GIT/Gerrit assignments

Installation package & document of git:

[\\Fsoft-filesrv03\FSU1\BU16\Project\Freescale\FSL\_KinetisSDK\FSL\_KinetisSDK\_S2\_2014\Reference\GIT](file:///\\Fsoft-filesrv03\FSU1\BU16\Project\Freescale\FSL_KinetisSDK\FSL_KinetisSDK_S2_2014\Reference\GIT)

[\\10.207.215.81\share\doc\GIT\](file:///\\10.207.215.81\share\doc\GIT\)

[\\10.207.215.81\share\installer\GIT\](file:///\\10.207.215.81\share\installer\GIT\)

**PLEASE STUDY GIT BEFORE DO THIS ASSIGMENT!!!**

For more document/tutorial please google first.

# Assignment 1 – git

One hour to complete this assignment, using git-bash with command-line to do this assignment. **Please store all commands used for each step**.

1. Create new local repository/local bare repository named **R0**
2. Clone new local repository from local repository/local bare repository created from #1, new repository named **Ra**
3. In Ra, create/edit source file(s). Add all file and commit these changes (commit **Ca0**) on branch **master**
4. Create/edit multi source files. Add some of changed files (not all files), then commit added files (commit **Ca1**).
5. Commit other changed files (which did not add in #4) as a commit (**Ca2**).
6. Modify more than 2 source files, then commit 1 file and reset other files (unchanged) as commit **Ca3**.
7. Push branch master to remote repository (**R0**).
8. Edit one file (one of committed file in #6), then add this change to commit **Ca3** (2 ways, commit --amend and soft reset then commit, answer how different between these ways?).
9. Push master branch to remote repository (**R0**) again.
10. Clone another local repository from local repository/local bare repository created from #1, new repository named **Rb**.
11. On **Rb**, create new branch named **feature1** from commit **Ca2** of **master** branch.
12. Edit 2 files then commit to **feature1** branch (commit **Cb1**).
13. Edit 1 file which is changed in commit **Ca3** (#6 or #8), edit the same location (in file) with change from **Ca3**. After that, commit this commit as **Cb2**.
14. Merge branch **feature1** (**Cb2**) with **master** branch (**Ca3**). Resolve conflicts then push back to **master** branch of remote repository **R0**.
15. Create another remote repository named **R1** by copy **R0**.
16. On **master** branch of **Ra**, edit some files then commit (**Ca4**).
17. Add new remote named **rmt-r1** has URL points to **R1**.
18. Push new changes of **master** branch (of **Ra**) to **R1**.
19. On **Ra**, create new branch named **test\_rebase** based on commit **Ca2**.
20. On **test\_rebase** branch, edit one file which is changed in commit **Ca3** or **Ca4** (#8 or #16), edit the same location (in file) with change from **Ca3** or **Ca16**. Then commit these changes to **Cat1**.
21. Rebase the **Cat1** to the latest of **Ra/master** with conflicts resolved.

**PLEASE STUDY GERRIT BEFORE DO THIS ASSIGNMENT!!!**

# Assignment 2 – gerrit

Mọi người luân phiên thử nghiệm các vị trí khác nhau (committer/developer – <D1, D2, D3>, reviewer<R1, R2>, approver<A1>) trên repository test.git (<http://10.207.215.71:81/#/admin/projects/test>). Ví dụ trong 1 trường hợp LinhNV15 làm D1, PhucNH6 làm D2, BinhPV làm R1, HoaND làm A1; sau đó luân phiên, PhucNH6 làm D1, BinhPV làm D2, HoaND làm R1, LinhNV15 làm A1; ...

Kết thúc Assignment 2 mọi người đều phải trải qua 3 vị trí (D, R, A) như trên.

Việc config cho gerrit mọi người đọc slide **Review\_Using\_Git\_Gerrit\_v0.4.pptx**

1. **A1** tạo branch mới trên gerrit (baseline từ latest của master), tên đặt khác nhau, ví dụ **Branch1**.
2. **D1** và **D2** cùng clone/checkout branch **Branch1** này.
3. **D1** tạo mới/chỉnh sửa files, sau đó commit (**B1C1**).
4. Push **B1C1** lên gerrit để review, add R1 và R2 vào reviewer.
5. **R1** và **R2** vào review, comment trên các chỉnh sửa này, và cho điểm -1 (không đồng ý, phải sửa lại) publish comments.
6. **D1** chỉnh sửa các files được comment, sau đó push lên thành patch set mới **B1C1-P2**.
7. **R1** và **R2** vào review, lần này sẽ accept, cho điểm +1 (chấp nhận commit) và publish comments cho **B1C1-P2**.
8. **A1** vào và approve cho commit **B1C1-P2**.
9. **D2** pull latest commit trên **Branch1** (**B1C1-P2**).