Software Engineering and Project Management

ST2SEPM – 2016-17

Software configuration – Version control by example
Subversion (svn)

I – Install Subversion (local) : *See installation tutorials on Moodle*

II – Reproduce tutorialpoints.com svn quick guide examples
<u>*(Skip the installation and networking part)*</u>
Link and PDF document available on Moodle

III – Practice
1 – New repository, adding, moving, deleting, resolving conflicts, tagging

1. Create a new directory project named *sepm*. In that project directory write a new *readme.txt* file containing some introduction information about your project, its name, developer names, date, company name *etc*. The file should start with the line "sepm – my new development endeavour". *This is the initial project directory.*
2. Create a new local svn repository named *sepm-svn*.
3. Import your *sepm* project in the *sepm-svn* repository (trunk). *Notice the difference between the initial project directory and the svn repository. Understand the role of the three folders (trunk, branches and tags).*
4. Checkout a working copy of the newly created project in a new directory named *sepm-wd*. *Notice the difference between initial project directory, svn repository and the working directory.*
5. Checkout a second working copy of the project in a different working directory, called *sepm-test*. *Now you can work in two distincts working directories, simulating a two developers team using the same repository.*
6. In the *sepm-test* working directory, modify the *readme.txt* file so that the first line looks like "**S**epm – my new development *endeavour***!**" (differences are underlined).
7. Check the status of your working directory. What do you get?
8. Commit the change. *Don't forget to write an informative message!*
9. In the first working directory (*sepm-wd*), change the first line to "sepm – my new **programming** *endeavour*". Try to commit the change. Resolve the conflict.
10. In your current working directory (*sepm-wd*), create three new folders named *src, bin* and *doc*. Move the *readme.txt* file in the *doc* directory and create a new *main.cpp* file inside *src* (the main function should print a silly message in the console).
11. Check the svn status. What happens?
12. Do what's necessary to add/remove the new directories and files and commit successfully.

13. Release version 1.0 of your project

2 – Tagging, branching, merging …

We would like to experiment a new way of printing a silly message to the user by printing a second silly message!

1. Create an experimental branch named *expe*
2. <u>On the main branch</u>, update the *main.cpp* file to change the content of the first displayed silly message. Commit.
3. <u>On the experimental branch</u>, update the *main.cpp* by adding a line of code to print the second silly message.
4. Update the repository.
5. You have heard that some important modifications have been made to the main file in the trunk branch that you should integrate in your experimental branch. Do what's necessary to merge the main branch into the experimental branch