

Untestable Code - Thoughts Snippets.

Recalling the difference between Mocking and Spying

Mocks create scaffolding onto which you must attach the body of the desired methods.

Spies create scaffolding around already existing code, giving you the capability to override that already existing behaviour.

Verification is the process of identifying if a method behaves the way you expected it to behave, so it is not assessing state values. Verification is a type of Assertion. Both Mocks and Spies can be used to perform Verification.

NSubstitute's `Returns()` call can be passed a value or a function (including a lambda expression) that returns a value. Most people think that only interfaces can be mocked. In actual fact, classes can be mocked. In this context, you're creating a Spy.



When substituting a class with NSubstitute using the `For<>()` method, all virtual methods will be stubbed and return the default value for the return type of the method. All non virtual methods are called directly and are not stubbed.

Instead, use `ForPartsOf<>()`. It only creates a stub for methods that you specify using `spy.Configure().the_method_you_want_to_override()`

NSubstitute has a neat little feature called **When...Do...**