Subversion for Software Developers

Course Summary

Description

Introduce students to the basic concepts of version control and their function. Provide context to what challenges version control systems are designed to address. Show how software developers can use different tools to create and maintain sets of software patches. Demonstrate how patches can be applied and organized to create a historical set of changes that describe a project’s history

Introduce Subversion and explain how it acts as a centralized version control system. Cover how a repository is organized and the ways it is possible to interact with data. Explain how Subversion differs from other version control systems, such as Git, Mercurial, or Bazaar.

Explore common Subversion operations such as project/repository creation, source checkout, adding/editing files, and committing changes. Discuss version control best practices and workflows. Provide hands-on exercises so that students become comfortable working with version control using both GUI tools and the command line. Describe how Subversion integrates with other tools and can be integrated with bug/issue tracking management and continuous testing servers

Topics

- An Introduction to Version Control: Getting started with version control and Subversion
- Subversion
- Subversion Tools
- Beyond Basics

Duration

Three Days

Subversion for Software Developers

Course Outline

I. *An Introduction to Version Control: Getting started with version control and Subversion*

A. What are the challenges of collaboration?
B. Distributed workflows
C. Tracking changes and history
D. Providing a framework for developers to experiment safely
E. What is version control?
F. A solution for the challenges of collaboration
G. A mechanism for working with a distributed group
H. A system for tracking changes
I. Types of version control systems: distributed versus central
J. What is Subversion?
K. Subversion History
L. Subversion Architecture

- Lab Tasks: Diff/Patch Operations
  - Use command line and GUI tools to compare changes to source code files
  - Create diff files that can be exchanged with other software developers for collaboration and code review
  - Show how diff files can be used to create snapshots of a project and it is then possible to step forward or backward through the history

II. *Subversion*

A. Installation
B. Creating Repositories and Projects
C. Basic Operations
D. Create
E. Checkout
F. Commit
G. Update
H. Add Edit
I. Delete
J. Rename
K. Move
L. Status
M. Diff
N. Revert
O. Log
P. Tag
Q. Branch
R. Merge
S. Resolve
T. Lock

- Lab Tasks: Subversion Core Operations
  - Show developers the most important Subversion operations
  - Work through examples of how to create repositories, add files to source control, generate branches, and perform merges
  - Work with students to create personal project repositories and experiment with Subversion’s core feature set

III. *Subversion Tools*

A. TortoiseSVN: Exposing the power of Subversion inside of the Windows Shell
B. SubClipse: Accessing Subversion from Eclipse
C. Versions: A Mac OS X SubVersion GUI
D. What additional capabilities the GUI brings to Subversion

- Lab Tasks: GUI Tools
  - Look at additional utilities included in TortoiseSVN and SubClipse (Versions/ SubClipse for Mac OS X users) and how they become a part of the larger Integrated Development Environment/Operating System

Subversion for Sofware Developers

Course Outline (cont.)

IV. ***Beyond Basics***

A. Workflows
B. Managing software releases
C. Discrete version management for applications teams
D. Management strategies for web teams
E. Integrating Subversion into web based tools: Bug/issue and time tracking utilities and beyond
F. Security best practices: Securing access to a repository and determining rights/privileges
G. Protecting repositories on disk
H. Subversion over SSH
I. Subversion over HTTP (Apache)
J. Subversion in a world of distributed version control
K. Continuous integration testing and software validation

- Lab Tasks
  - Demonstrate and practice advanced operations related to workflow
  - Show how Subversion can be integrated with external tools, such as bug/issue trackers
  - Look at the methods for securing access to a repository