Code Academy

Lecturer

Rokas Slaboševičius

Introduction to Software Testing

Data

Today you will learn

01 What is Software Testing?

02 Unit Testing

What is Software Testing?

Definition: Process of evaluating a software application to detect differences between given input and expected output.

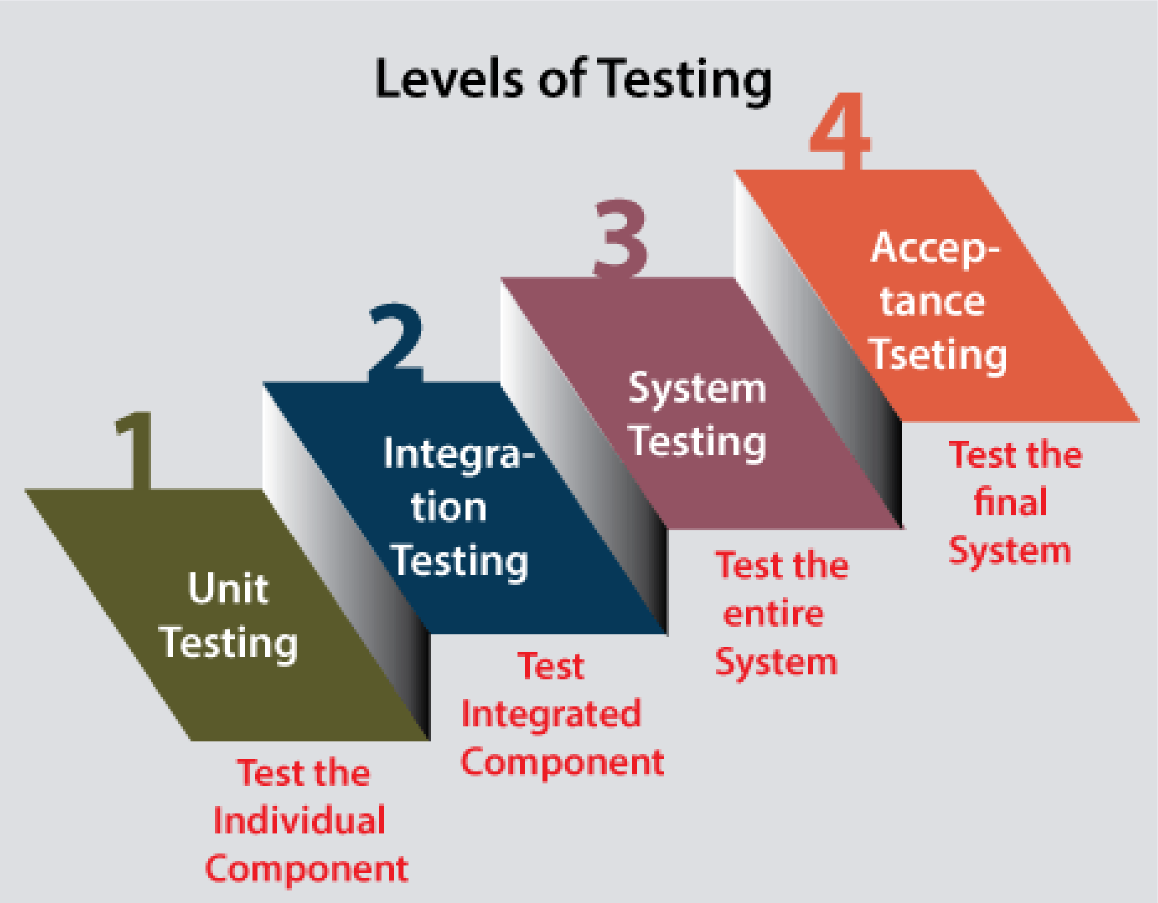
Goals: Ensure software reliability, performance, and quality.

Importance and Purpose of Testing

- Ensure software correctness, security, and usability
- Prevent defects and reduce development costs
- Improve user satisfaction and confidence in the software

Levels of Testing

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing



Unit Testing

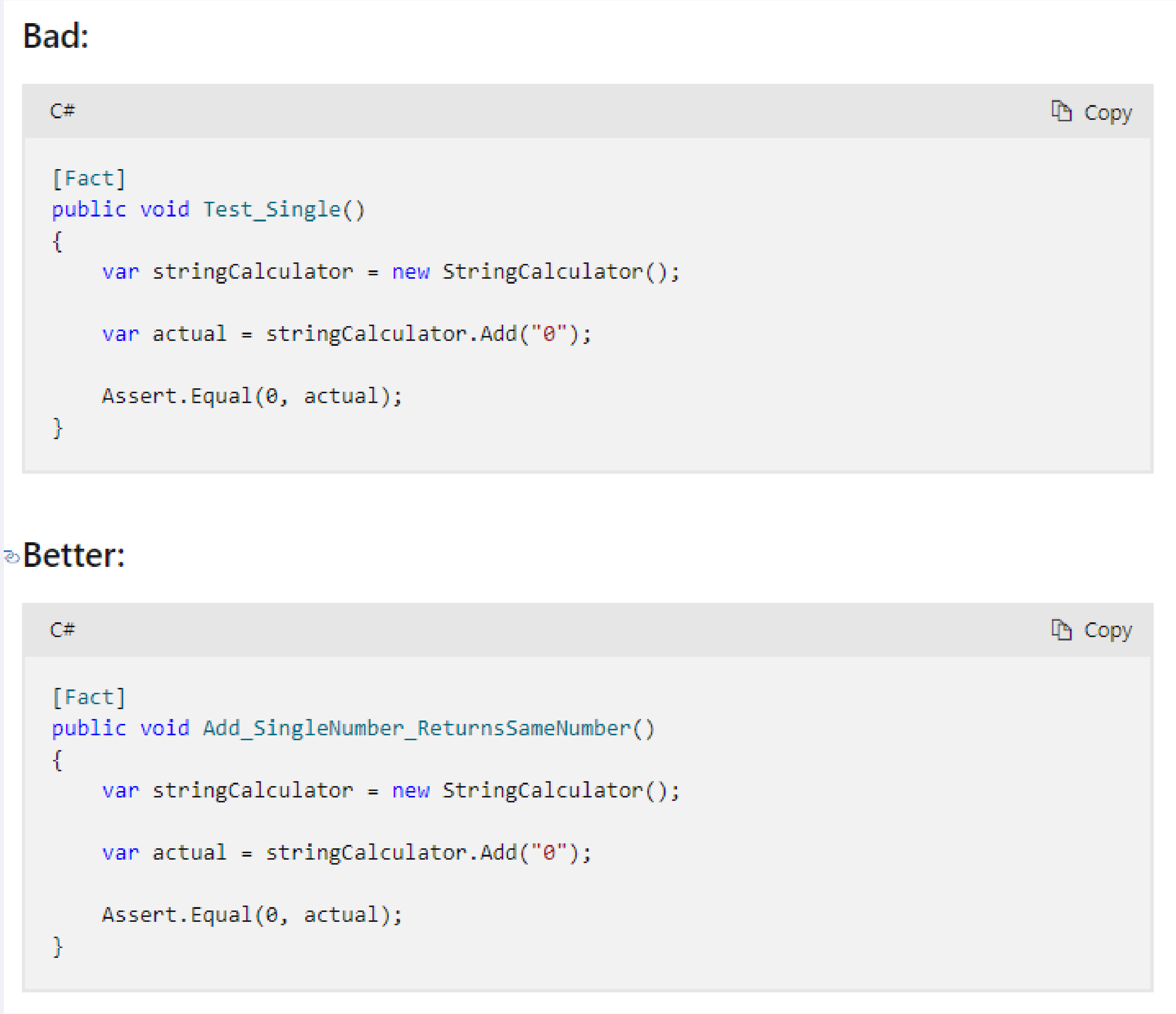
- Definition: Testing individual components or pieces of code to verify each part functions correctly.
- Purpose: Identify bugs at an early stage in development.

Good Unit Test

- Isolated: Tests a single piece of functionality
- Repeatable: Can be run multiple times with the same results
- Self-validating: Automatically checks the test result and provides immediate feedback
- Timely: Written just before the production code that makes the test pass

Naming your tests

- The name of the method being tested.
- The scenario under which it's being tested.
- The expected behavior when the scenario is invoked.



Bad:

```csharp
[Fact]
public void Test_Single()
{
    var stringCalculator = new StringCalculator();

    var actual = stringCalculator.Add("0");

    Assert.Equal(0, actual);
}
```

Better:

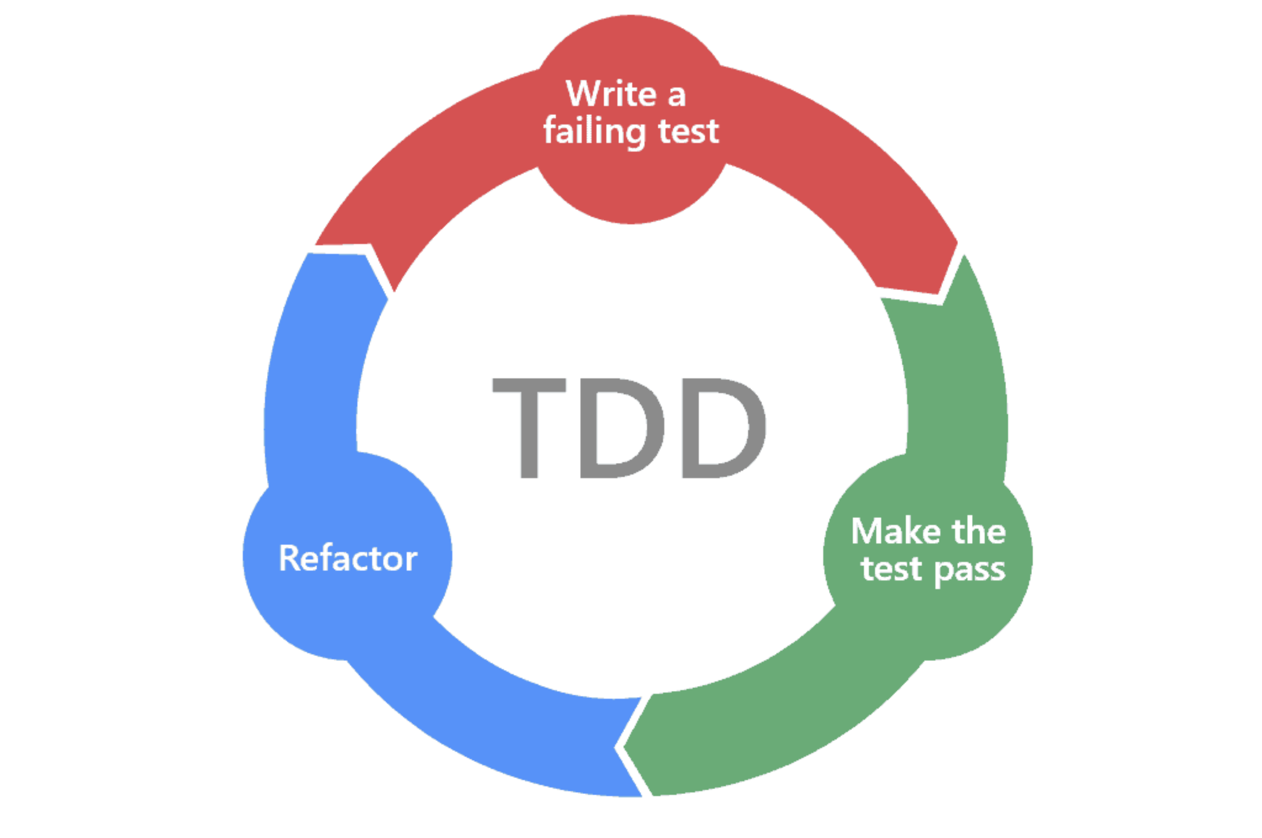
```csharp
[Fact]
public void Add_SingleNumber_ReturnsSameNumber()
{
    var stringCalculator = new StringCalculator();

    var actual = stringCalculator.Add("0");

    Assert.Equal(0, actual);
}
```

Test-Driven Development (TDD) Basics

- TDD - A software development approach where tests are written before writing the code itself.
- Benefits: Ensures a robust and error-free codebase, promotes modular code, improves code coverage.


Write a failing test
TDD
Make the test pass
Refactor