Basic Tools for System Management

Frederic.Mallet@unice.fr

Université Nice Sophia Antipolis
Membre de Université Cote d'Azur

Objectives

- **Learn how to manage your own computer**
  - Using the shell
    - Linux / Windows
  - Version control and git
  - Installing a Virtual Machine
    - Linux Mint 18.4
  - Using bash
  - Eclipse: JDT and Modeling

Pull Request

■ Feature of Git hosting sites (e.g bitbucket)

■ Goal: merge a branch into the project

- Notification to team members

- Feedback or comments

- Approval of the content: code review

■ Two configurations

- Single remote repository

- Multiple remote repositories:

  • merge a fork repository into the upstream, when submitter does not have access to the upstream repository

When to open a pull request ?

- When the branch is created

- When you are stuck and want comments

- When the branch is ready for review or merging

Single repository Pull Request

- **Preparation**
  - Create a feature branch
  - Work on it
  - Push the branch to the remote repository
    - --set-upstream: setup a local tracking branch
- **Create a pull request**
  - From the website: explanation, pick reviewers
  - When reviewing: might decline, edit or approve
- **Initiate merging (online or with local client)**
  - git merge --no-ff
  - git merge –squash
- **Remove remote branch labels**
  - git push -d <remote> <branch>

Forking

■ Forking
- Copy a remote repository to your own online account (both remote)
- Experiment or learn
- Also a way to create branches
- Different source of truth independent from the upstream repository

■ Creating a fork
- Service of the git hosting (bitbucket or github)

■ Synchronizing a fork
- Sync now = merge commit on the forked repository
  - This merge commit is not available on the upstream repository
  - Need to pull to update the local repository

Multi-repository Pull Request

■ Fork the upstream repository

■ Create a branch

■ Open the pull request from the fork

- This will open a pull request on the upstream repository

■ If the owner of the upstream repository decides to merge

- Either from the website or

- Locally

  • Add the forked repository as a remote
  • Perform and push the merge

Git Workflow

- **Centralized workflow**
  - Only a single branch
  - No pull requests/discussion

- **Feature Branch workflow**
  - Work done on feature/topic branches
  - Single remote repository
  - Submit using pull requests

- **Forking workflows**
  - Involves multiple remote repositories, does not need write access to upstream
  - Pull requests from forked remote to upstream remote
  - Can rebase your forked branch
  - May be out of sync

- **Gitflow workflows**

Gitflow workflows

- Designed by Vincent Driessen

- Git-flow is a command line tool that can simplify the application of the flow

https://www.atlassian.com/git/tutorials
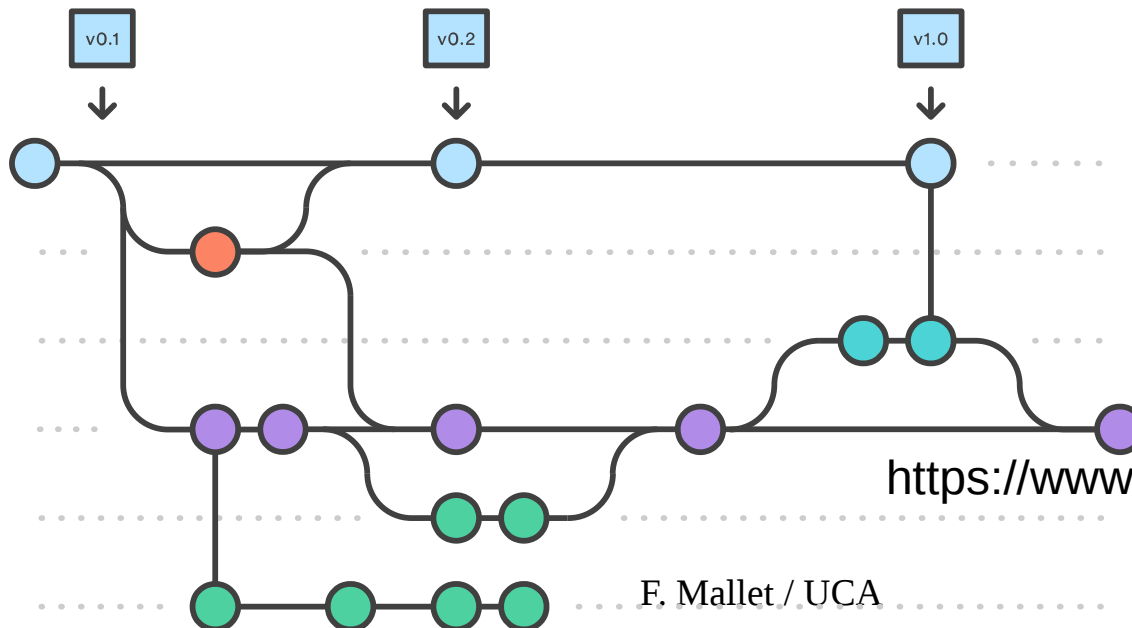
Some "merging" rules

- Only merge commit on the master branch
  - Except for the initial commit

- Only merge from the release branches or hotfix branches
  - Never merge from the develop branch
  - Allows fixing bugs in the release without disrupting the develop or features
  - Ease the testing process

- Always merge release and hotfix commits into the develop branch
  - Otherwise bugs may reappear later
  - May need rebase on feature branches after merging hotfix and release