Review III

Review III

In this lesson you will read and remember the definitions of the Concepts we have studied so far.

Review III

1. Refactoring your Design
2. SRP Single Responsibility Principle
3. DRY Don't Repeat Yourself
4. Refactoring and Standup meetings
5. Definition is done
6. Ship out / Release Software with Quality and Value
7. Defend your Software from yourself and your peers
8. Technique #1 - Version Control with your code repository
9. Technique #2 - Control your dependencies
10. Functional Testing
11. Technique #3 - Always Exercise your Code-Unit Testing
12. Technique #4 -Let your peers understand your Code-Document it!

Review III

1. Refactoring your design: Refactoring is the process of modifying the structure of your code, without modifying its behavior.

2. SRP Single Responsibility Principle: Every object in your system should have a single responsibility, and all the object’s services should be focused on that single responsibility.

3. Refactoring and Standup meetings: Refactoring is the process of modifying the structure of your code, without modifying its behavior.

Review III

4. DRY Don’t Repeat Yourself: DRY is about having each piece of information and behavior in your system in a specific, single place. The DRY principle is a principle of Software Development aimed at reducing repetition of information of all kinds.

5. Definition is done: When everything is complete , then you have definition, it’s done.

6. Ship out / Release Software with Quality and Value: Your Software must always have awesome quality and awesome value, aim at Perfection, settle for Functionality.

Review III

7. Defend your software from yourself and your peers: Test Driven Development (TDD) is all about writing your Code with testing in mind. It’s about writing Tests before any code and letting these Tests drive your Implementation.

8. Technique #1 - Version Control with your code repository: Version Control is a super cool technology you can use, it’s one of the best Defensive tools you can use. Version Control lets you create a repository to keep your code in a single place to ease backup and recovery.

Review III

9. Technique #2 - Control your dependencies: You need to find a way to keep things independent, but working together at the same time to make Testing easier. You need to find a way to apply Independent Tests. Create Mock Objects to do this.

10. Functional Testing: Your tests need to guarantee your Software's optimal Functionality.

11. Technique #3 - Always Exercise your Code-Unit Testing: Unit testing is all about creating tests that run automatically to test the smallest components of the code for their business logic.

Review III

12. Technique #4 - Let your peers understand your Code-Document it! Inform your Customers of any changes you have made to the code Document your changes Analyze them along with your peers. Keep them informed with reasons.

Review III

We hope you liked the information. Remember you can always post your questions, should you have any.

Thank you