Version Control Workshop

Neil Fraser
Google
1600 Amphitheatre Parkway
Mountain View, CA, 94043
fraser@google.com

ABSTRACT

This three hour workshop takes participants on a tour of popular Version Control systems, particularly Subversion and Git. By the end of the workshop each participant will be proficient in using both of these systems. The focus is on solving real-world problems, such as resolving conflicting changes or rolling back a change. This workshop is not about the theory or academic underpinnings of such systems. Participants are required to bring a Macintosh, Linux or Windows laptop.

Categories and Subject Descriptors

D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement – *Version control*

I.7.1 [**Document and Text Processing**]: Document and Text Editing – *Version control*

General Terms: Management.

Keywords: Subversion, Git.

1. INTRODUCTION

Version Control systems have become an indispensable tool for organizations that maintain digital resources. Often used for maintaining source code, Version Control is also used for text documents, engineering diagrams, content management, and wikis. Version Control allows multiple people to work on a project simultaneously without fear of overwriting each other's work. Version Control also allows one to trace the origin of any change back to the original author and the context in which it was made.

Despite its utility, Version Control remains underused. Due to their steep learning curves, owners of smaller projects sometimes shy away from the overhead of setting up Version Control. Likewise, many users of Version Control do not exploit the more advanced features such as personal branching due to unfamiliarity.

Copyright is held by the author/owner(s).
DocEng'11, September 19–22, 2011, Mountain View, California, USA.
ACM 978-1-4503-0863-2/11/09.

GOALS

This workshop is intended to teach how to use Version Control in small to medium-scale environments. Unlike most documentation, the workshop is not structured around particular features of the software. Instead a series of real-world scenarios are posited and one sees how they might be resolved with Version Control.

Subversion and Git have been chosen for this workshop since they represent opposite designs within the world of Version Control. Subversion follows a client-server architecture with rigidly defined workflows. Subversion's methodology is similar to CVS and Perforce. Git follows a distributed architecture with more ad-hoc workflows. Git's methodology is similar to BitKeeper and Mercurial. Gaining hands-on experience with both Subversion and Git will give participants a solid understanding of the options and trade-offs present in Version Control systems.

The workshop starts by examining single-person (non-collaborative) scenarios using Subversion:

- Single-edit workflow. Check out, edit a file, commit. This is the most common use case.
- Multiple-edit workflow. Managing a change list across several files including additions and deletions.
- Parallel edits. Handling two edits by the same user at the same time (such as a large edit that is interrupted by a high-priority small edit).

Next the workshop examines multiple-person (collaborative) scenarios where the participants share a single repository.

- Conflicting changes. Dealing with collisions and three-way merges.
- History. Identifying who was responsible for a particular change and how to dig through the repository history to answer questions.
- Rollback. Undoing another contributor's changes.

Each of these scenarios uses one or more plain-text documents, with each step building upon the previous step.

After learning how to use an existing repository, the workshop then examines how to setup a new repository (up to now the participants would have been using a private repository running on a local server).

The workshop then repeats all of the above scenarios with Git instead of Subversion.

Finally the workshop looks at a few of the graphical interfaces for Subversion and Git, as well as a brief overview of some of the other popular Version Control systems that are available.

Participants leave with a solid understanding of how to use both Subversion and Git in the real world.

2. TECHNICALITIES

Each participant is required to bring a Windows, Macintosh or Linux laptop upon which they have rights to install software. Chrome OS is not suitable. Participants will install clients for Subversion and Git, both of which are free and open source. Laptops must have wireless capabilities (either WiFi or cellular). Free WiFi will be available.

The workshop lasts for three hours, with a short break at the midpoint. Punctuality in starting is encouraged as catching up is problematic if one doesn't have the required software installed. However, joining at the midpoint when we switch from Subversion to Git is possible, should one already be familiar with Subversion.

3. PRESENTER

Neil Fraser is a software engineer at Google. In addition to using Version Control systems professionally, Mr Fraser is also a contributor to Google Docs, MobWrite, the Diff Match Patch code library, and the Differential Synchronization algorithm – all of which have elements of Version Control at their core.

4. PREPARATION

To speed up getting started with the workshop, it would be appreciated if participants could follow a few preparatory steps on their laptop prior to the workshop. This should take about ten minutes. Email fraser@google.com if there are any problems.

1. Check for Subversion. Open a command prompt and verify that you get a help screen when you type: `svn help`

   If not, then install it:

   - Linux: `apt-get install subversion`
   - Macintosh: http://www.open.collab.net/downloads/community/
   - Windows: http://sourceforge.net/projects/win32svn/

2. Check for Git. Open a command prompt and verify that you get a help screen when you type: `git help`

   If not, then install it:

   - Linux: `apt-get install git`
   - Macintosh: http://code.google.com/p/git-osx-installer/downloads/list?can=3
   - Windows: "Full installer for official Git 1.7.x"

     http://code.google.com/p/msysgit/downloads/list?can=3

3. Fill out the form at http://goo.gl/ikQQL

5. REFERENCES

[1] Collins-Sussman, B., Fitzpatrick, B. W., and Pilato, C. M. 2011. *Version Control with Subversion.* O'Reilly Media. http://svnbook.red-bean.com/

[2] Chacon, S. 2011. *Git Documentation.* http://git-scm.com/documentation