

Lab00: Git & Github Guidelines

Lecturer: NGUYEN Thi Thu Trang, trangntt@soict.hust.edu.vn

1. GIT & GITHUB

1.1. INTRODUCTION TO GITHUB

<https://lab.github.com/githubtraining/introduction-to-github>

1.2. GITHUB AND EGIT IN ECLIPSE

Git tutorial 1: creating github repo and sharing project in Eclipse

<https://www.youtube.com/watch?v=cdsMIX9gB94>

Git tutorial 2: Committing, pushing, pulling and resolving conflicts with git and Eclipse

<https://www.youtube.com/watch?v=M88sKbRDR8Y>

2. GIT WORKFLOW

2.1. GitHub flow

<https://guides.github.com/introduction/flow/>

2.2. Release flow

<https://devblogs.microsoft.com/devops/release-flow-how-we-do-branching-on-the-vsts-team/>

- We can create as many branches as we need.
- We name branches with meaningful names. See Table 1-Branching policy.
- We had better keep branches as close to master as possible (i.e., avoid subbranch); otherwise, we could face merge hell.
- Generally, when we merge a branch with its origin, that branch has been history. We usually do not touch it a second time.
- We use release branch as our assignment submission.

Name of Branch	Naming convention	Origin	Merge to	Purpose
feature or topic	+ feature/feature-name + feature/feature-area/feature-name + topic/description	master	master	Add a new feature or a topic
bugfix	bugfix/description	master	master	Fix a bug
		feature	feature	
hotfix	hotfix/description	master	master & releases [1]	Hotfix a bug
refactor	refactor/description	master	master	Refactor
		feature	feature	
release	release/versionXX	master	none	Assignment submission