Seminar 9 – Software Configuration Management

The goal of this exercise is to give a hands-on experience of using some common SCM tools

(Subversion and Git) and to learn and read about practices related to SCM. Moreover, the purpose is

also that the student should train in reading manuals, tutorials etc. to solve common problems.

Recommended “Reading”

We recommend the following sources of information:

* Git- SVN Crash Course: http://git.or.cz/course/svn.html
* The Git Community Book: http://book.git-scm.com/
* YouTube video – TechTalk about Git: http://www.youtube.com/watch?v=8dhZ9BXQgc4
* Git help. Run “git” from the command line.
* GitHUB – public git repositories: http://github.com/
* The Git homepage: http://git-scm.com/
* Subversion resource page: http://subversion.tigris.org/
* A SVN Tutorial for Unix : http://artis.imag.fr/~Xavier.Decoret/resources/svn/index.html
* Free online book on subversion : http://svnbook.red-bean.com/
* Subversion help. Run “svn help” from the command line.
* Sourceforge. Lots of available opensource SVN repositories. http://sourceforge.net/

Task

In the section “Work Items” below, you will find a list of items that are common to do when working

with a SCM system. However, depending on if you are using Git or Subversion, the way to do it can

be quite different. For each of the items, your task is to do the following:

* Try to do the work item using both Git and Subversion, i.e., select some project/set of files, download an opensource project from the web etc., and do the tasks by running git and svn commands. You can either use the command line or available GUIs for the SCM system.
* Document the steps you did by copying the used commands or example screenshots. Give a good description, so that a reader of your document understands what you have been doing.
* Write a short reflection of the differences of the workflow with Git and Subversion for the particular work item.

Work Items

* You have a source code project (a set of files) on your local hard drive and want to create a new repository, so that you can track changes.
* You want to down load an existing open source project available on the web and then browse the history of changes in the project.
* Your are thinking of developing a complete new part in a project and therefore wants to create a development branch from the main trunk (called master in Git). You do some modifications in the new branch and then merge back the changes to the trunk.
* You are working on a project and discover a bug in one of the files in the latest version of the project. You want to figure out who added this bug, so that you can contact him/her.
* You are planning to make a new release of your software and want to mark in the SCM system the exact version for this release (let us call it version 1.2). You are then editing and committing some new changes to the repository. Suddenly, your boss says that he wants to have an email with a snapshot of the source code for version 1.2
* You have made a couple of changes in some files on your local workspace. Suddenly you discover that your changes were plain wrong and you want to go back to the version that currently exists in the repository.
* You have traced back that a bug was introduced at a specific revision in the repository. You now want to figure out the exact lines of code that introduced the bug.

Tips and Resources

* Git can be downloaded from: http://git-scm.com/
* gitk is a very nice tool to browse repositories. If you are running Mac OS X, GitX (http://gitx.frim.nl/) is also a good alternative to gitk.
* use command “module add prog/git” to enable git. If you run “module initadd prog/git”, git will be available the next time you log in. Subversion can be added in the same way; just replace word “git” with “subversion”.
* Subversion tools for most platforms can be downloaded here: http://subversion.tigris.org/

Pass condition

The following conditions must hold to pass this exercise:

* Each of the work items are tested and documented in a way so that a reader easily can follow each step of your description.
* Each work item is done for both Subversion and Git.
* A short reflection is written down for each item.

Reporting

Hand the report as a PDF-file in the lecturer.