SWE20001 – Development Project 1: Tools and Practices

Distinction Task <9.1D>

Duy Phuong Nguyen (101204984)

Lab: Friday 2:30pm

Tutor: Huai Liu

Team number: 3

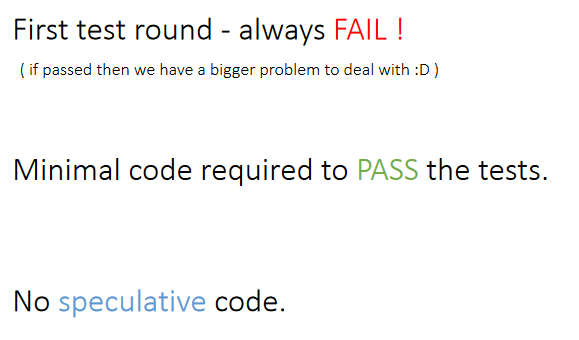
Henry Tran 102075482

Edward Knight 102700043

Duy Phuong Nguyen 101204984

# Test driven development (TDD)

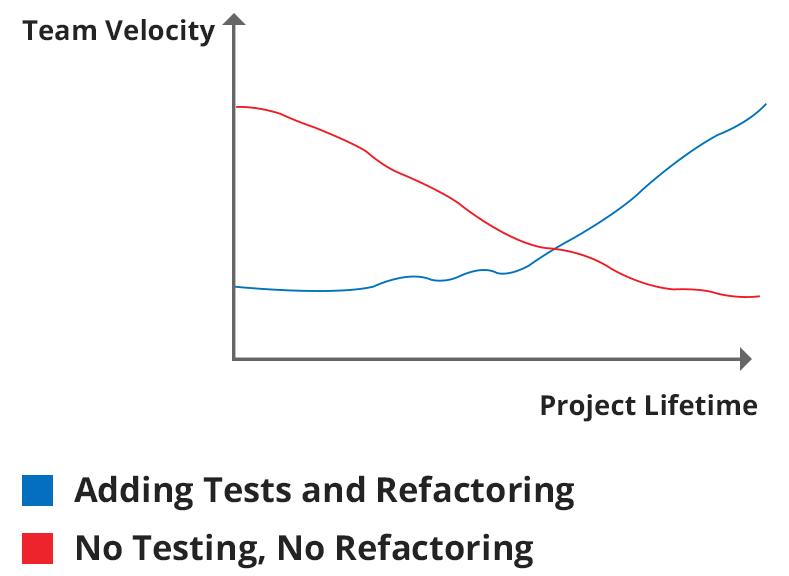
Test driven development is a development process which is very popular in software industry. The test will be created before the code to specify behaviours, show how the code will work. There are 3 main steps in TDD: The test is always failed in the first test round. Developer team need to implement more code to pass the test. The code then updated until all the requirements are passed with the shortest code.



# Pros and Cons of TDD process

**Pros:**

* TDD is a powerful tool for developer to archive a good product with quality designed code and good testing process in a shortest time.



Based on a report from xeonstack.com, TDD has a good impact on project lifetime and team velocity. Source[: https://www.xenonstack.com/blog/test-behaviour-driven-development/](:%20https:/www.xenonstack.com/blog/test-behaviour-driven-development/)

* Provide clear definitions/ requirement from customers. This helps to reduce bugs, errors, improves quality of codes.
* Splitting the project into smaller part using TDD process reduces the ratio of errors per line. Source: <https://en.wikipedia.org/wiki/Testdrivendevelopment#Keep_the_unit_small>
* Easy to refactor when new team member join the project or new requirements from customers.
* Detect any difficulty and errors in early stage of development.

**Cons:**

* Developer team must maintain the code and the test at the same time so TDD increase the workload of the project. This becomes bigger problem when we have a big project with many complex test (multiple level test with GUI, API, etc.)
* TDD process is write the simplest to pass the test at the beginning so it may create time expansion for the project from the start. Members find it difficult and boring. The simple code also reduces the vision of the project because we focus on testing specific/ single function not the whole project.

# How to use TDD

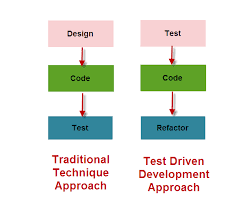
There are 3 level of applying TDD in a project

* The first level is beginner: It’s a basic level when developer team write the unit test before the code and then write sufficient code to pass the requirement of the test.
* The second level is Intermediate: Can apply different strategies to the test, able to find the bug with unit test and re-use previous unit test.
* The third level is Advanced: User is able to apply TDD in design and also technical domain (OOP, data transfer, managing teamwork, user-interface, etc.). Can create a map of unit test for characteristic/functionalities.

After understanding what your level is, Test-driven development can be deployed by the following steps:

1. Write an unit test based on requirement from customer to specify the purpose of project or a single function. The test case will be simple at first and then updated during the project. Other tests are also added to cover all the aspect of the project.
2. Run the test which is mean to be fail at first because the code is not added. We have to run the test to check if it is failed or not because the code/function is not implemented yet. If it passes the test, our test is wrong.
3. Write the simplest code to pass the test. Extra code will be added later and the unit test also updated to check the new code.
4. Refactoring the code to avoid duplicate and inconsistency. It also help the code clean and simple.
5. Repeat all the above steps until the project is finish.

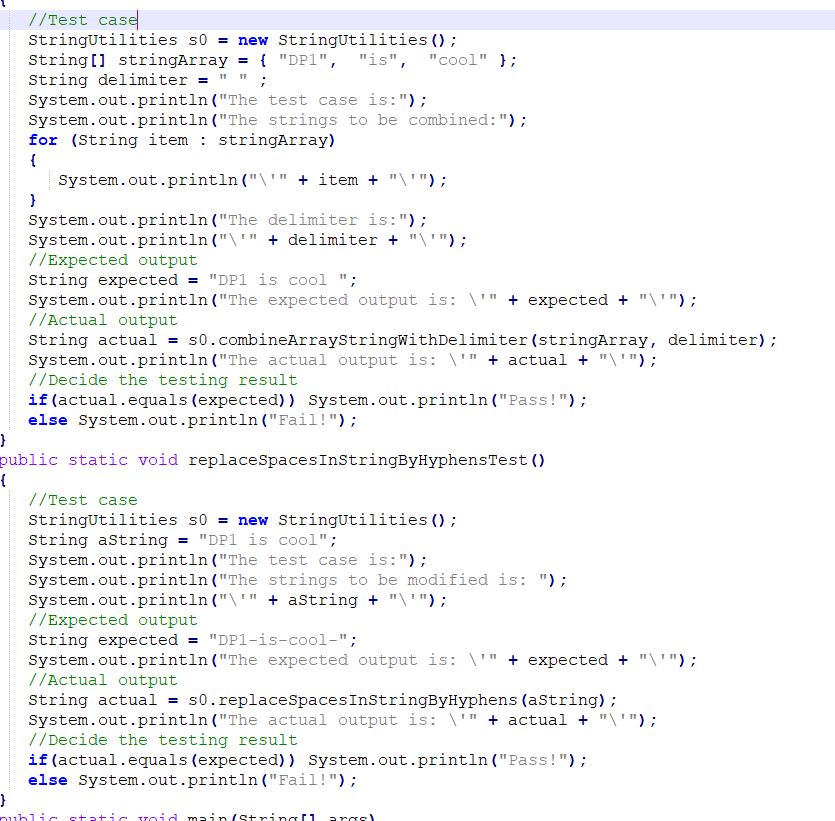
# What is different between Traditional testing and TDD



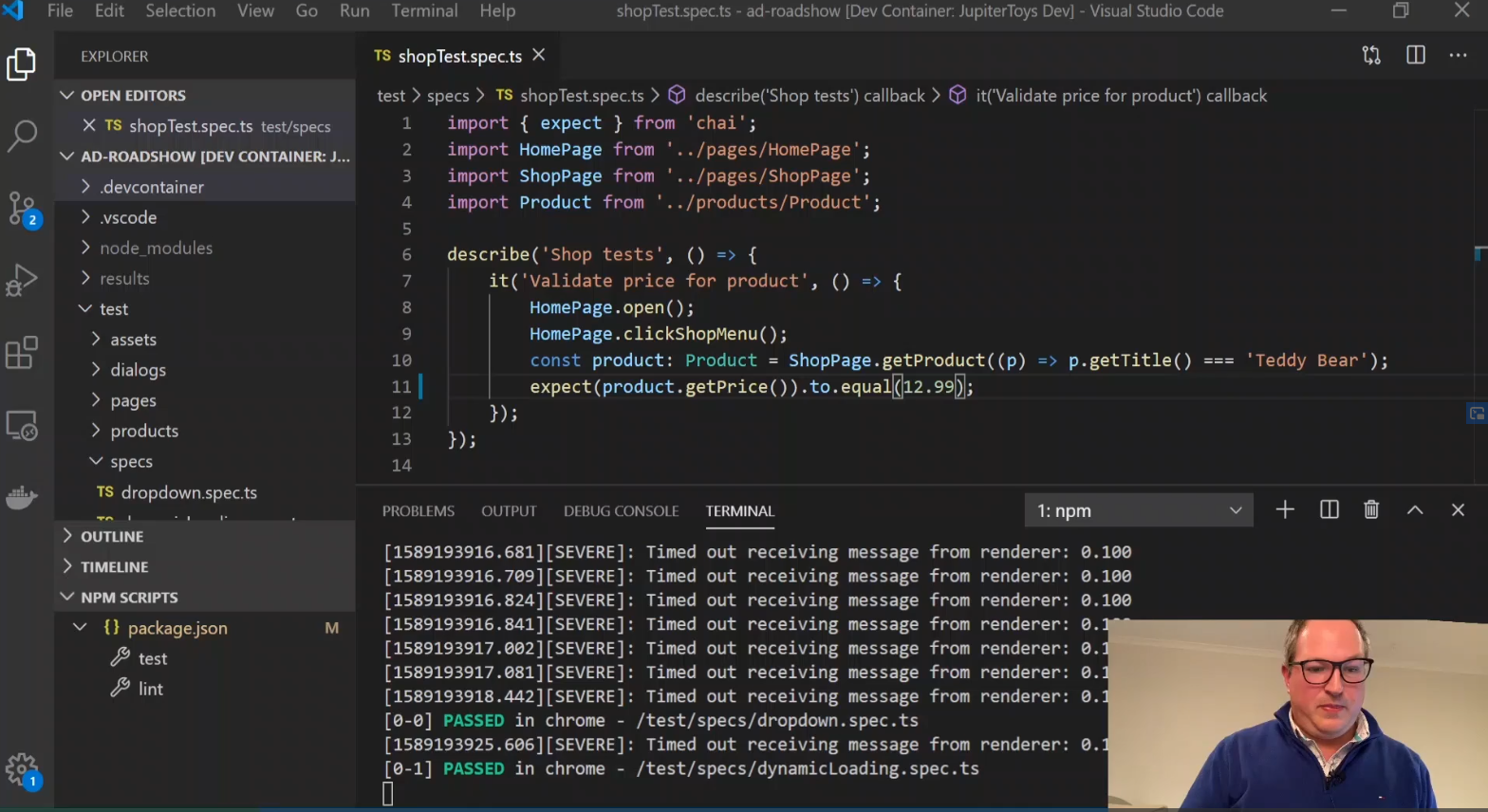
In traditional testing, all the code is designed and implemented before test case, unit test is created. The test is verification step to ensure all the code is working correctly. While TDD, the test is created before the code.

# Personal experience on TDD

We have a chance to learn about Java Unit Testing in task 7.2 D.

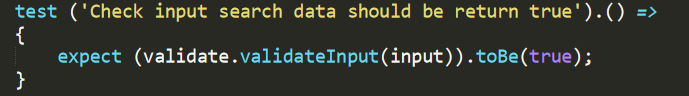


In week 9 lecture. Our guest lecture also demo a method to use JUnit test for website but it’s a bit hard for us to understand due to online lecture

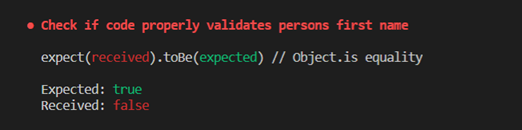


I have a chance to practice TDD in our open D/HD project. We tried to create a simple test at the beginning. It helps to provide me practical experience with TDD process.

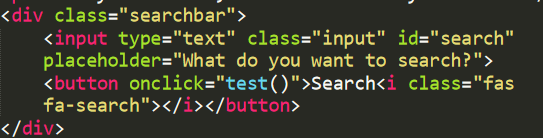
Our first idea of the project is creating a website with search bar function so our customer can find what they need with just a click. So I first tried to create a test to receive input from user and transfer that input to search function (google, bing).



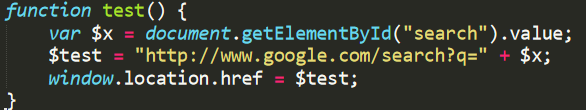
Simple test to check input data to be validate and the same with expected result.



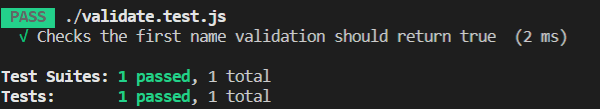
During the first time, the test is failed due to no code submission.



We added new code for the search bar function so it can submit input data to the sever.



We refractor the code by adding more code to the function so it can upload the input data to google.com.



We run the test again to make sure our code can pass the test. After that, we can implement more code for new features.

# Conclusion

Over the last decade, TDD becomes popular and powerful in software developing. It helps to produce clean code, bug-free content with test cases cover all aspect of the program. However, it has some drawbacks such as time consuming at early stage, increase work load for developers.