cs304
Software Engineering

TAN, Shin Hwei

陈馨慧

Southern University of Science and Technology

Slides adapted from cs427 (UIUC) and cs409 (SUSTech)

Offline+online presentation

Goto:
https://docs.qq.com/sheet/DWHNWa2NLZWxPQW5j?u=eeb2ed95d8204647af74f8f74810c55e&tab=vfg7ya&_t=1650593292008

- To pick your slot and make sure to attend the full lab section (as you need to wait for others to finish their presentation).

- To get the Tencent meeting link for those who cannot attend the offline meeting as you are not yet back on campus due to COVID. All students who are back on campus should attend the face-to-face meeting

Progress Report Presentation

Checklist for TA/Student

What to submit?

What to submit? **Create a tag called "version-1.0" and put all the following in the tag (Only things in the tag will be considered).**

-10 point if no tag

All source code and code comments

o Should write Javadoc for each added/modified public method

o For each of the modified/added block of code, add one line of comments before the code block to mark the corresponding issue. The format of the comments should be as below:

//CS304 Issue link: https://github.com/EvoSuite/evosuite/issues/191

All tests

o For each of the added test, add one line of comments before the test to mark the corresponding issue. The format of the comments should be as below:

//CS304 (manually written) Issue link:
https://github.com/EvoSuite/evosuite/issues/191

README.md. **All answers for the question (except for Javadoc comments and JUnit test) and link for pull requests (PR). Include the group name and name and id of each team member.**

.gitignore

Create tag

- Create a tag called "version-1.0"
  - The tag will mark a version for submission (no change is allowed after making the tag)
    - If you need to change the submission, then you need to delete and recreate the tag
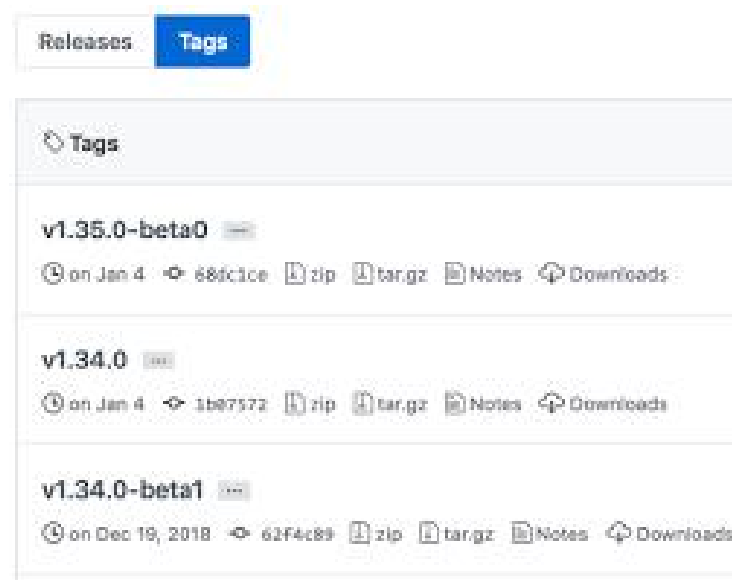
Code Review

Steps in Code Review

1. The **leader** will first introduce their selected projects. Then, he will summarize the progress of the team in a few sentences and introduce the role of each team member.

2. The **designer** will explain the design of their implementation and describe the design plan for the next iteration. The designer will also show a demo of the implemented issues.

3. The TA will pick one person randomly to run the static analysis tool.

4. The person (**developer**) in charge of each issue will describe how it works.

5. The person in charge of **documentation** will explain the part that he has implemented and the Javadoc comment for each method.

6. The **testers** will run all tests and shows the tests results and code coverage results.

7. The **team leader** will end the code review by showing and explaining their plan for next iteration. The TA will give some suggestions for future improvement.
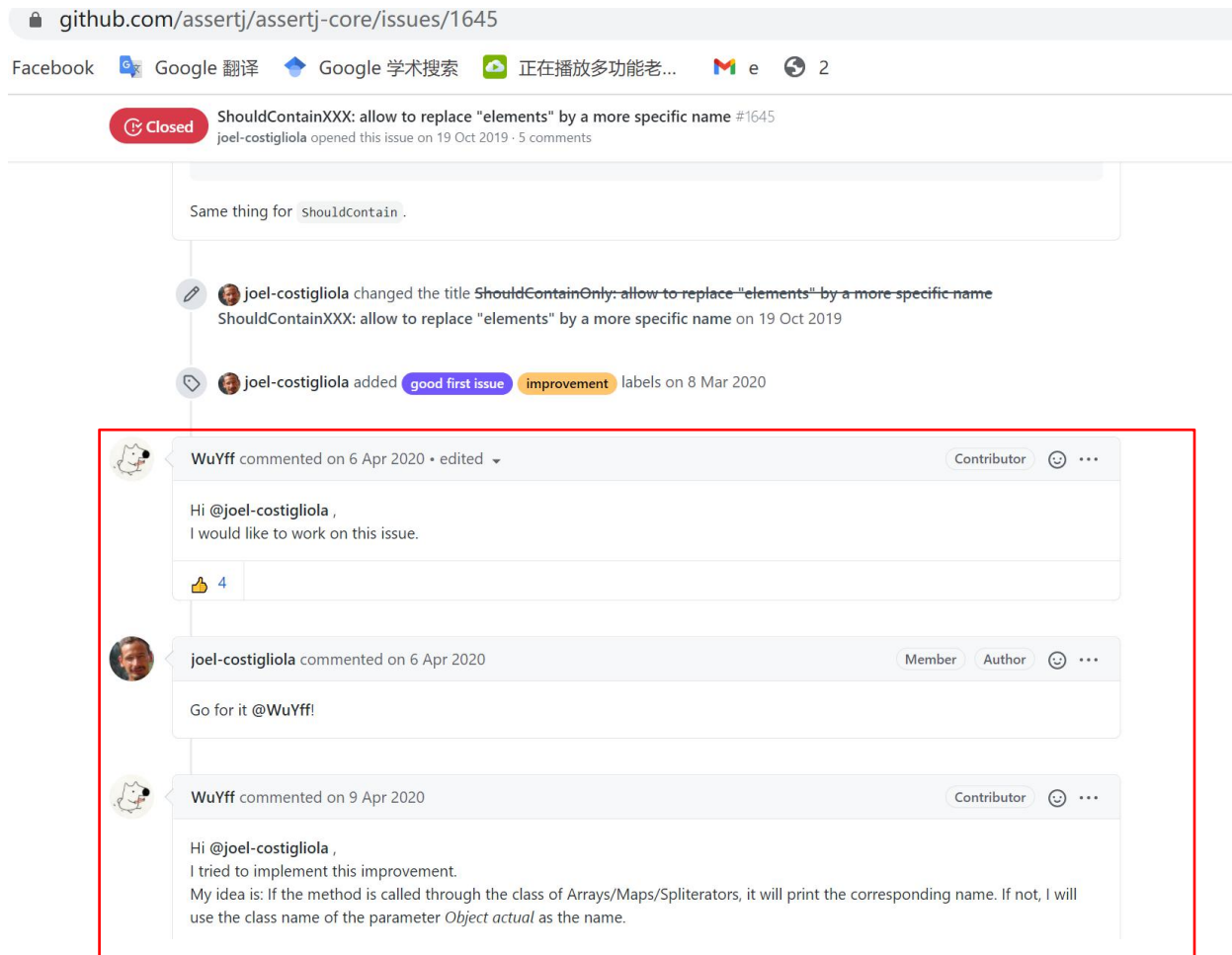
Project

Rules for submitting pull requests

c) Implement all the user stories of the issues that you have chosen. Include the links to each pull request implemented. **For each issue, there should be a unique pull request. Read the "PR-Rules.pdf" under the Project Resources folder in Sakai for rules on submitting a pull request. We will deduct 1 point for each violation of the rule.** *(Maximum Total: 12 points, 4 points for each issue implemented)*

Communication with developers

- **Get approval from developer about your intention to contribute**

https://github.com/assertj/assertj-core/issues/1645

- Leave a comment saying "I would like to work on this issue"
- Tell developer your idea of the implementation to check if it matches the requirement

What not to do

Rules for submitting pull requests

Project Rules:
What not to do when making pull request (PR)?

➢Need to change the code to resolve conflicts so that developers can merge cleanly

Project Rules:
What not to do when making pull request (PR)?
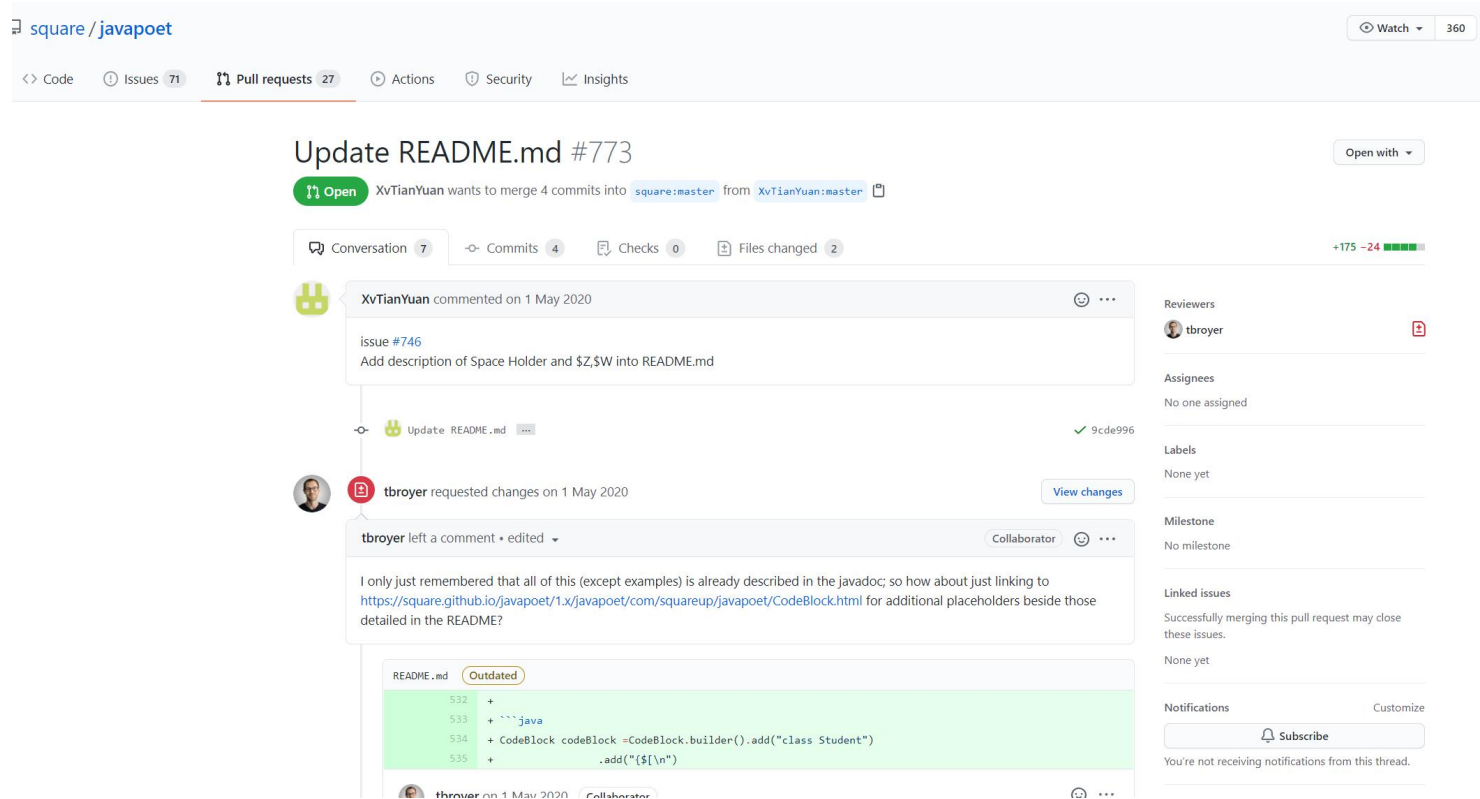

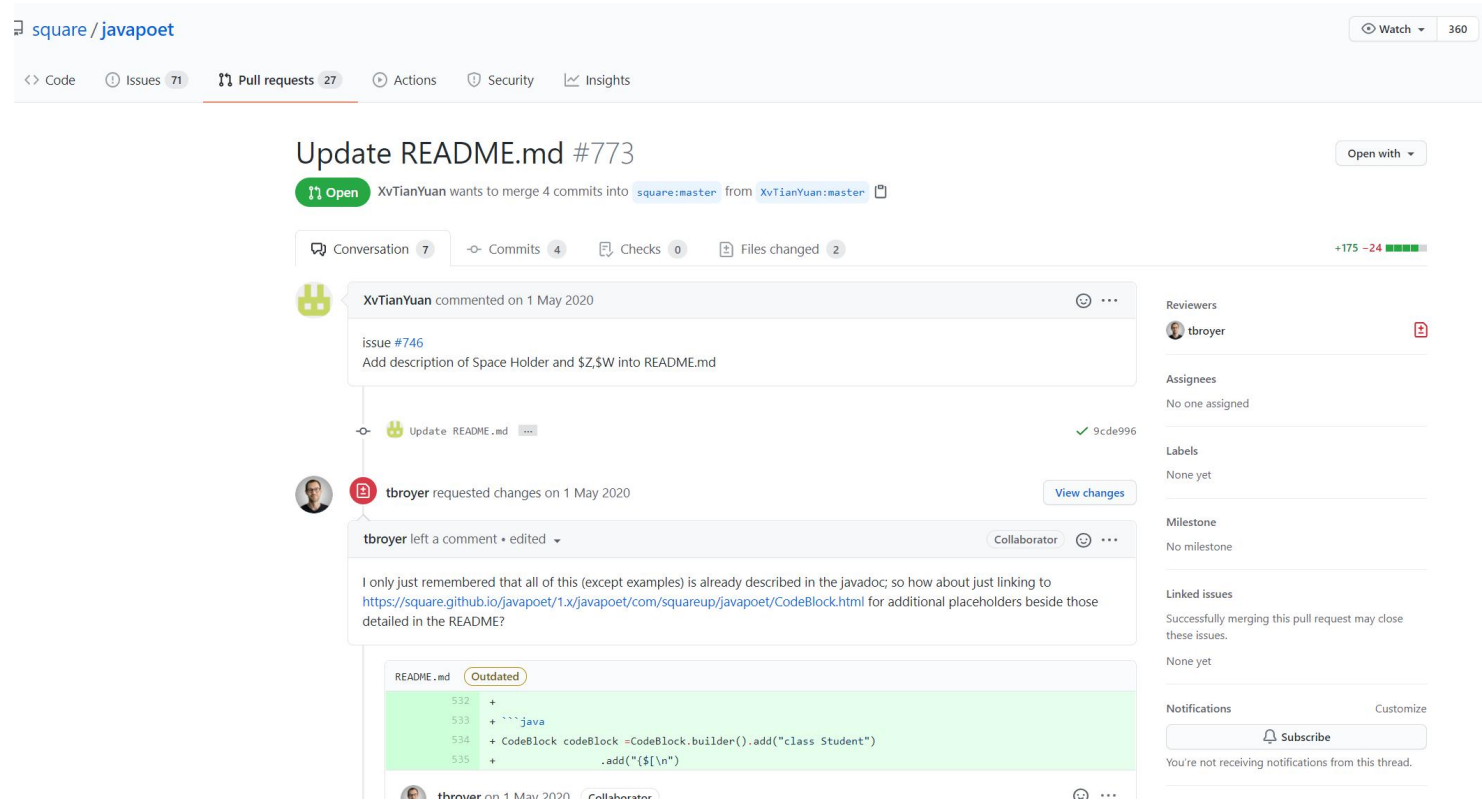➤No communication with the user and developer. Only fixing the way you like

Project Rules:
What not to do when making pull request (PR)?

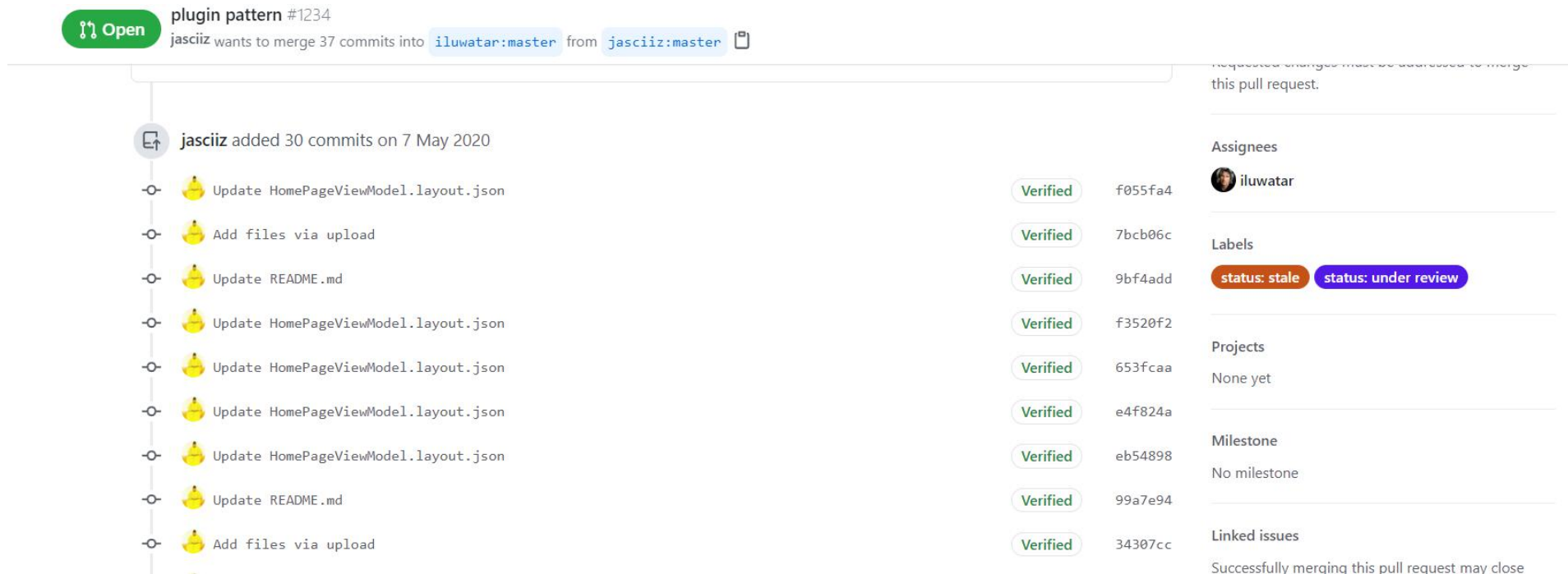➢Do not choose documentation related issue.

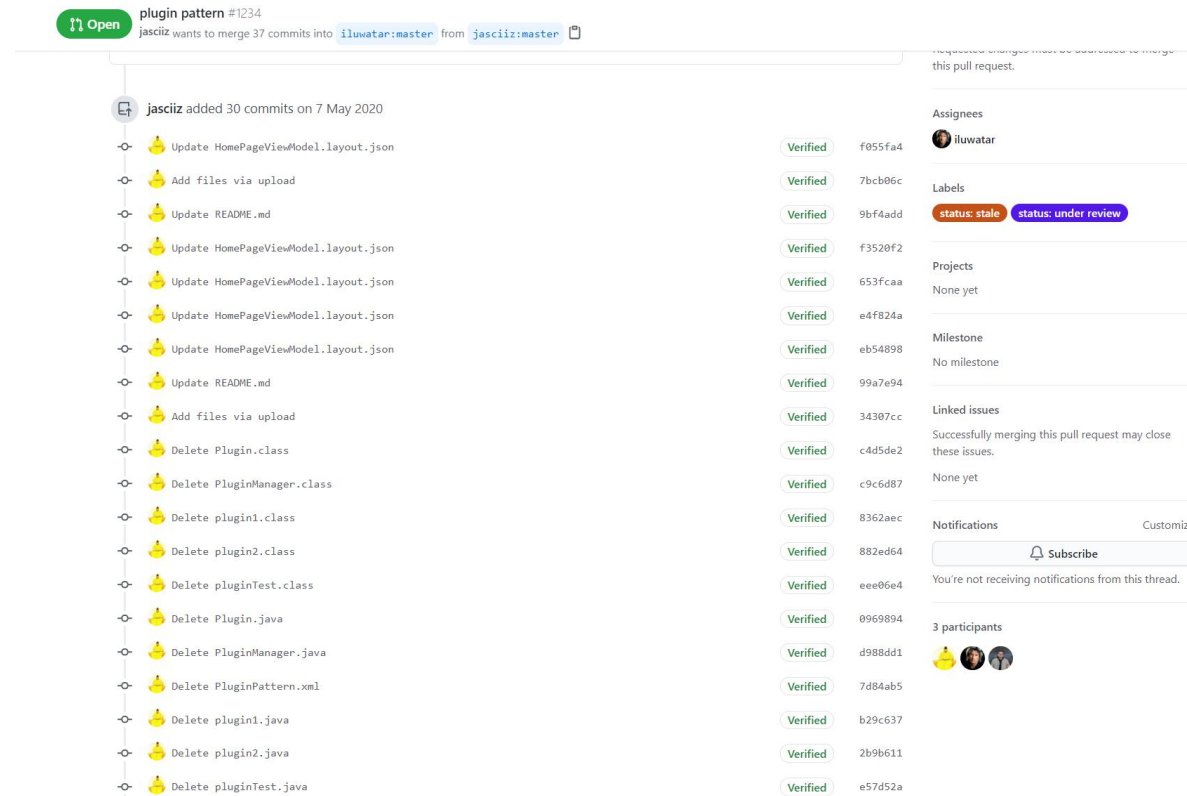# Project Rules:
What not to do when making pull request (PR)?

➢Do not choose documentation related issue.

Project Rules:
What not to do when making pull request (PR)?

➢Do not adding too many commits to your PR. You PR should be short
➢Make minimal changes to current code. A PR that change less lines will be more likely to be accepted!

Example from: https://github.com/iluwatar/java-design-patterns/pull/1234

Project Rules:
What not to do when making pull request (PR)?

➢Do not adding too many commits to your PR. You PR should be short

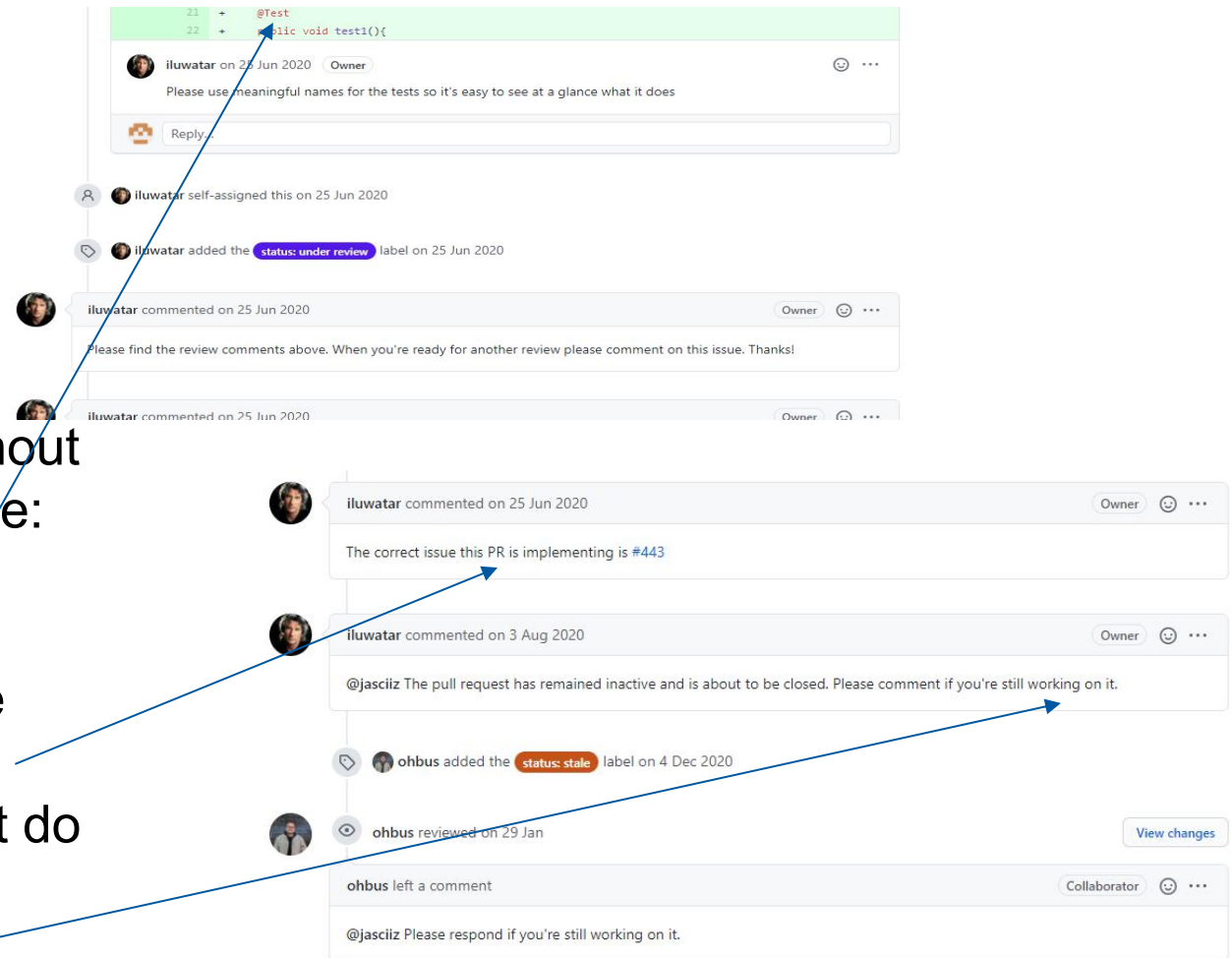From: https://github.com/iluwatar/java-design-patterns/pull/1234

Project Rules:
What not to do when making pull request (PR)?

➤ **Hit-and-run PR**: Commits and run away without responding. This is very bad practice because:
  ➤ Waste time of developer in code review
  ➤ Developers need to correct your mistake because you didn't even check if you are implementing the correct issue
  ➤ Developer mention you several times but do not respond. Leave a bad reputation in GitHub

Example from: https://github.com/iluwatar/java-design-patterns/pull/1234