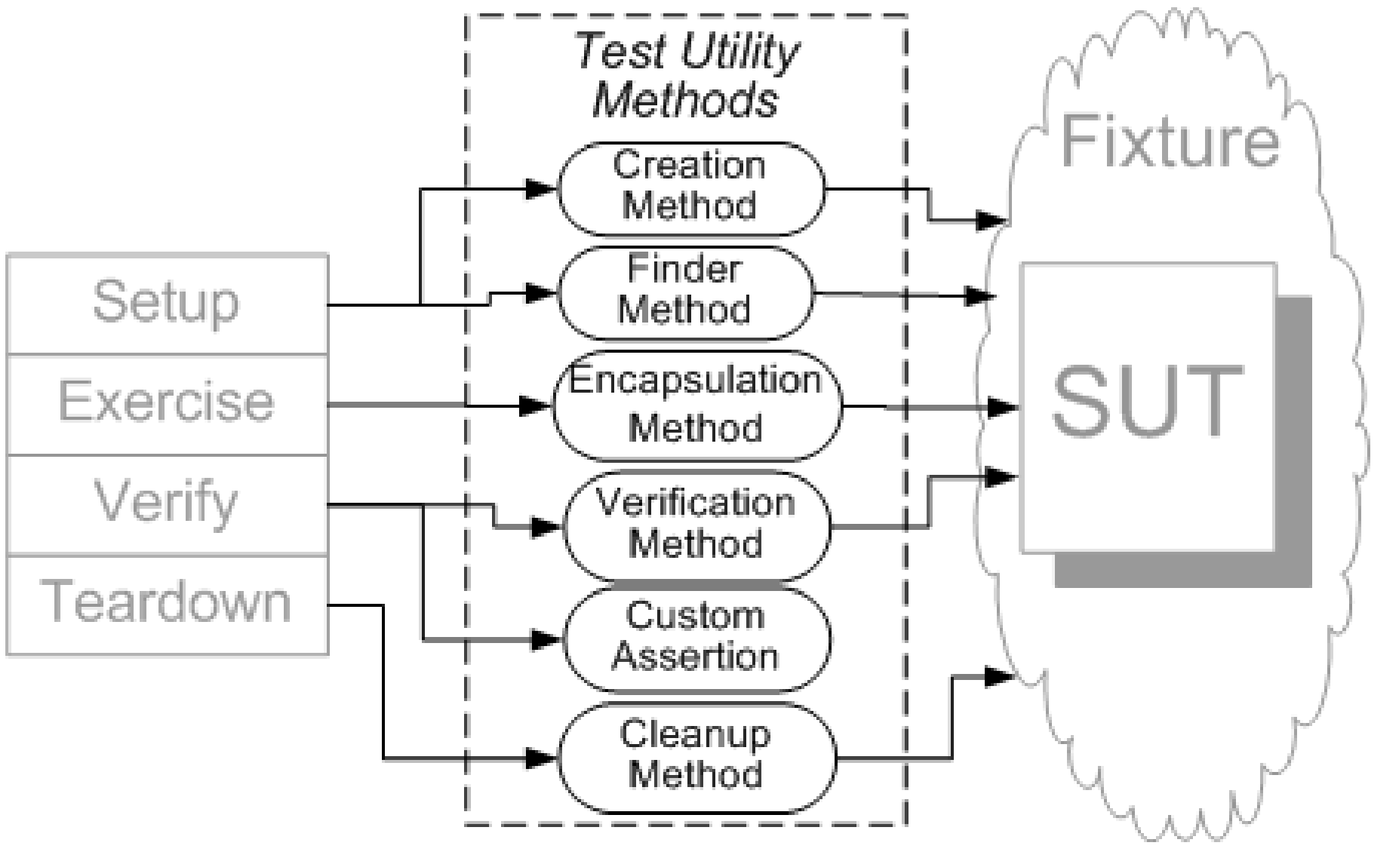
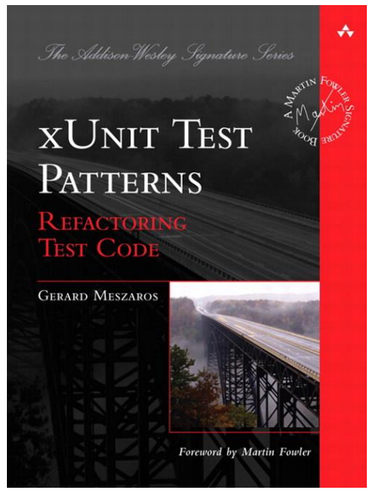
Helper Methods

Helper Methods

- Decrease Code Duplicates
- Decrease maintenance
- Increase readability & understandabilty of tests
- Increase productivity
- Can increase overall complexity of a test suite
- Inject dependencies between tests

Helper Methods





Creation Methods

Creation Method

```
[TestMethod]
public void SavedElementsShallStoreDataInDataBase()
{
    var repository = new StubRepository();
    var sut = new ViewModel(repository);

    var data = new Data();
    data.Id = "id";
    data.Payload = "data";

    sut.Save(data);

    Assert.IsTrue(sut.Data.Any(x => x.Id == data.Id));
}
```

Creation Method

```
[TestMethod]
public void SavedElementsShallStoreDataInDataBase()
{
    var repository = new StubRepository();
    var sut = new ViewModel(repository);

    var data = new Data();
    data.Id = "id";
    data.Payload = "data";

    sut.Save(data);

    Assert.IsTrue(sut.Data.Any(x => x.Id == data.Id));
}
```

Creation Method

```
[TestMethod]
public void SavedElementsShallStoreDataInDataBase()
{
    var repository = new StubRepository();
    var sut = new ViewModel(repository);

    var data = CreateTestData();

    sut.Save(data);

    Assert.IsTrue(sut.Data.Any(x => x.Id == data.Id));
}
```

Creation Method

```
[TestMethod]
public void SavedElementsShallStoreDataInDataBase()
{
    var repository = new StubRepository();
    var sut = new ViewModel(repository);

    var data = CreateValidTestData();

    sut.Save(data);

    Assert.IsTrue(sut.Data.Any(x => x.Id == data.Id));
}
```

Creation Method

```
[TestMethod]
public void SavedElementsShallStoreDataInDataBase()
{
    var repository = new StubRepository();
    var sut = new ViewModel(repository);

    var data = CreateTestData("id");

    sut.Save(data);

    Assert.IsTrue(sut.Data.Any(x => x.Id == data.Id));
}
```

Creation Method

```
[TestMethod]
public void SavedElementsShallStoreDataInDataBase()
{
    var repository = new StubRepository();
    var sut = new ViewModel(repository);

    var data = CreateTestData();

    sut.Save(data);

    Assert.IsTrue(sut.Data.Any(x => x.Id == data.Id));
}
```

Creation Method

```
[TestMethod]
public void SavedElementsShallStoreDataInDataBase()
{
    var repository = new StubRepository();
    var sut = new ViewModel(repository);

    var data = CreateTestData();

    sut.Save(data);

    Assert.IsTrue(sut.Data.Any(x => x.Id == data.Id));
}
```

Creation Method

```
[TestMethod]
public void SavedElementsShallStoreDataInDataBase()
{
    var sut = InitializeSut();

    var data = CreateTestData();

    sut.Save(data);

    Assert.IsTrue(sut.Data.Any(x => x.Id == data.Id));
}
```

Sut Factory Method

```
[TestMethod]
public void SavedElementsShallStoreDataInDataBase()
{
    var sut = ViewModelSetup.CreateSut();

    var data = CreateTestData();

    sut.Save(data);

    Assert.IsTrue(sut.Data.Any(x => x.Id == data.Id));
}
```

Subject Test Mother

```
[TestMethod]
public void SavedElementsShallStoreDataInDataBase()
{
    var sut = ViewModelSetup.CreateSut();

    var data = DataGenerator.CreateTestData();

    sut.Save(data);

    Assert.IsTrue(sut.Data.Any(x => x.Id == data.Id));
}
```

Subject Test Mother

```
[TestMethod]
public void SavedElementsShallStoreDataInDataBase()
{
    var sut = ViewModelSetup.CreateSut();

    var data = DataGenerator.CreateTestData();

    sut.Save(data);

    Assert.IsTrue(sut.Data.Any(x => x.Id == data.Id));
}
```

Initialize Method

```
private ViewModel sut;

[TestInitialize]
public void Initialize()
{
    var repository = new StubRepository();
    sut = new ViewModel(repository);
}
```

Encapsulation Methods

Encapsulation Method

```
[TestMethod]
public void ControlShallBeCollapsedWhenAllButtonsAreInvisible()
{
    var sut = CreateSut();

    eventAggregator.GetEvent<ActionButtonActivationEvent>()
                   .Publish(VisibleActionButtons.None);

    Assert.AreEqual(Visibility.Collapsed, sut.Visibility);
}
```

Encapsulation Method

```
[TestMethod]
public void ControlShallBeCollapsedWhenAllButtonsAreInvisible()
{
    var sut = CreateSut();

    eventAggregator.GetEvent<ActionButtonActivationEvent>()
                   .Publish(VisibleActionButtons.None);

    Assert.AreEqual(Visibility.Collapsed, sut.Visibility);
}
```

Encapsulation Method

```
[TestMethod]
public void ControlShallBeCollapsedWhenAllButtonsAreInvisible()
{
    var sut = CreateSut();

    SetActionButtonState(VisibleActionButtons.None);

    Assert.AreEqual(Visibility.Collapsed, sut.Visibility);
}
```

Encapsulation Method

```
public void SetActionButtonState(VisibleActionButtons state)
{
    eventAggregator.GetEvent<ActionButtonActivationEvent>()
                   .Publish(state);
}

public void DisableActionButtons()
{
    eventAggregator.GetEvent<ActionButtonActivationEvent>()
                   .Publish(VisibleActionButtons.None);
}
```

Encapsulation Method

```
[TestMethod]
public void ControlShallBeCollapsedWhenAllButtonsAreInvisible()
{
    var sut = CreateSut();

    DisableActionButtons();

    Assert.AreEqual(Visibility.Collapsed, sut.Visibility);
}
```

Encapsulation Method

```
[TestMethod]
public void ControlShallBeCollapsedWhenAllButtonsAreInvisible()
{
    Initialize();

    DisableActionButtons();

    Assert.AreEqual(Visibility.Collapsed, sut.Visibility);
}
```

Encapsulation Method

```
[TestMethod]
public void ControlShallBeCollapsedWhenAllButtonsAreInvisible()
{
    DisableActionButtons();

    Assert.AreEqual(Visibility.Collapsed, sut.Visibility);
}
```