Types of Tests:

Unit Tests – tests a unit of an application without its external dependencies

Pro

-Cheap to write

-Execute fast

Con

-no external dependencies so not reliable.

Integration Test – Test the application with its external dependencies

Pro

-Gives more confidence in the health of the application

Con

-Takes longer to execute

End-to-End test - Record the interaction of a user with an application

Pro

-Gives the greatest amount of confidence

Con

-Very Slow

-Very Brittle

Test Pyramid

-Favor Unit tests over End to End Tests

-Cover unit test gaps with integration tests

-Use end-to-end tests sparingly

Testing Frameworks

-Nunit

-MSTest

-xUnit

-Resharper

-Rider

Test Driven Development

-Write a failing test

-Write the simplest code to make the test pass

-Refactor if necessary

Benefits

-Testable Source Code

-Full coverage by Tests

-Simpler Implementation

Fundamentals of unit testing

Characterizes of good tests

No Logic

Clean, readable, maintainable

Isolated

Not too specific/general

What to test and what not to test

Test

Query:

Test all execution paths and resolve in the right value

Command:

Does an action

May return a value

Test the outcome

Don’t Test:

Language Features

3rd Party Code

Naming and Organizing Tests

Number of tests => number of execution paths

[MethodName]\_[Scenario]\_[ExpectedBehaviour]

[]