed teams.
 - ✓ Understand how to transform your organization's delivery process to enable continuous delivery.

Morning

- Introduction of trainers and participants
- Learning objectives
- Case studies
- Introduction to continuous deployment, deployment pipelines, principles, and prerequisites
- Automated tests, where they fit in the delivery process
- Refresher on continuous integration
- Exercise to understand fundamentals of continuous delivery

Afternoon

- Building deployment pipelines for component-based architectures
- Dependency management
- Going live, A/B testing, business metrics
- Data management and migration
- Environment and infrastructure management: provisioning, management, monitoring
- Managing continuous delivery continuous improvement: modeling, evolving the pipeline, your organization's delivery process.