Software Studio

軟體設計與實驗

Git Advanced

Hung-Kuo Chu

Department of Computer Science

National Tsing Hua University

CS2410

What are we going to learn?

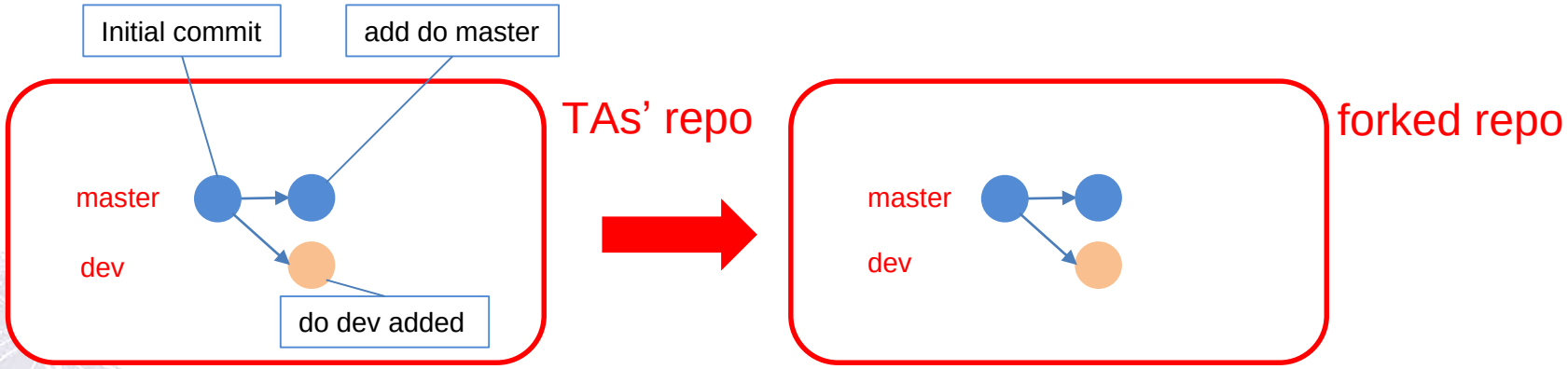
We are going to learn to…

- Fork / Pull request (merge request)
- Clone a specified branch from remote repo
- Solve merge conflicts
- Publish branches to remote repo

What are we going to do?

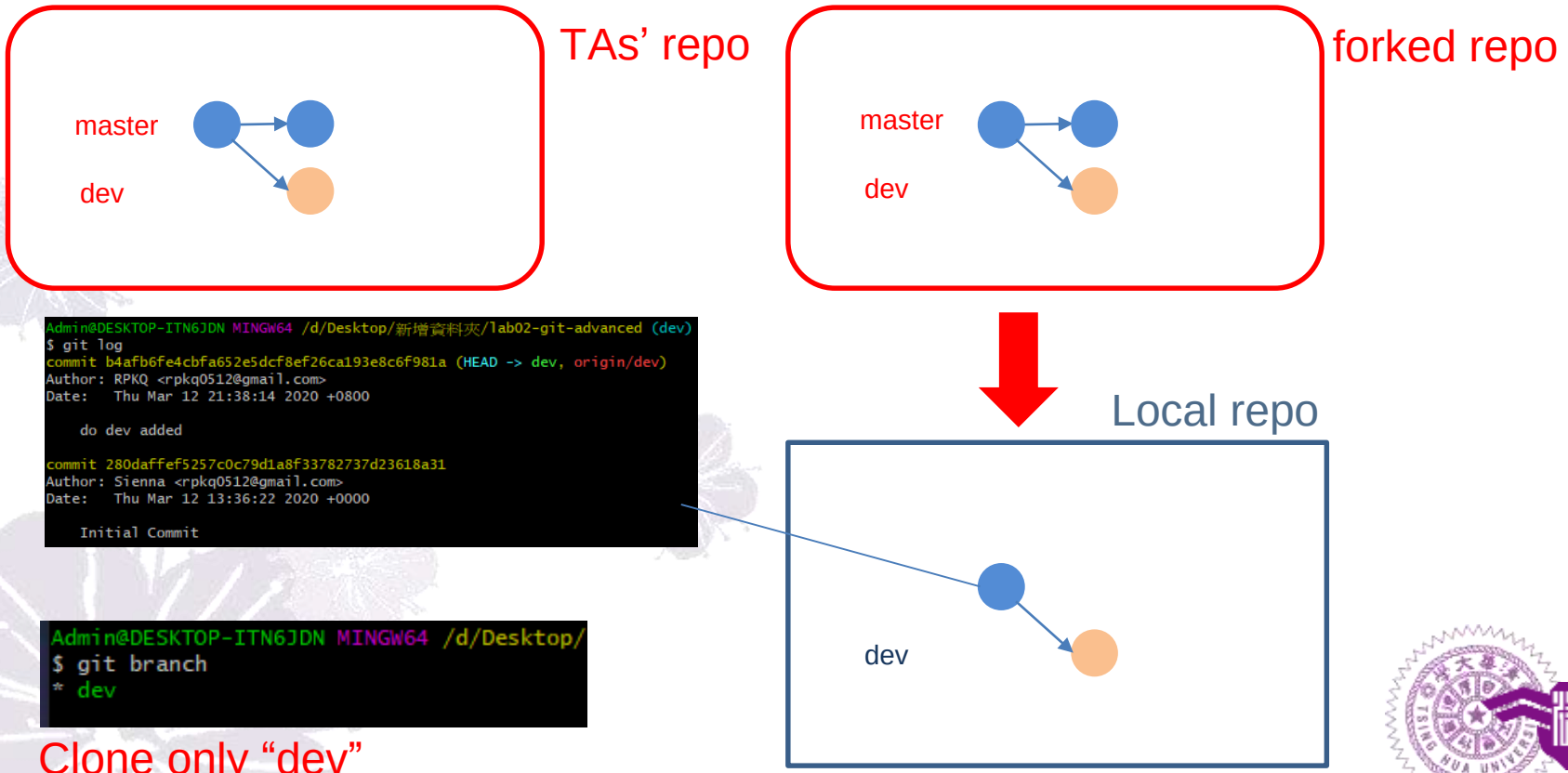
Todo

- fork repo



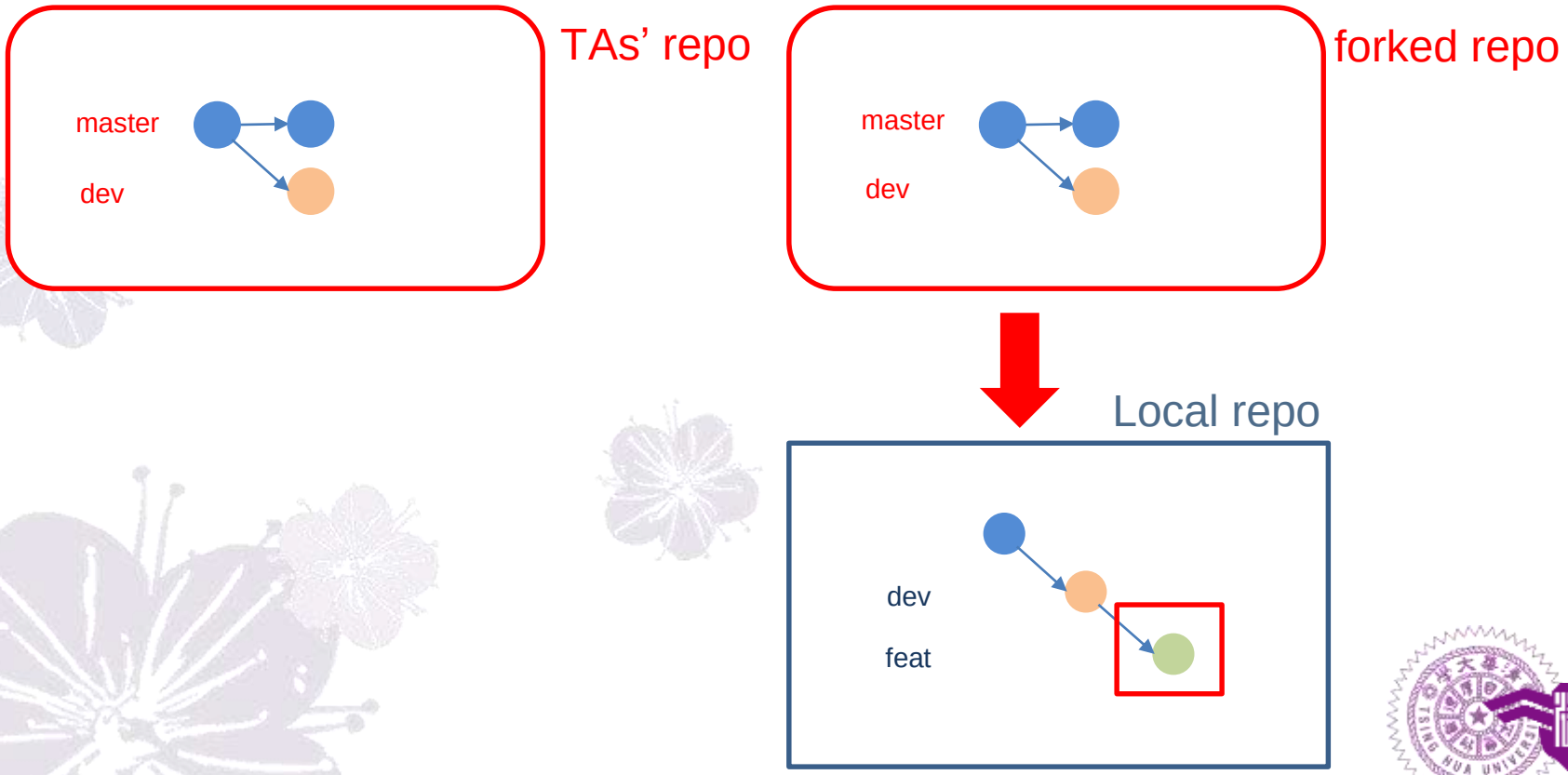
Todo

- Clone “dev” from forked repo



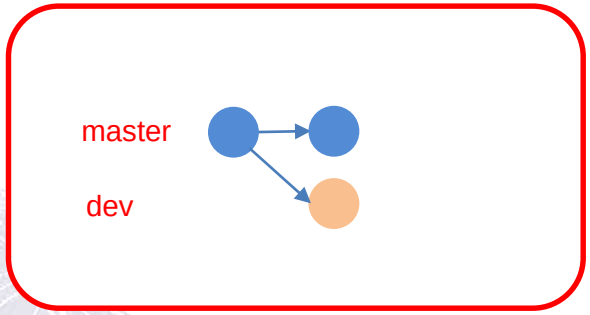
Todo

- Create a new branch “feat” and modify it

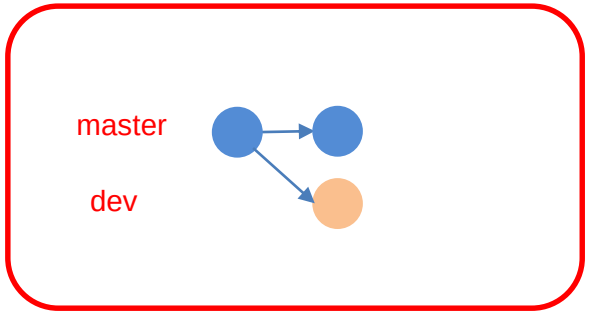


Todo

- Modify “dev”

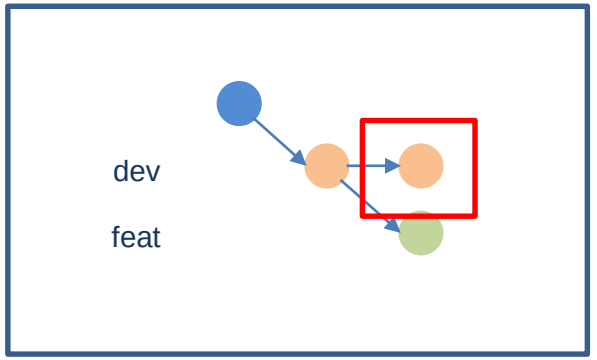


TAs’ repo



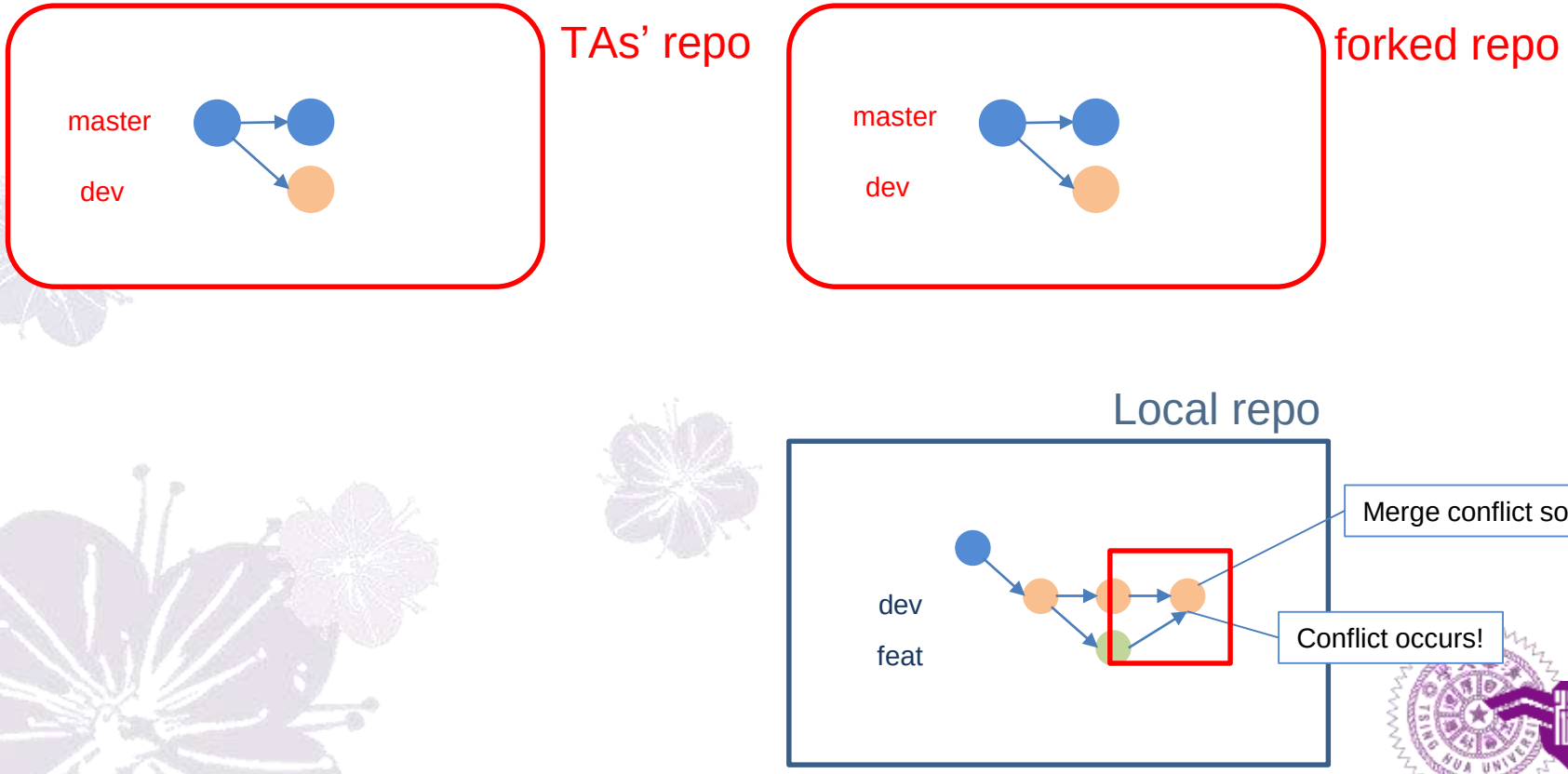
forked repo

Local repo



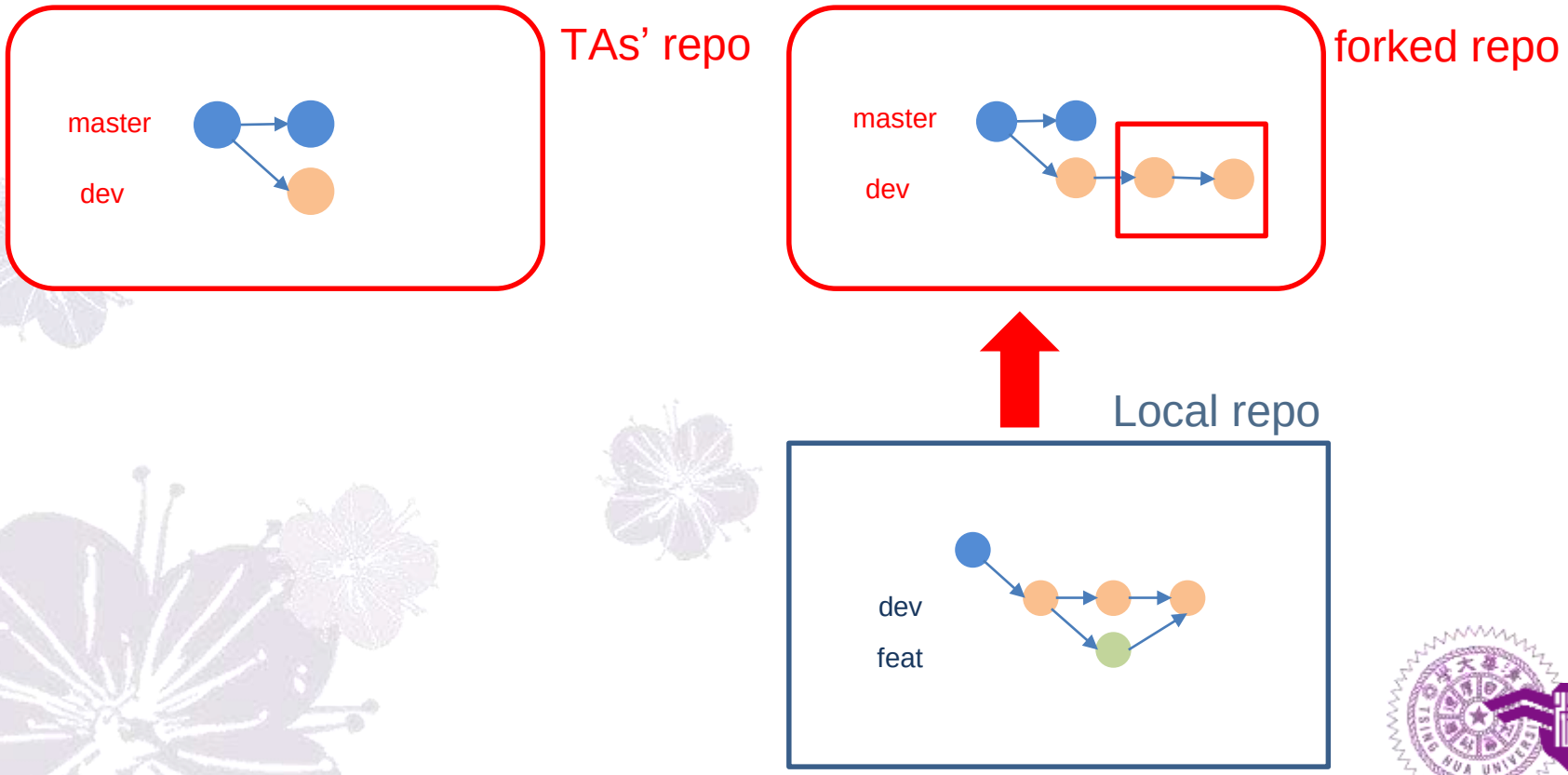
Todo

- Merge “feat” to “dev” and solve the conflict



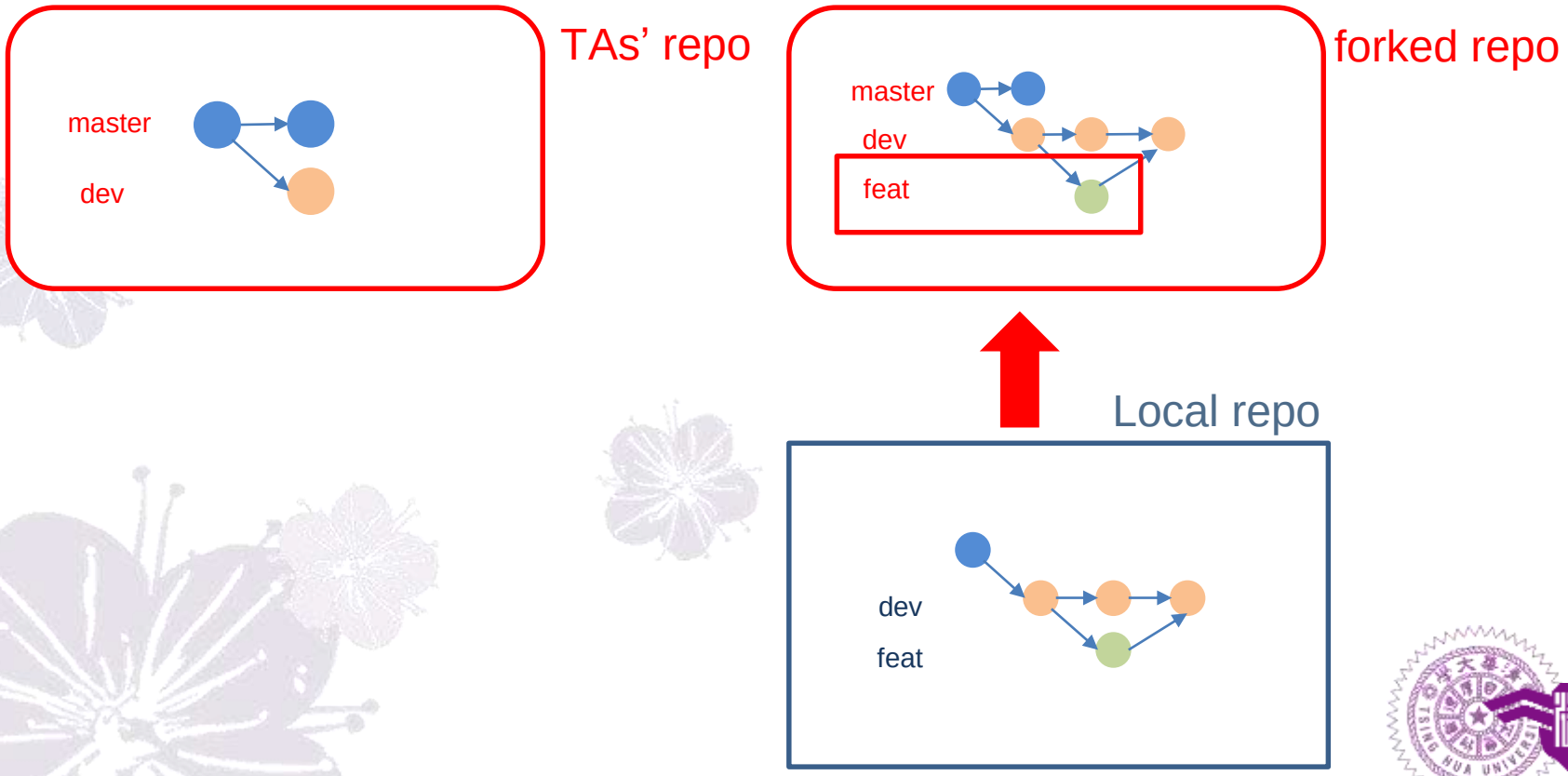
Todo

- Push changes from local “dev” to remote “dev”.



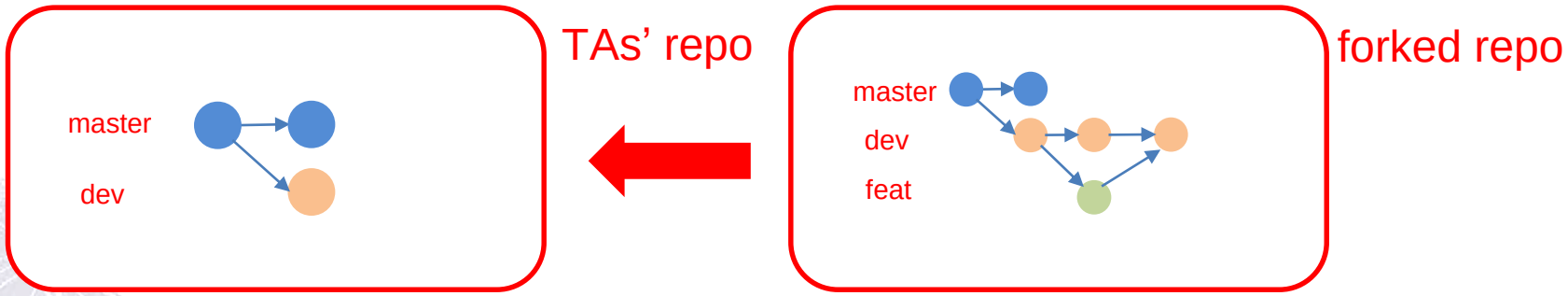
Todo

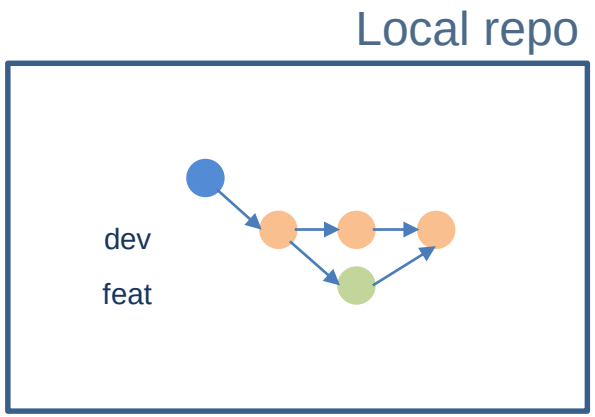
- Publish local “feat” to remote “feat”



Todo

- Make merge request from “dev” to “dev”.





Todo Layout

Todo Layout

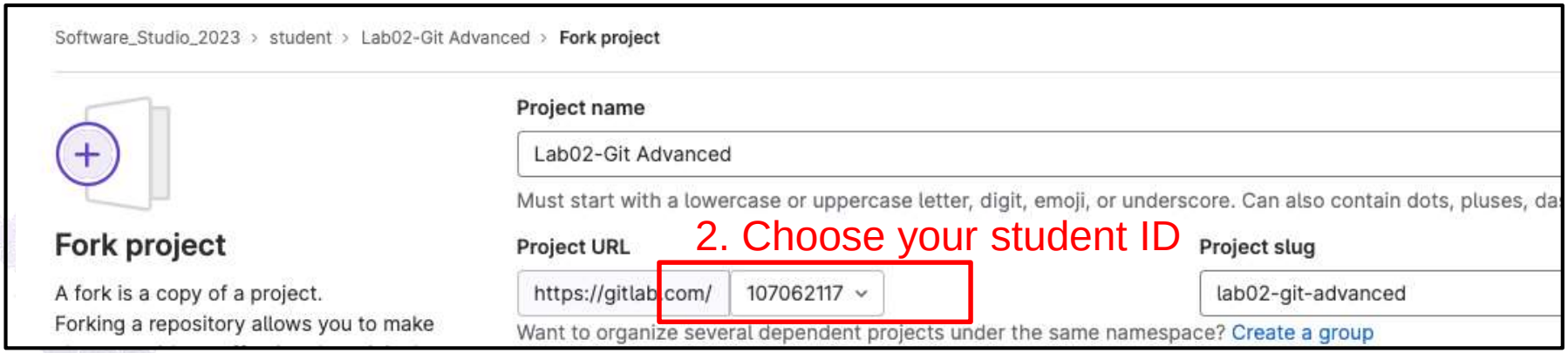
- **Fork** the project to your account.
- **Clone** branch “dev”.
- Create a **new branch “feat”**
  - add “do feat” to the list
  - Add and commit the change.
- Switch back to “dev”
  - add “do dev2” to the list
  - Add and commit the change.
- **Merge** “feat” to “dev”
  - **Solve the conflict** by keeping “do feat”
  - Add and commit the change.

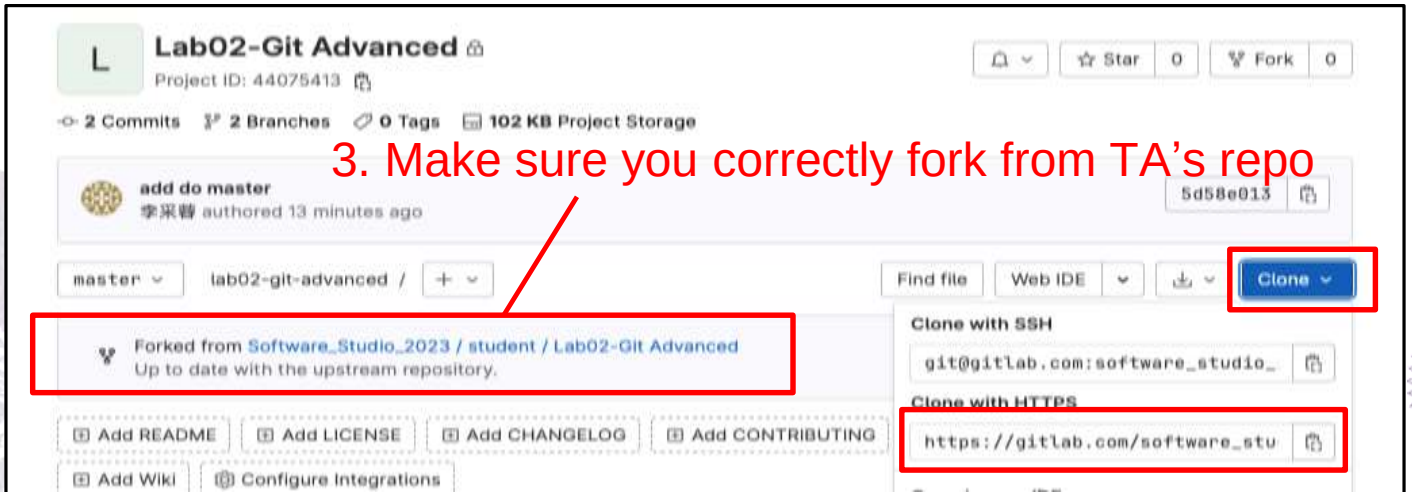
Todo Layout

- **Push** “dev” to remote repo.
- **Publish** “feat” to remote repo.
- Make **merge request** from “dev” to “dev”.

Step by Step

Todo – Fork Repo





Todo – Clone “dev”

1. Open a new folder with VSCode.
2. Ctrl + ` to open the terminal.
3. Clone “dev” from your repo. Do not clone TAs’ repo!

```
PS D:\Desktop\New folder\lab02-git-advanced>
```

remote branch name

- **`git clone --single-branch --branch dev <URL>`**
  (If you simply use “git clone” then you will clone “master” instead of “dev”)
- **`git branch`**

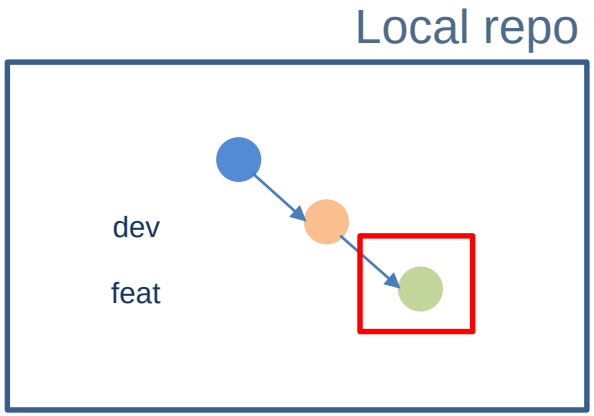
```
(base) licairongdeMacBook-Pro:lab02-git-advanced licairong$ git branch
* dev
```

Todo – Branch “feat”

1. Create branch “feat” and switch to it.
2. Add “do feat” to the list and commit the changes.

```
PS D:\Desktop\New folder\lab02-git-advanced>
```

- **`git checkout –b feat`**
- Modify the list content: add “**do feat**”
- **`git add *`**
- **`git commit –m “add do feat”`**

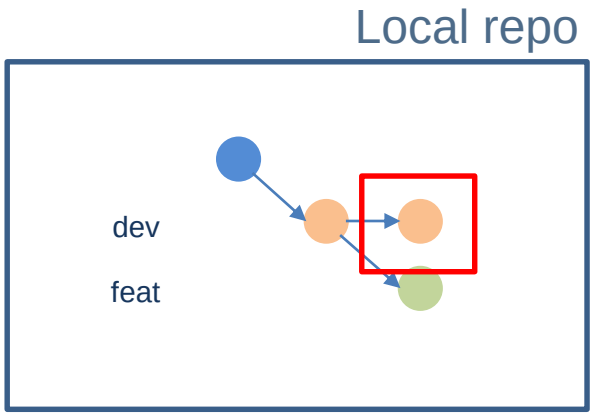


Todo – Branch “dev”

1. Switch back to “dev”.
2. Add “do dev2” to the list and commit the changes.

```
PS D:\Desktop\New folder\lab02-git-advanced>
```

- **`git checkout dev`**
- Modify the list content: add “do dev2”
- **`git add *`**
- **`git commit –m “add do dev2”`**

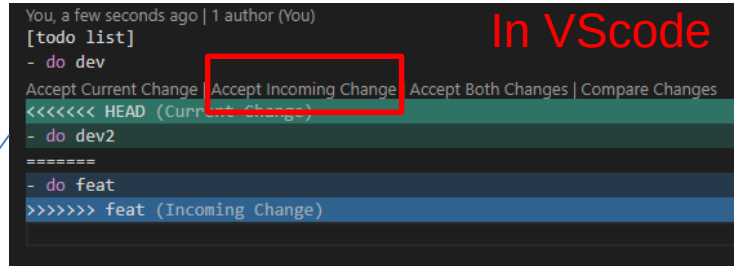


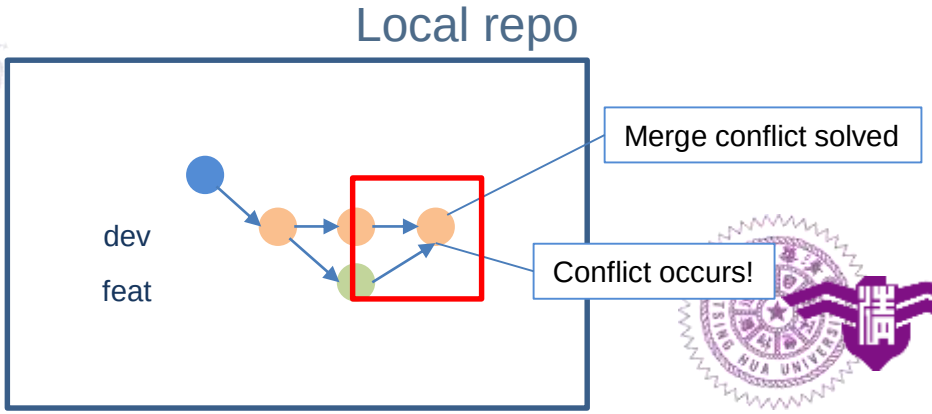
Todo – Merge and Conflict

1. Merge feat to dev.
2. Solve the conflict by keeping “do feat”.
3. Add and commit the change.

```
PS D:\Desktop\New folder\lab02-git-advanced>
```

- `git merge feat`
- solve the conflict: keep “do feat”
- `git add *`
- `git commit –m “merge conflict solved”`





Todo – Push and Publish Branch

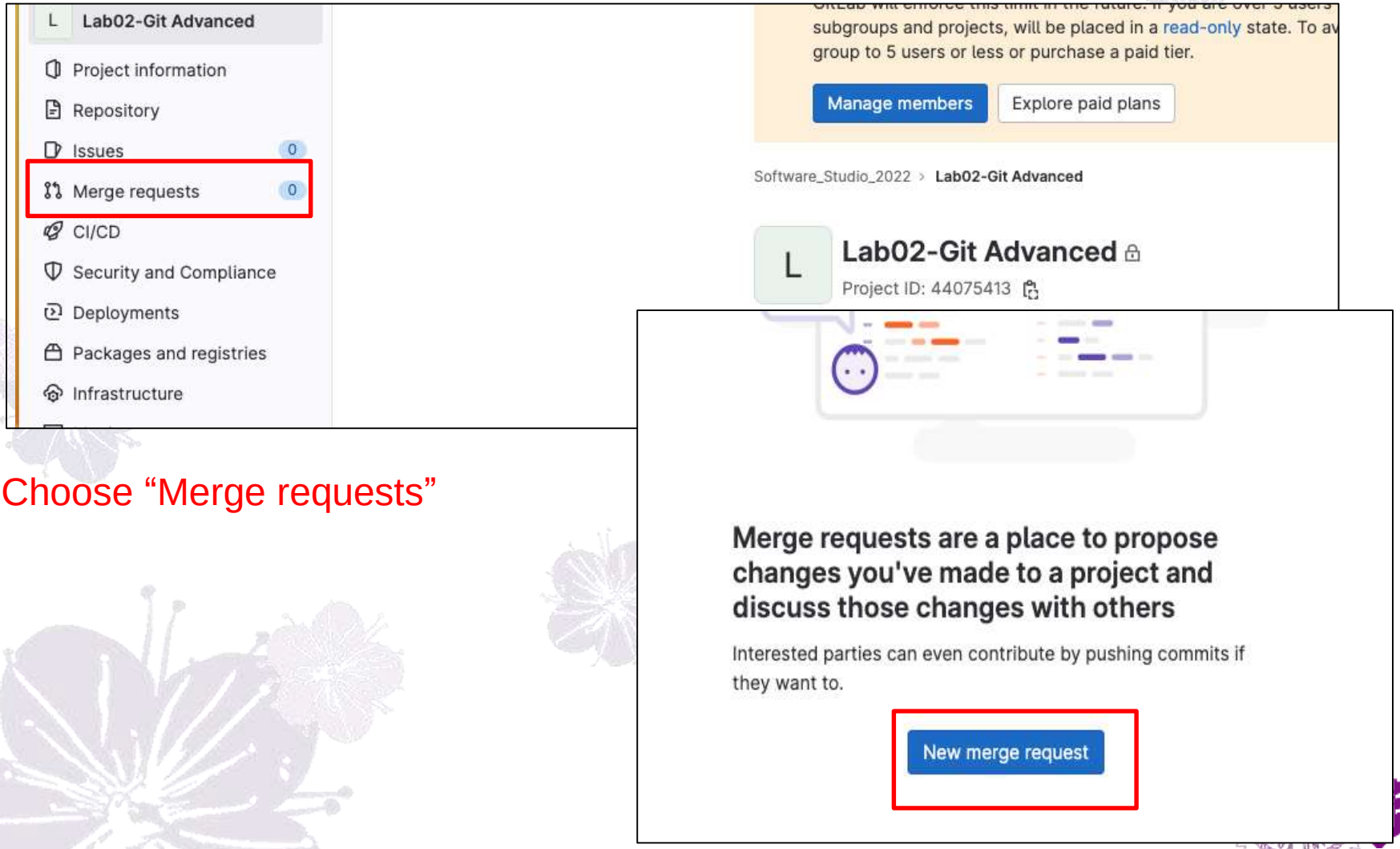
- **“dev”** : Push to update the remote branch “dev”.
- **“feat”** : There is no “feat” branch in remote repo. Use “-u” to publish a local branch to remote project.

```
PS D:\Desktop\New folder\lab02-git-advanced>
```

- `git push`
- `git checkout feat`
- `git push -u <URL> feat`

You should be able to see the remote “feat” you published.

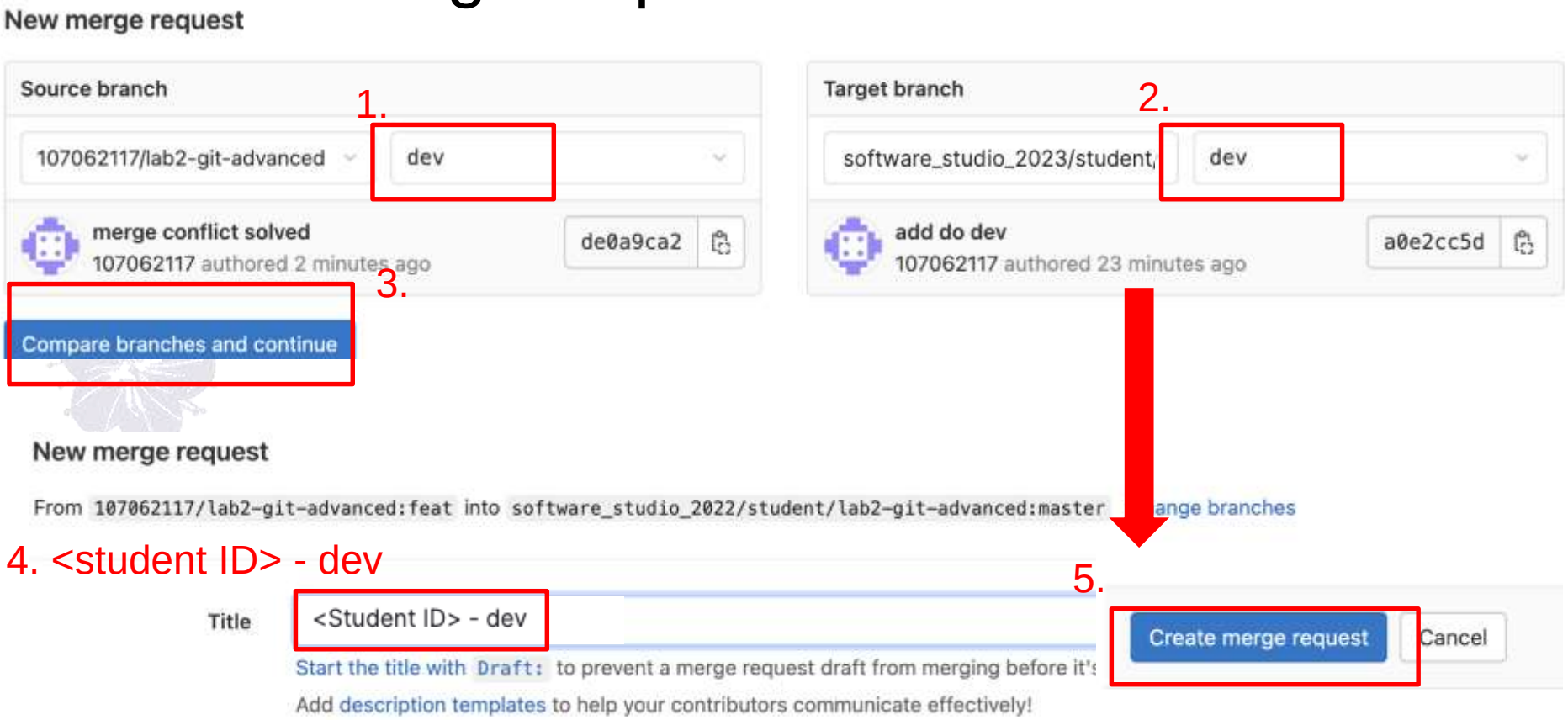
Todo – Pull(Merge) Request



Choose “Merge requests”

Todo – Pull(Merge) Request

- Make merge requests from “dev” to “dev”

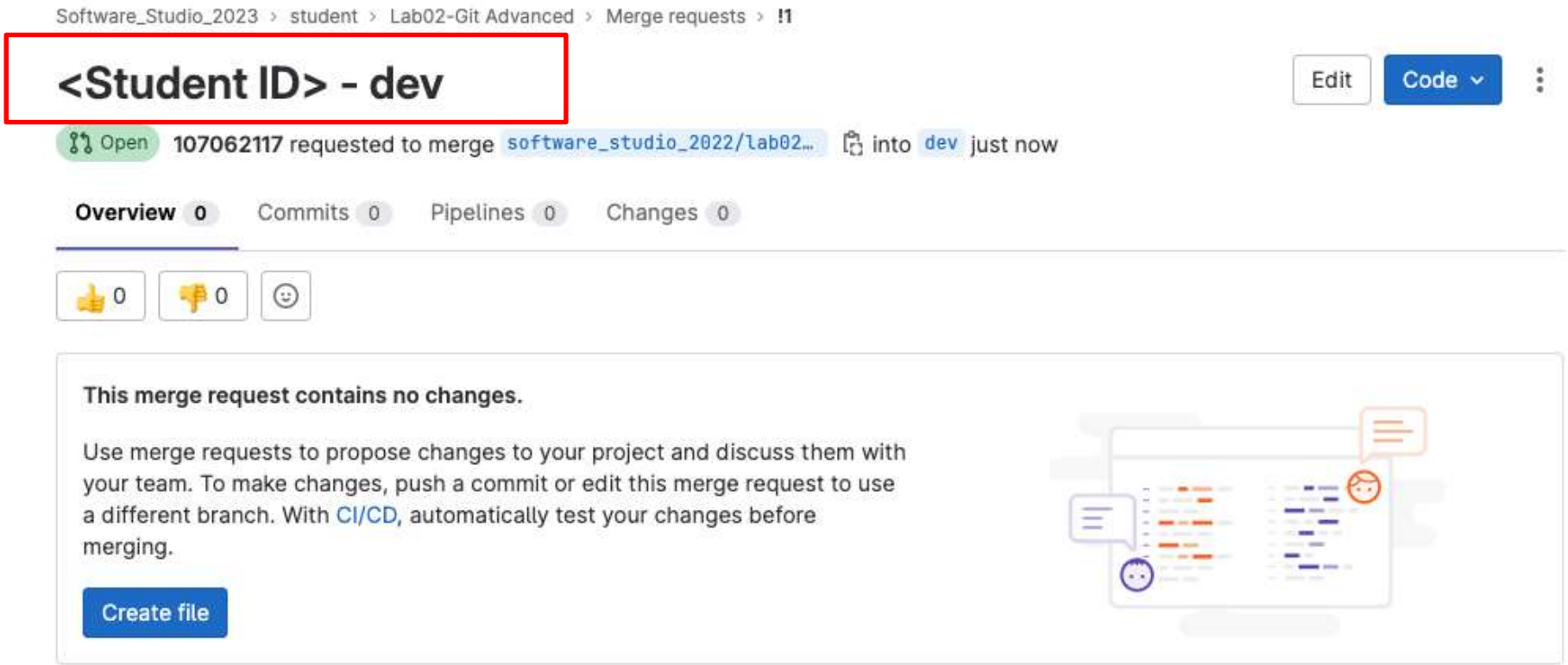


Grading Policy

- Branch “dev”
  - Merge Request “dev” to “dev” 10%
  - log message 20%
  - list content correct 20%
- Branch “feat”
  - “feat” branch exists in remote repo 30%
  - log message 10%
  - list content correct 10%

Grading Policy

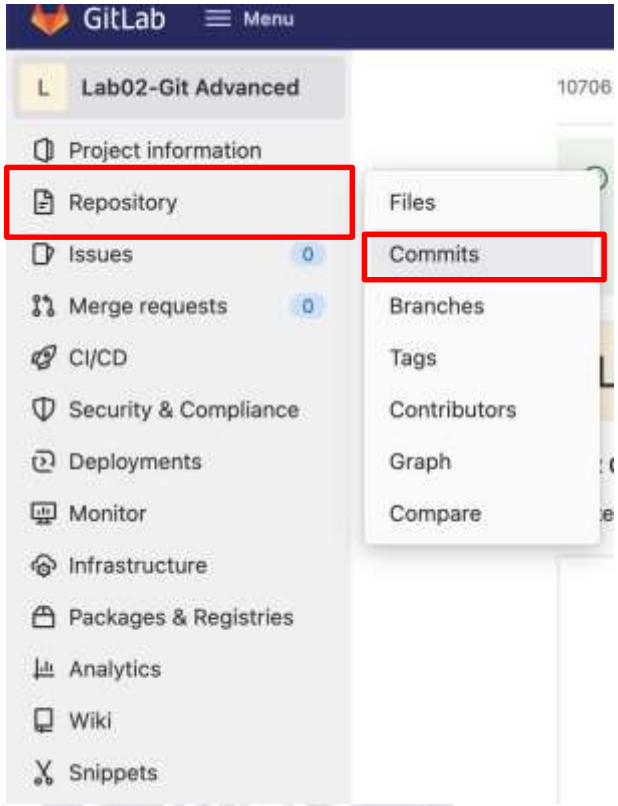
1. Merge request



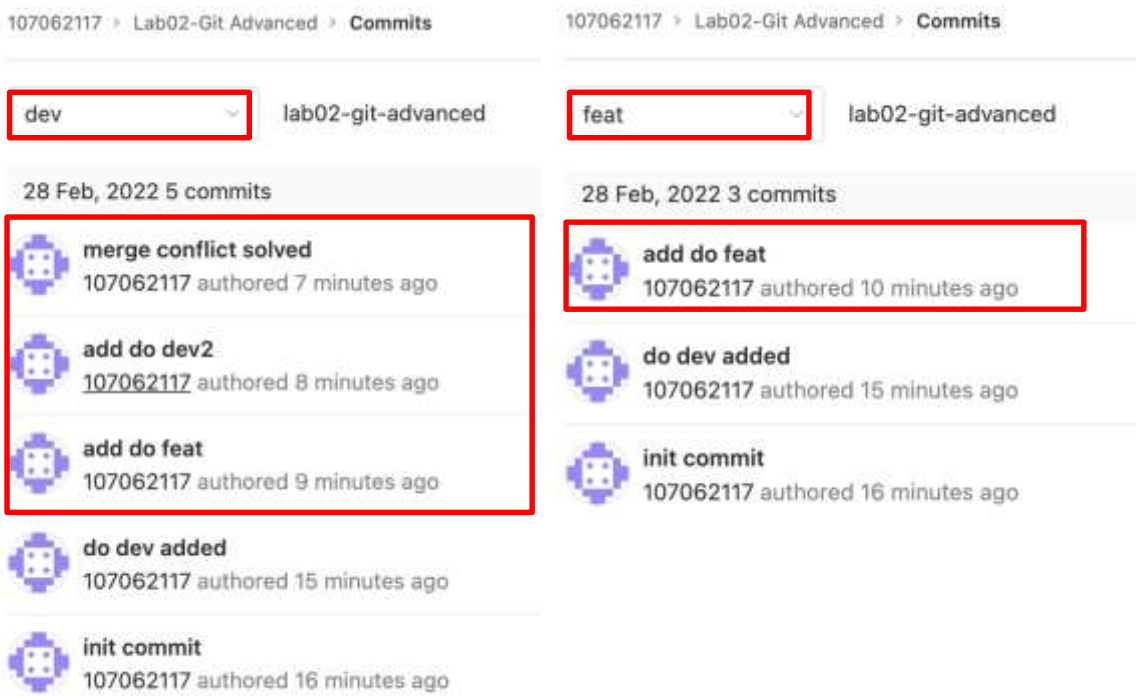
Grading Policy

2. Commit logs of dev and feat

1. Repository -> Commits



2. Show dev and feat



Grading Policy

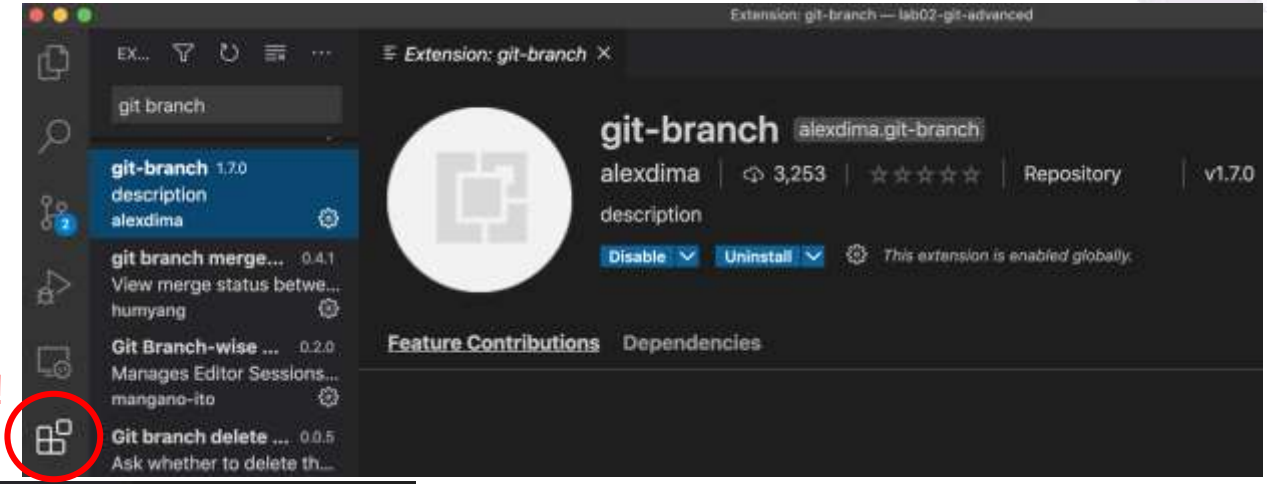
3. Content of the following 3 commits

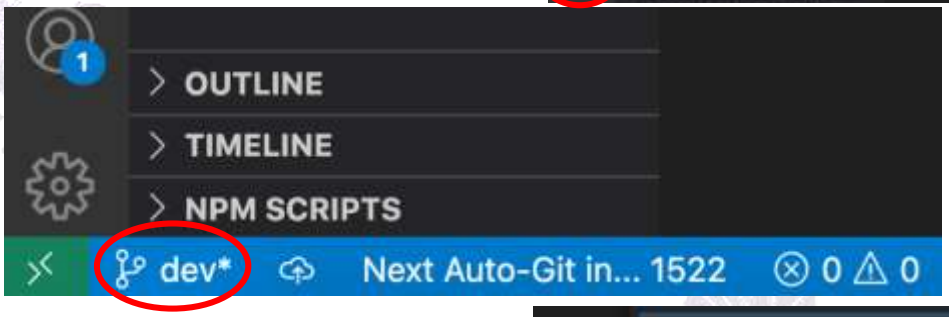
Notice

- If you damage TA’s repo, you will be **-20** points.
- Please **fork** the repo before syncing any changes.

Appenidx – git plug-in (branch)

You can install git plug-in here!





thank you!
Question