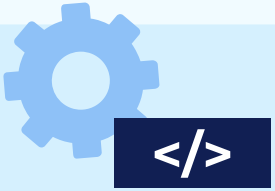
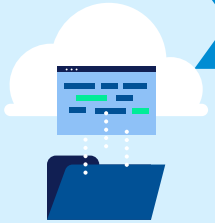


Unit Testing using C#

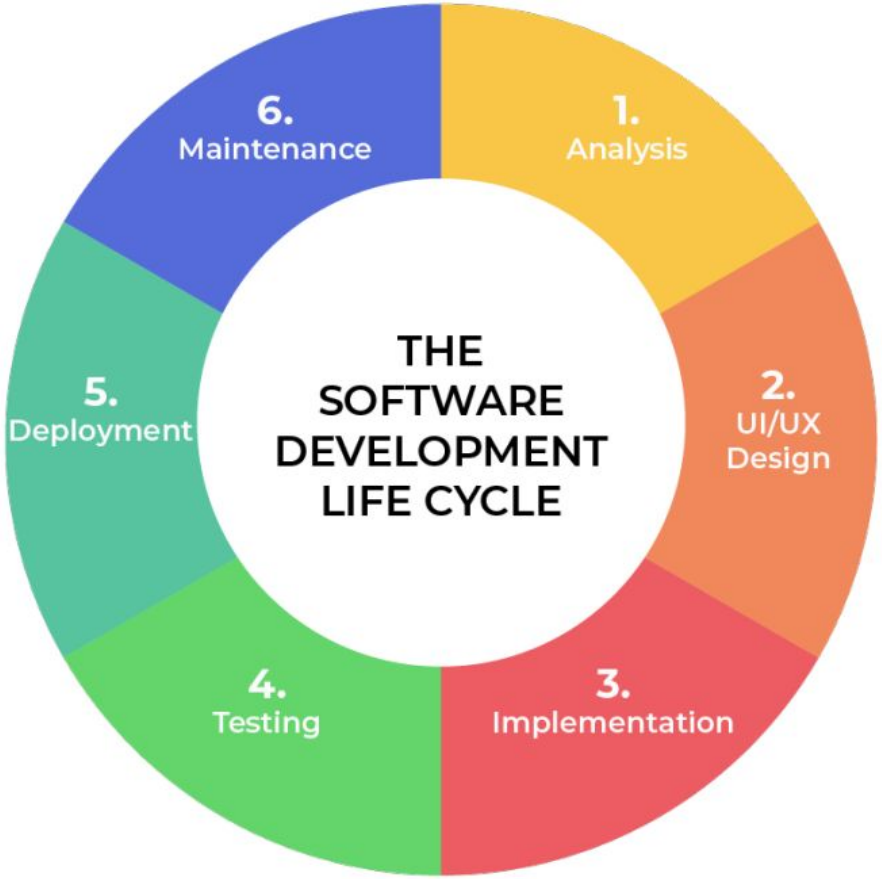
XUnit Framework

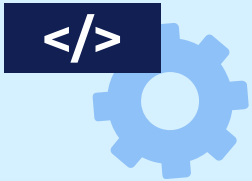


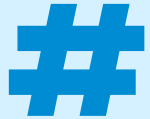
By Rehab Hesham

</>

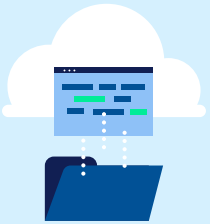
Software Development Life Cycle





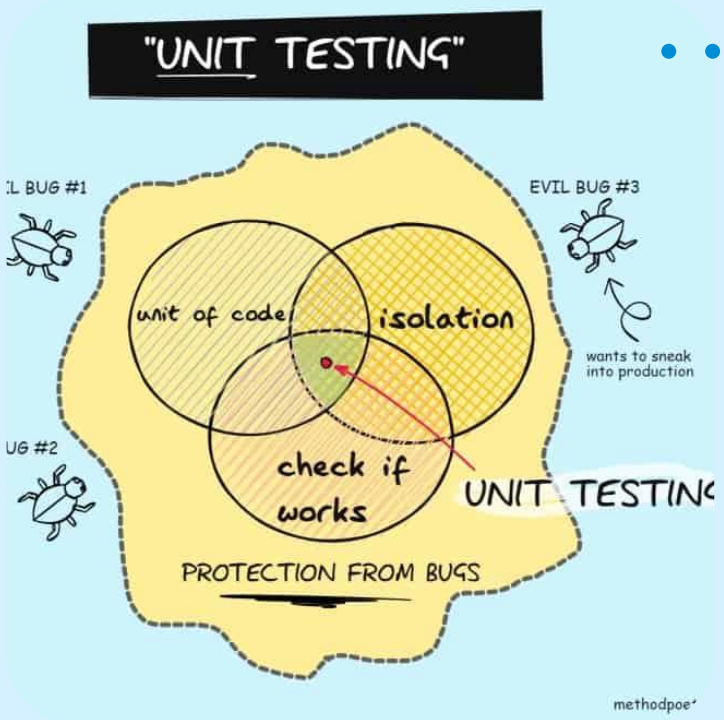


What is Unit Testing?



What is Unit Testing?

Unit testing is a process of verifying that individual units of code (methods, classes, etc.) work as intended. You can be confident that your code works as expected by writing unit tests and then running them as part of your build process.



TYPES OF SOFTWAR E TESTING

Unit Testing

Test specific function only. Test cases only are used.

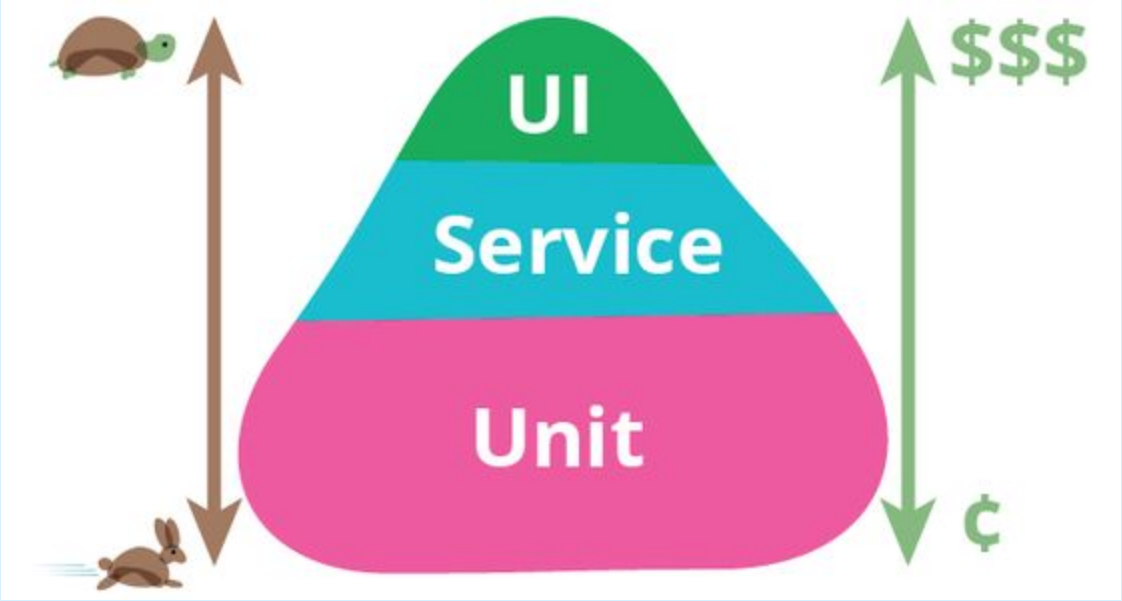
Integration Testing

Test multiple behaviors together, test scenarios

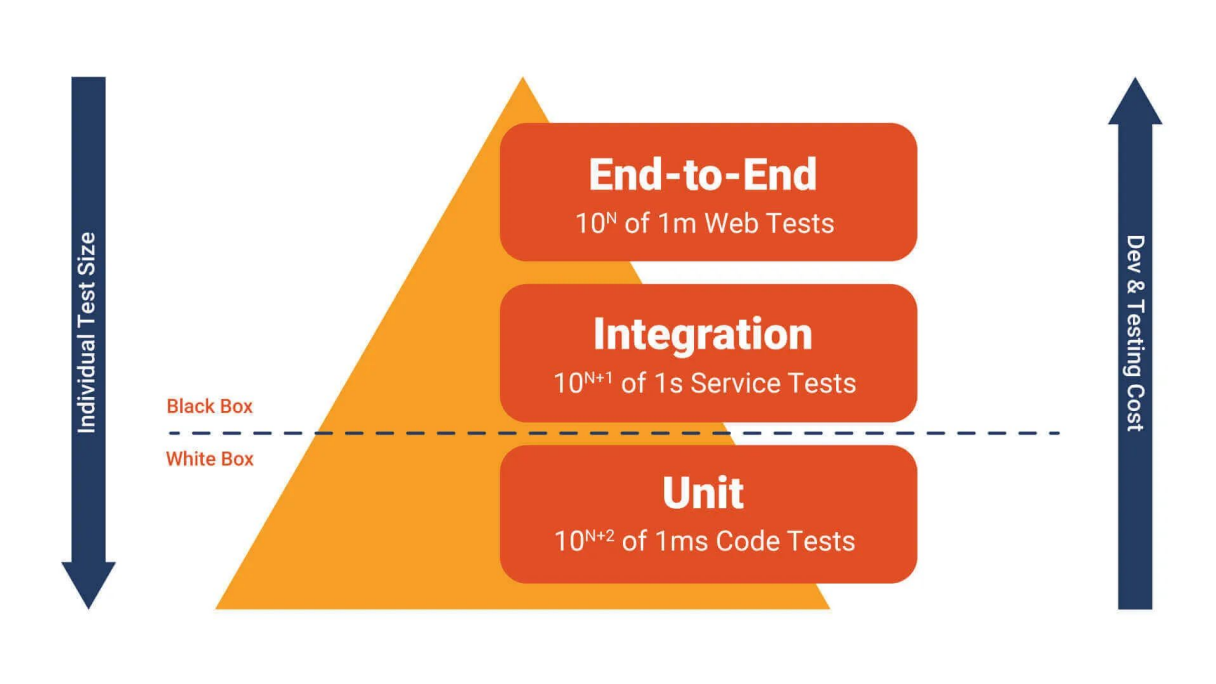
Acceptance Testing

Done by the client before delivering.

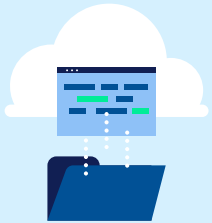
Test Pyramid



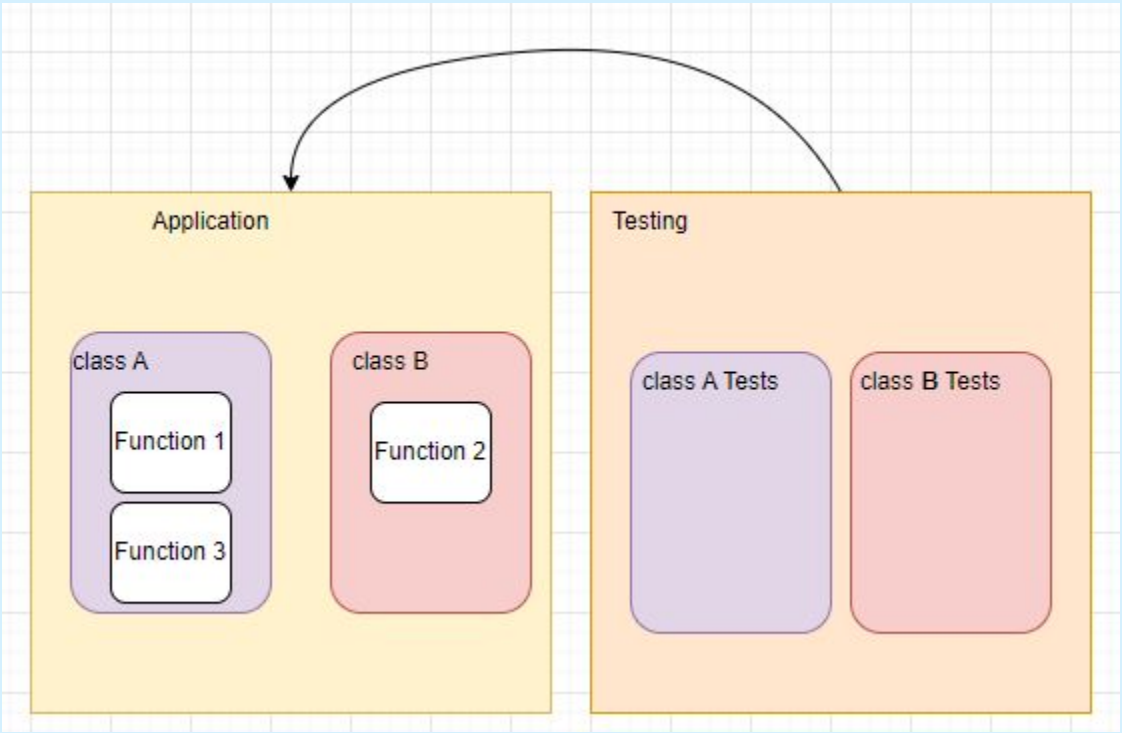
Test Pyramid

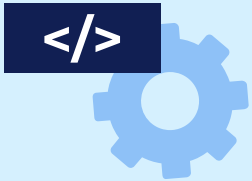


Testing is a Consumer

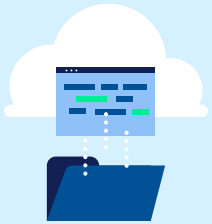


Testing is a Consumer





What is a unit testing framework ?



Unit Testing Framework

01

xUnit

It’s open-source, and you can use it on any platform that supports .NET.

02

NUnit

It’s open-source and has many features that make it easier to write unit tests.

03

MS Test

It’s popular because it’s easy to use and integrates well with Visual Studio.

Naming Convention

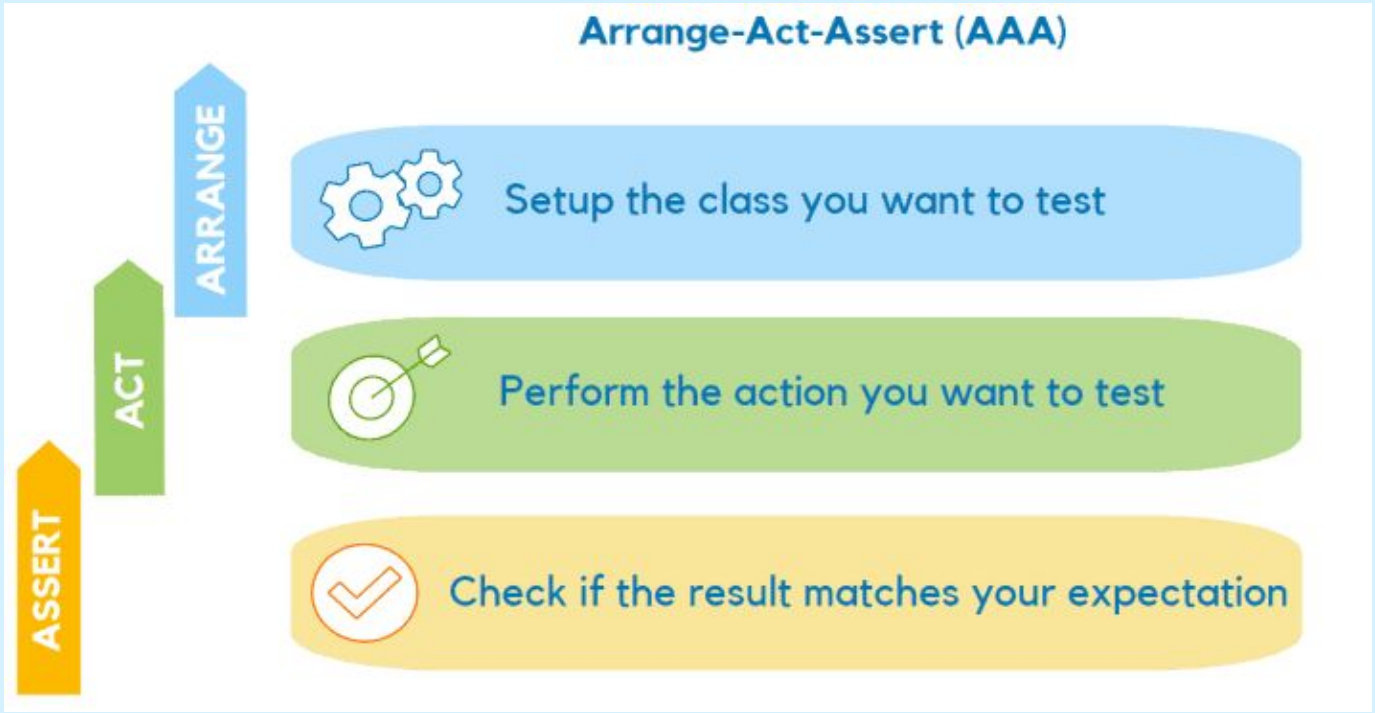
class

[className]Tests

Methods

[MethodName]_[caseUnderTest]_[ExpectedBehavior]

Structure of a unit test AAA



Rules of unit testing

- Test each function independently.
- It has one path (no If / else)
- Doesn’t depend on other functions.
- Avoid logic in tests
- Using clear convention (Naming testing pattern)

Boolean Assertions

Method
Assert.True(bool actual)
Assert.False(bool actual)

String Assertions

Method
Assert.Equal(expectedString, actualString);
Assert.EndsWith(expectedString, stringToCheck);
Assert.StartsWith(expectedString, stringToCheck);
Assert.Equal(expectedString, actualString, ignoreCase: true);
Assert.StartsWith(expectedString, stringToCheck, StringComparison.OrdinalIgnoreCase);

String Assertions

Method
var regEx = @"\A[A-Z0-9+_.-]+@[A-Z0-9.-]+\Z"; Assert.DoesNotMatch(regEx, "this is a text"); Assert.Matches(regEx, "this is a text");

Numeric Assertions

Method
Assert.InRange<T>(T actual, T low, T high)
Assert.NotInRange<T>(T actual, T low, T high)

Equality Assertions

Method
Assert.Equal<T>(T expected, T actual)
Assert.Equal<T>(T expected, T actual, int precision)
Assert.NotEqual<T>(T expected, T actual)

Reference Assertions

Method
Assert.Null(object object)
Assert.NotNull(object object)
Assert.Same(object expected, object actual)
Assert.NotSame(object expected, object actual)

Type Assertions

Method
Assert.IsAssignableFrom<T>(object obj)
Assert.IsType<T>(object obj)

Exception Assertions

Method
Assert.Throws(System.Exception expectedException, Action testCode)
Assert.Throws<T>(Action testCode) where T : System.Exception

Collection Assertions

Method
Assert.Empty(IEnumerable collection)
Assert.NotEmpty(IEnumerable collection)
Assert.Contains<T>(T expected, IEnumerable<T> collection)
Assert.DoesNotContain<T>(T expected, IEnumerable<T> collection)