

Mohammadmahdi Farrokhy

Clean Code:

A Handbook of

Agile Software Craftsmanship

Chapter 09: Unit Tests

Test Driven Development

It's really important to write effective unit tests to ensure code quality, reliability, and maintainability. They provide us early detection of bugs during development. It's also so important to keep test code as clean as production code. Clean tests satisfy the need for documentation, as they describe the logical flow of a class or even the entire software.

The Three Laws of TDD

Test Driven Development (TDD) is a practice where tests are written before the actual code. TDD is guided by three laws:

1. Begin by writing a failing test that cannot be compiled.
2. Write just enough production code to make the failing test compile and pass logically.
3. Develop the production code only to pass the test, without exceeding the requirements.

Test Driven Development

- Tests Must Be Clean.

- As the product evolves, tests evolve too.
- Dirty tests lead to challenges during modification.
- An unclear test requires additional time and effort to cover new product code.
- Test code holds equal importance as product code. Test code is not a second class citizen; it demands careful thought, design and attention.

Test Driven Development

- Tests Give You Power.

- Unit tests empower your code by enhancing its flexibility, maintainability, and reusability.
- When you have tests in place, you gain the confidence to modify your code without fear.
- In the absence of tests, any modification becomes a potential bug.
- Regardless of your architecture's flexibility or the elegance of your design, the absence of tests leads to hesitation in making changes.

Clean Tests

- One Test – One Assertion

This rule might look a little much too strict, but it has a great benefit. Tests with only one assertion are fast and easy to understand. But we can be more flexible and change the rule into another:

- One Test – The Minimum Number Of Assertions Possible.

Clean Tests

- **One Test – One Concept**

- Best practice is to test a single concept inside each test method.
- Avoid lengthy test methods that test multiple subjects consecutively.
- This principle aligns with the earlier guideline and results to minimal assertions in test methods.

Clean Tests

- Tests Should Be F.I.R.S.T

Clean tests also follow five other guidelines:

1. **Fast:** Test should run quickly.
2. **Independent:** Test should not depend on other tests.
3. **Repeatable:** Test should produce the same result consistently.
4. **Self-Validating:** Test should have a boolean output.
5. **Timely:** Test should be written before production code, not after it.