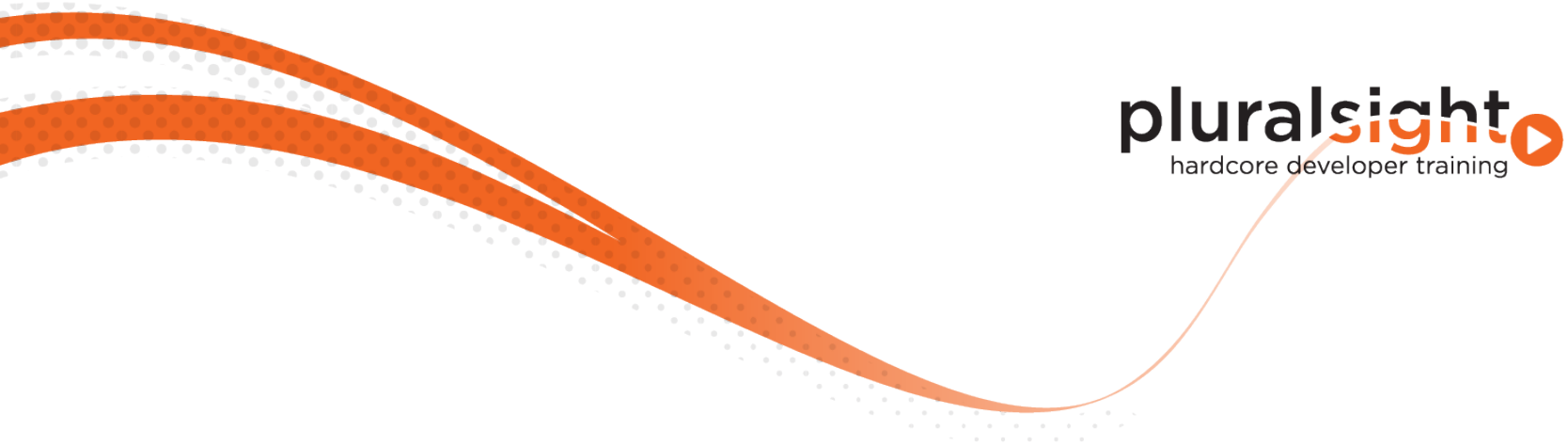
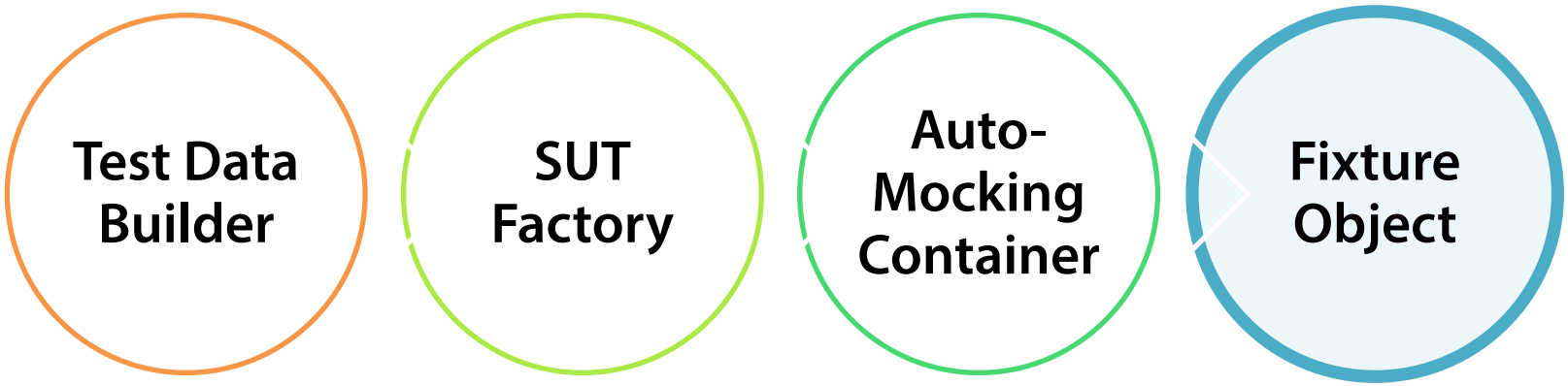
Advanced Unit Testing
Test Utility Code

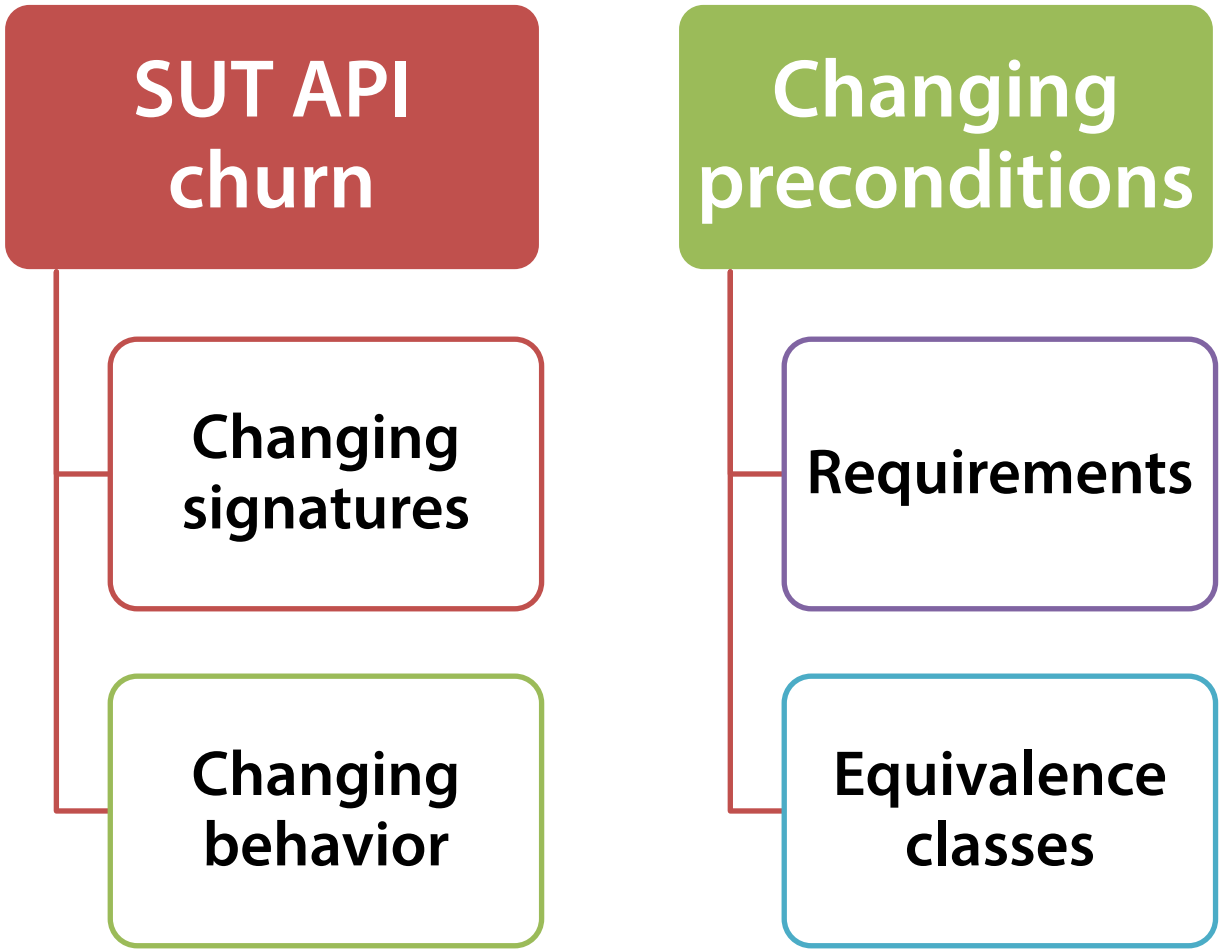
Mark Seemann
http://blog.ploeh.dk



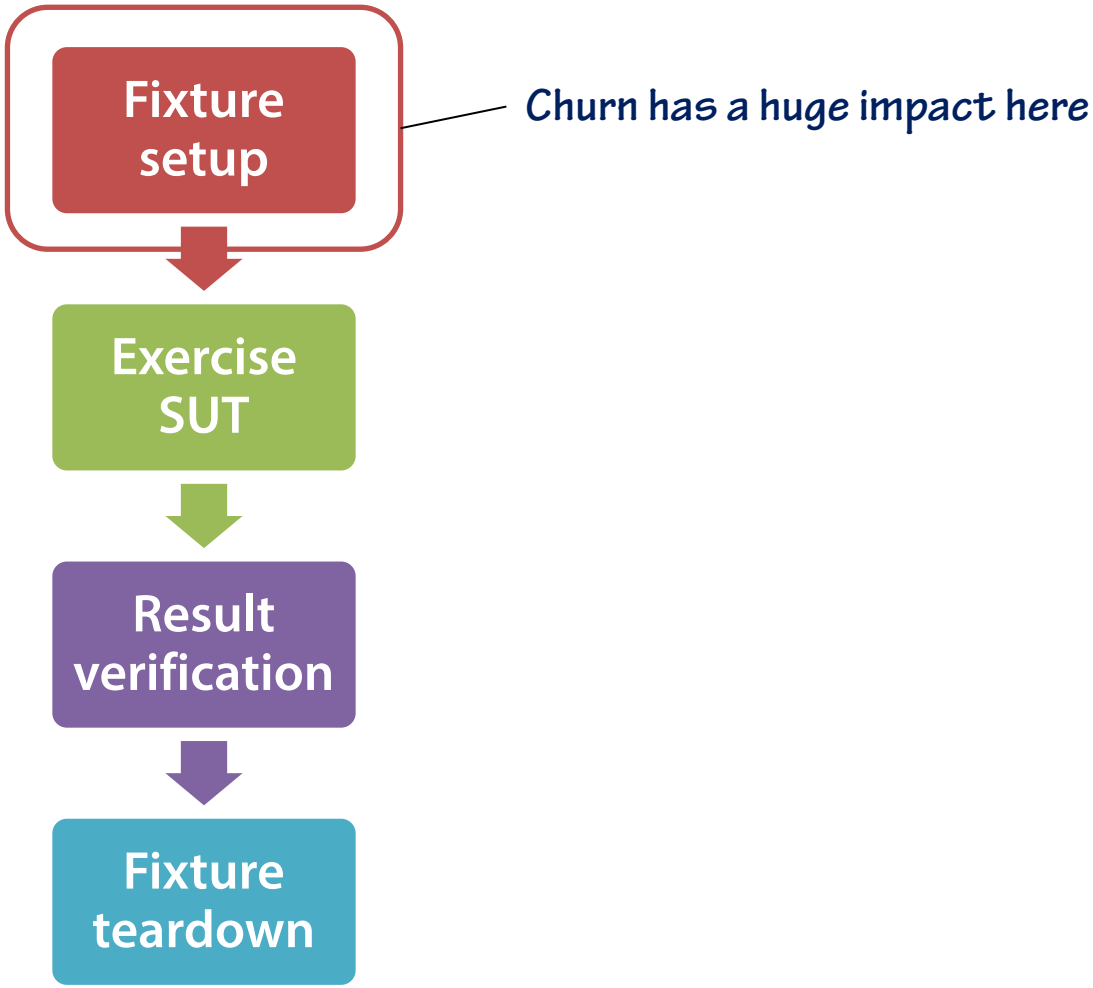
Outline


Test Data Builder
SUT Factory
Auto-Mocking Container
Fixture Object

Brittle tests



Four-Phase Test



Demo

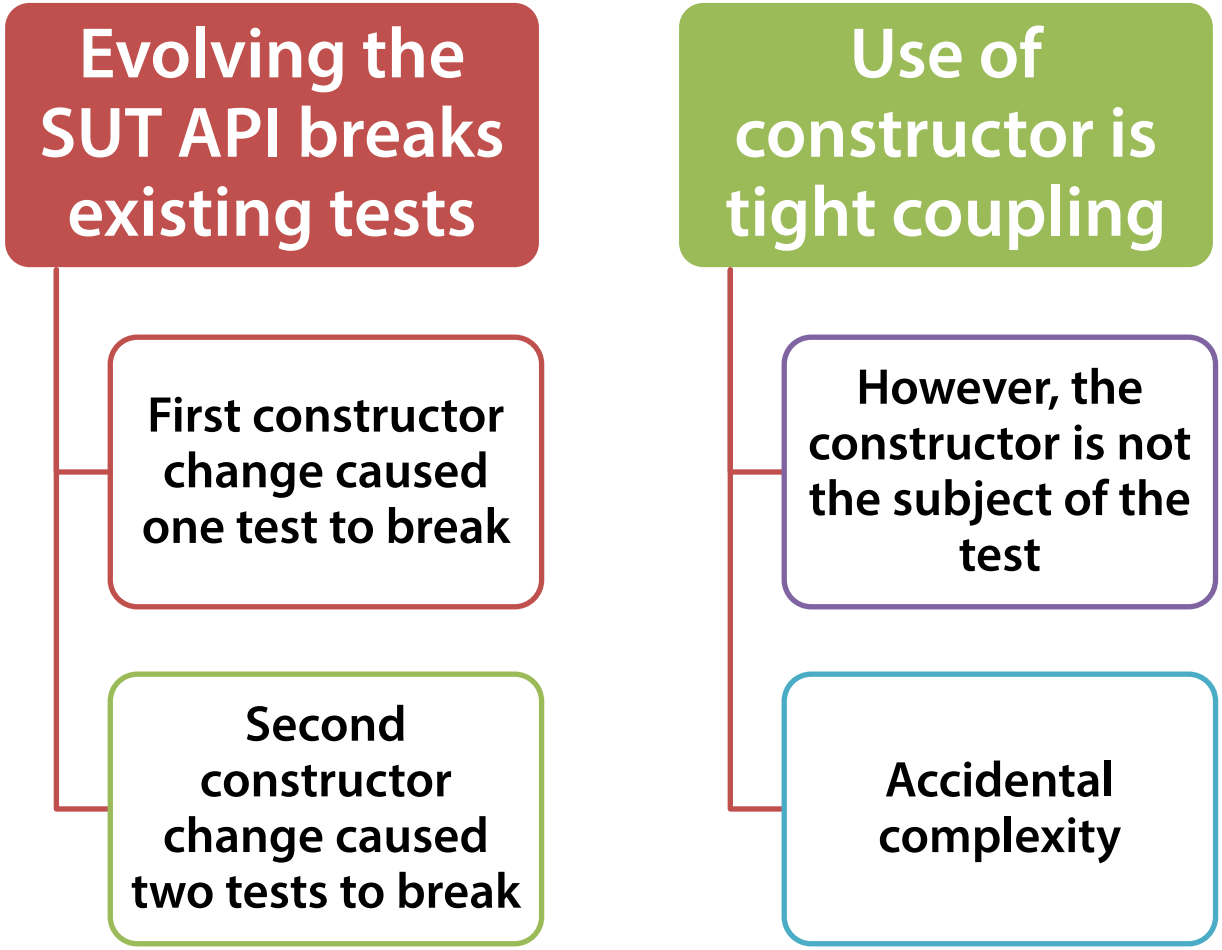
Motivating example

TDD

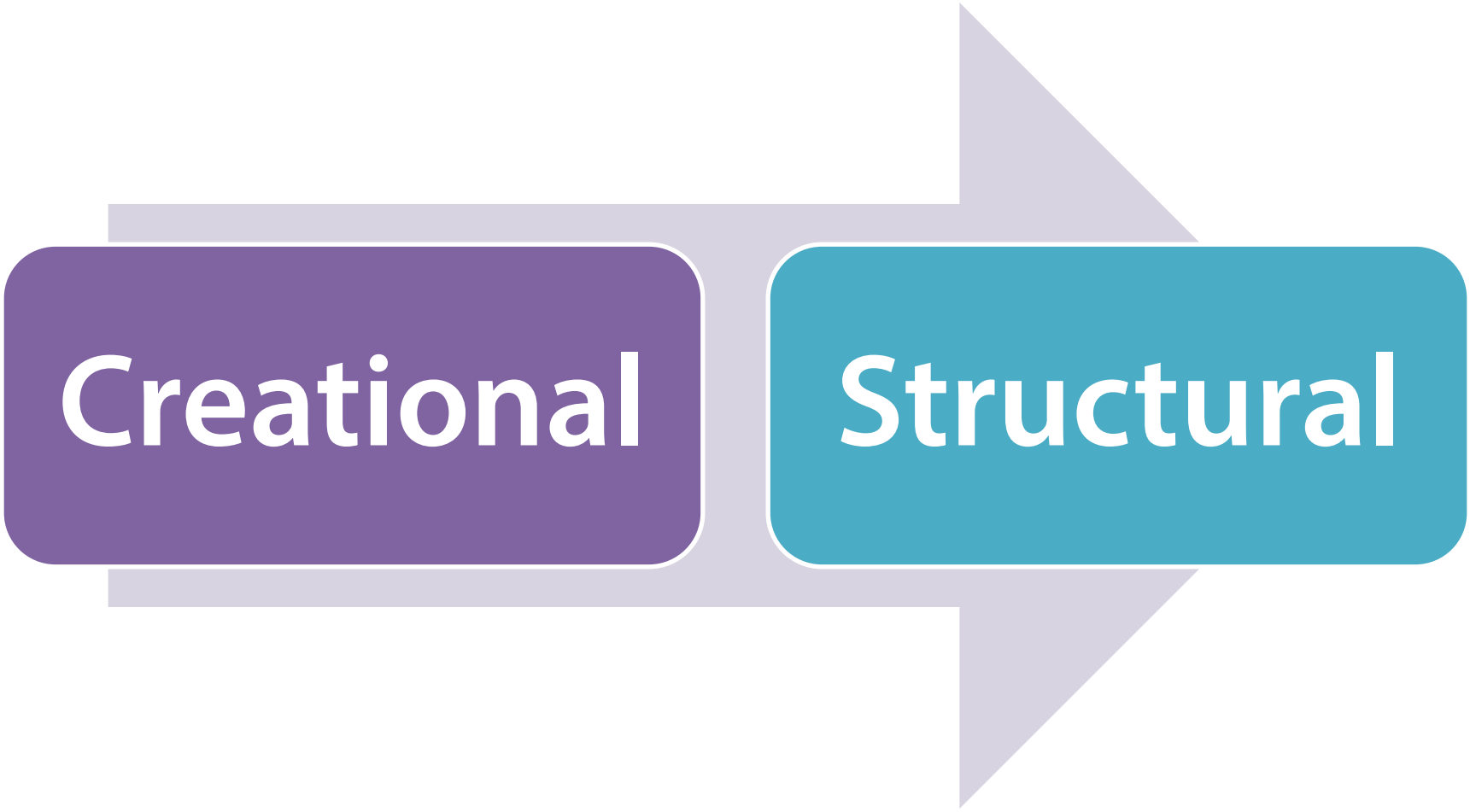
Shopping Basket Controller

http://bit.ly/Wpavw0

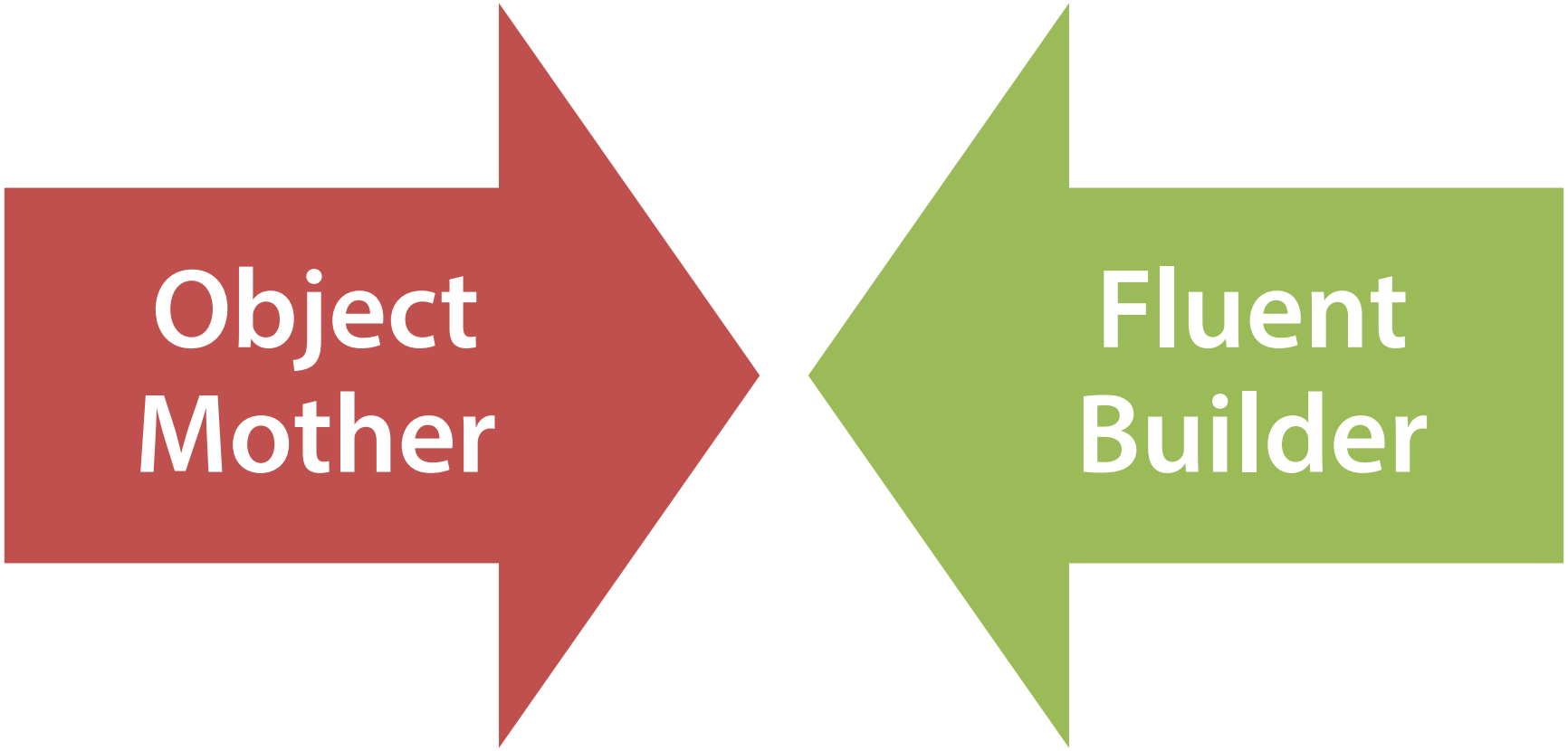
Demo recap


Evolving the SUT API breaks existing tests
First constructor change caused one test to break
Second constructor change caused two tests to break
Use of constructor is tight coupling
However, the constructor is not the subject of the test
Accidental complexity

Fixture setup patterns



Test Data Builder



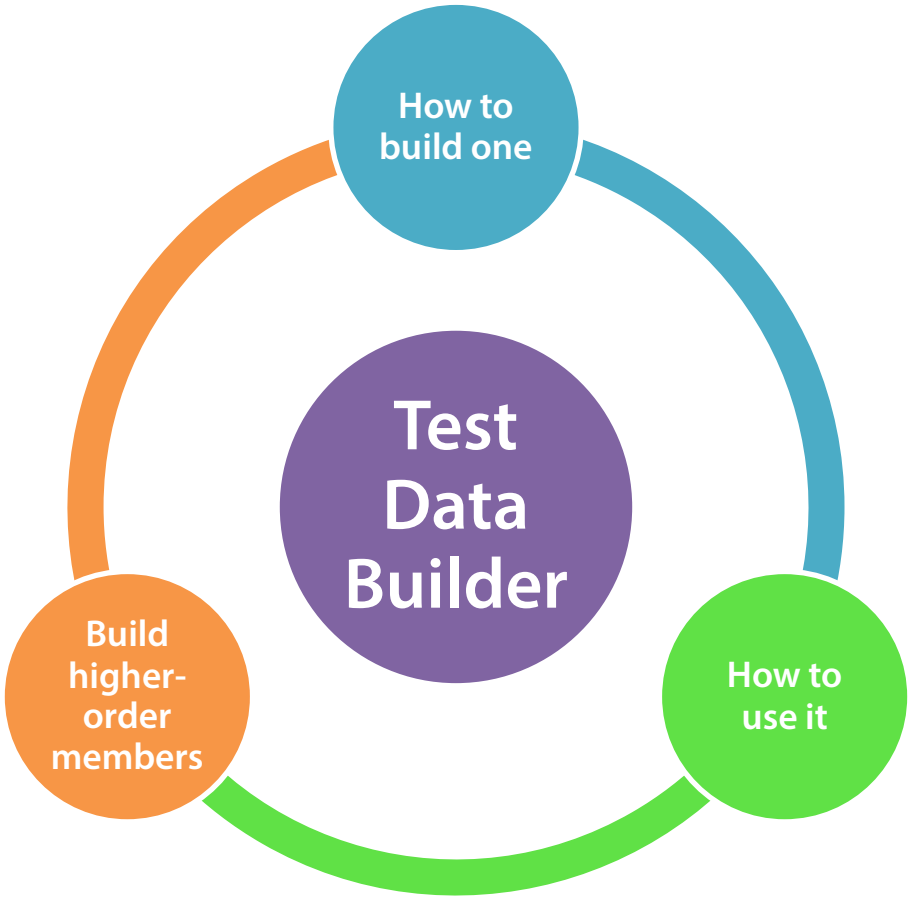
Object Mother

```
var basket = CreateWithoutDiscount();
var basket = CreateWithSmallDiscount();
var basket = CreateWithLargeDiscount();
var basket =
    CreateWithoutDiscountButWithSpecialOffer();
var basket =
    CreateWithSmallDiscountAndSpecialOffer();
var basket =
    CreateWithLargeDiscountAndSpecialOffer();
```

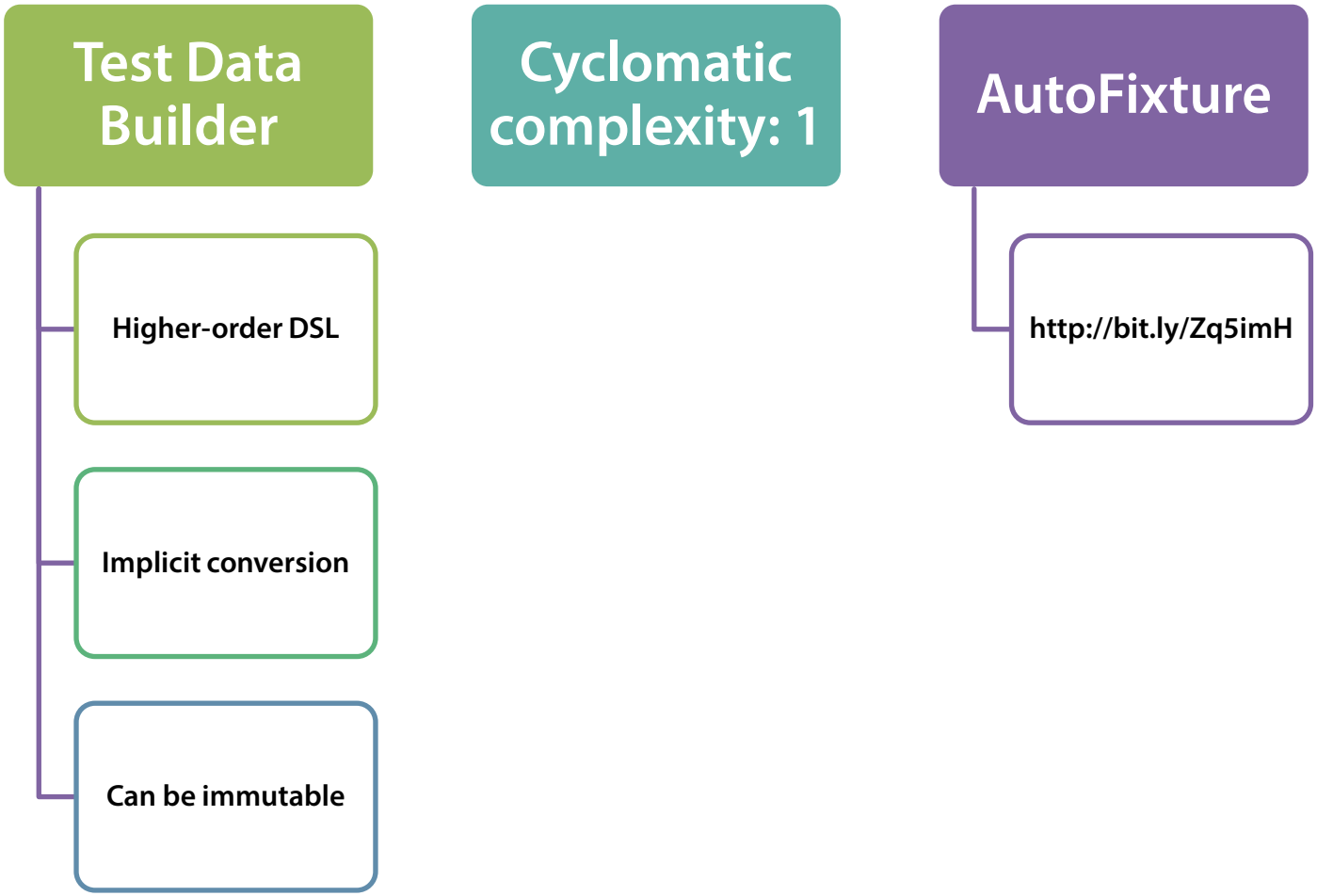
Fluent Builder

```
var basket = new BasketBuilder().Build();
var basket = new BasketBuilder()
    .WithSmallDiscount().Build();
var basket = new BasketBuilder()
    .WithLargeDiscount().Build();
var basket = new BasketBuilder()
    .WithSpecialOffer().Build();
var basket = new BasketBuilder()
    .WithSmallDiscount()
    .WithSpecialOffer().Build();
var basket = new BasketBuilder()
    .WithLargeDiscount()
    .WithSpecialOffer().Build();
```

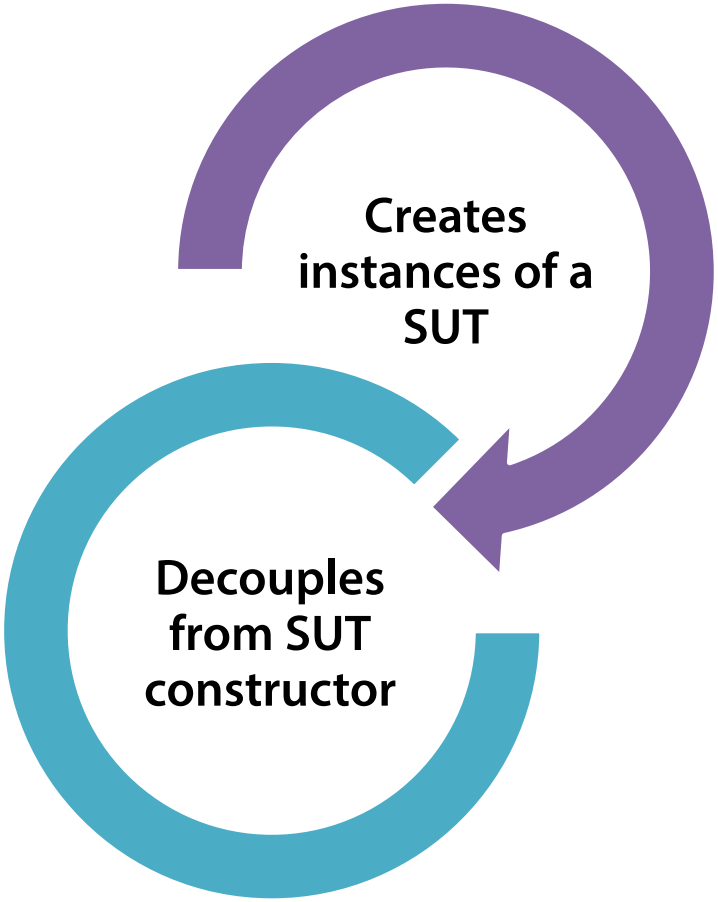
Demo


How to build one
Test Data Builder
Build higher-order members
How to use it

Demo recap


Test Data Builder
Higher-order DSL
Implicit conversion
Can be immutable
Cyclomatic complexity: 1
AutoFixture
http://bit.ly/Zq5imH

SUT Factory



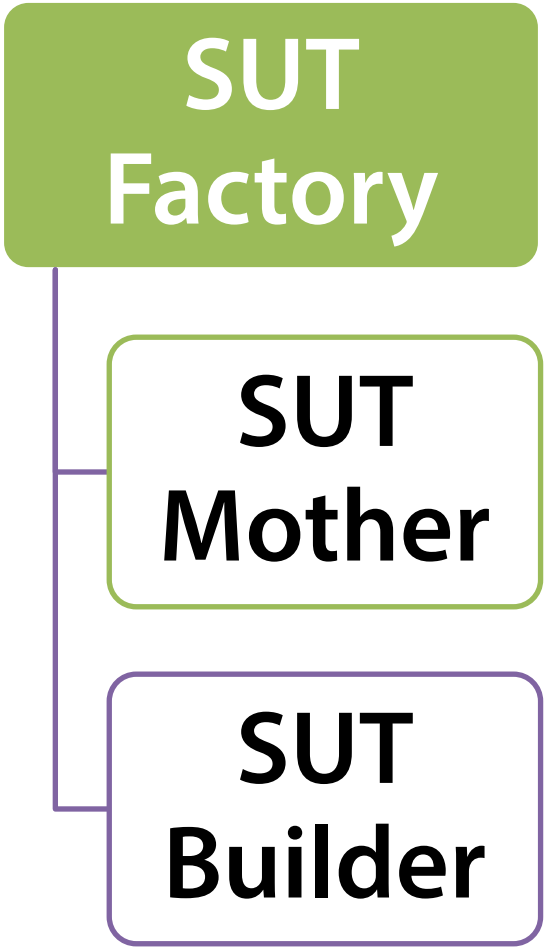
SUT Mother

```
var sut = CreateSut();

var channelMock = new Mock<IChannel>();
var sut = CreateSut(channelMock.Object);
```

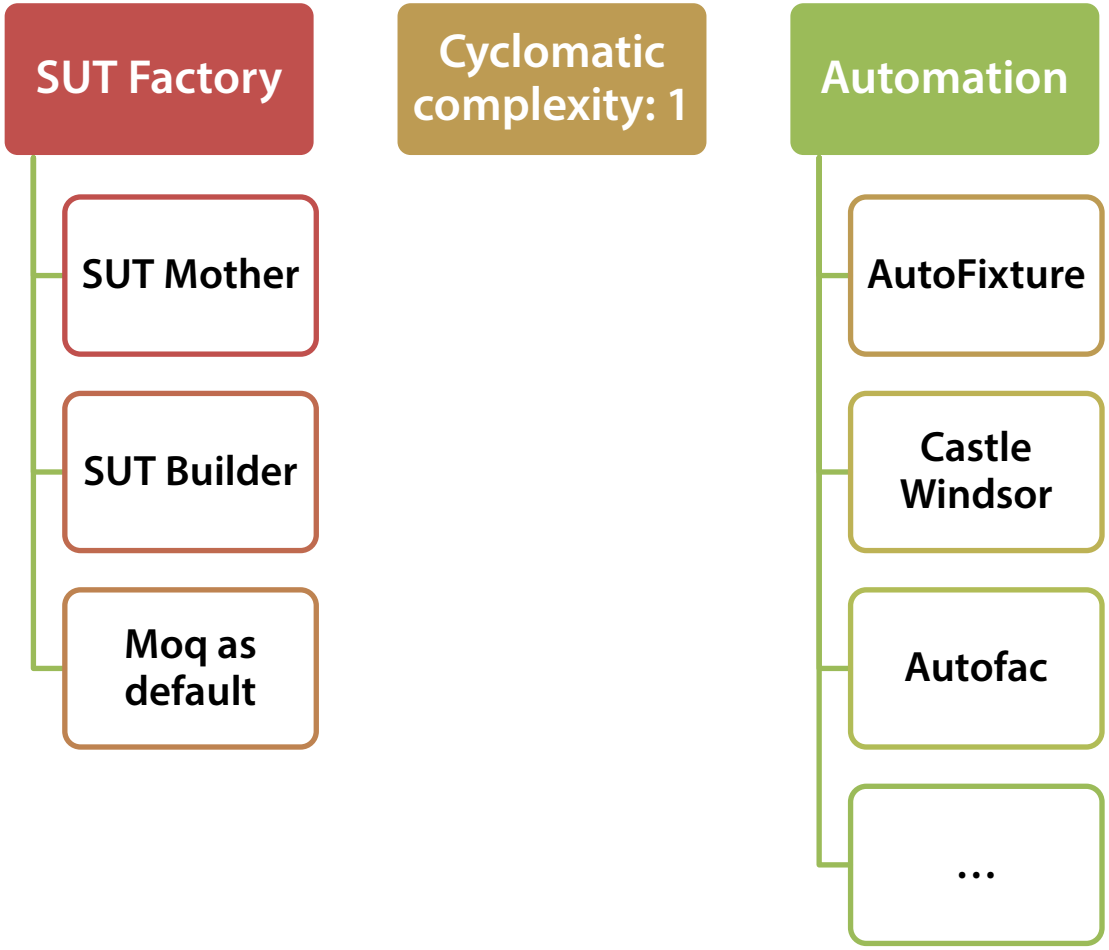
SUT Builder

```
var sut = new BasketControllerBuilder().Build();

var channelMock = new Mock<IChannel>();
var sut = new BasketControllerBuilder()
    .WithChannel(channelMock.Object).Build();
```

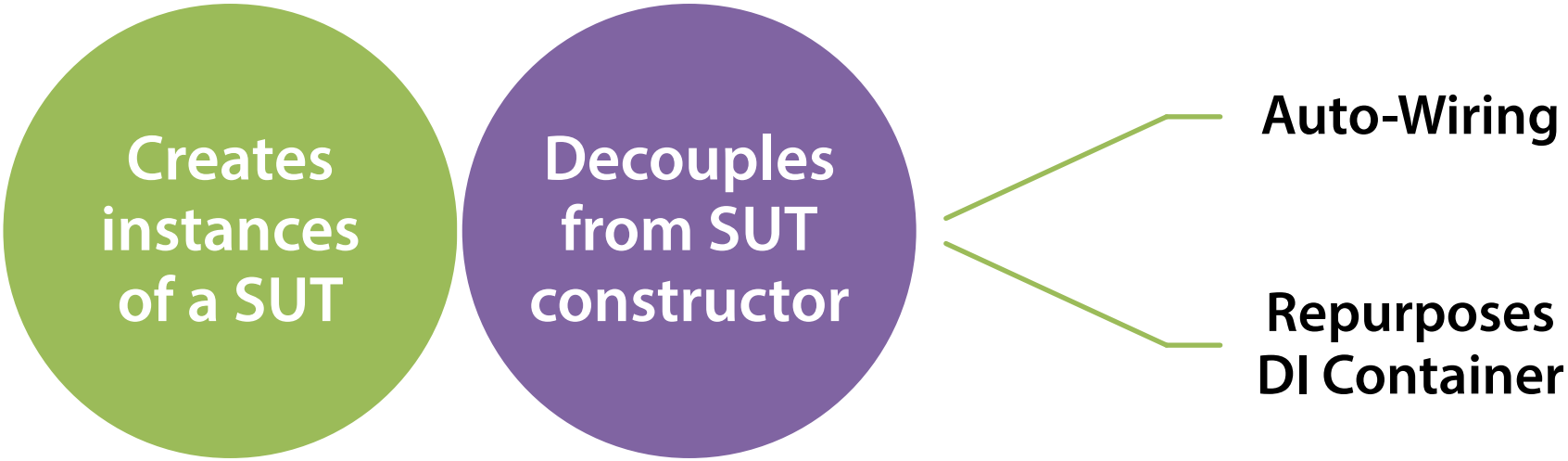
Demo



Demo recap


SUT Factory
SUT Mother
SUT Builder
Moq as default
Cyclomatic complexity: 1
Automation
AutoFixture
Castle Windsor
Autofac
...

Auto-Mocking Container



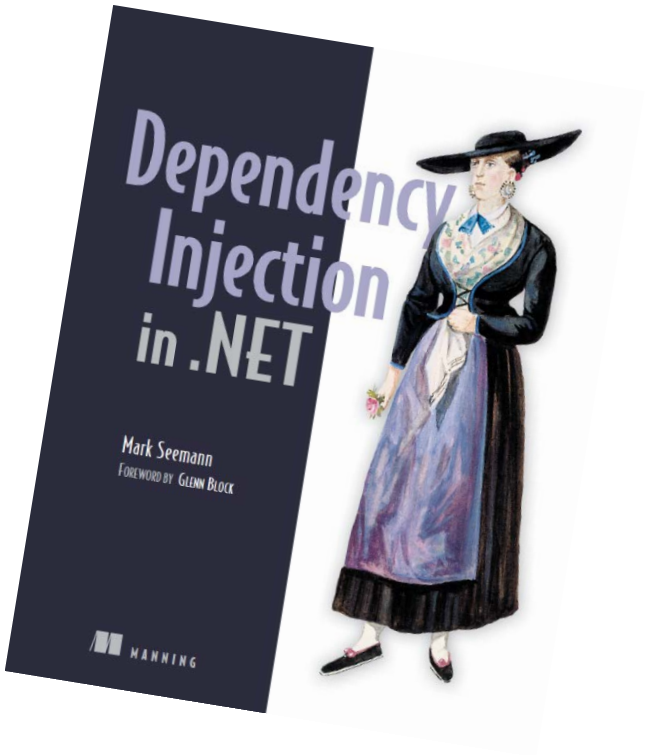
Auto-Mocking Container

```
var container = new WindsorContainer()
    .Install(new ShopFixture());
var sut = container.Resolve<BasketController>();


var channelMock =
    container.Resolve<Mock<IChannel>>();
```

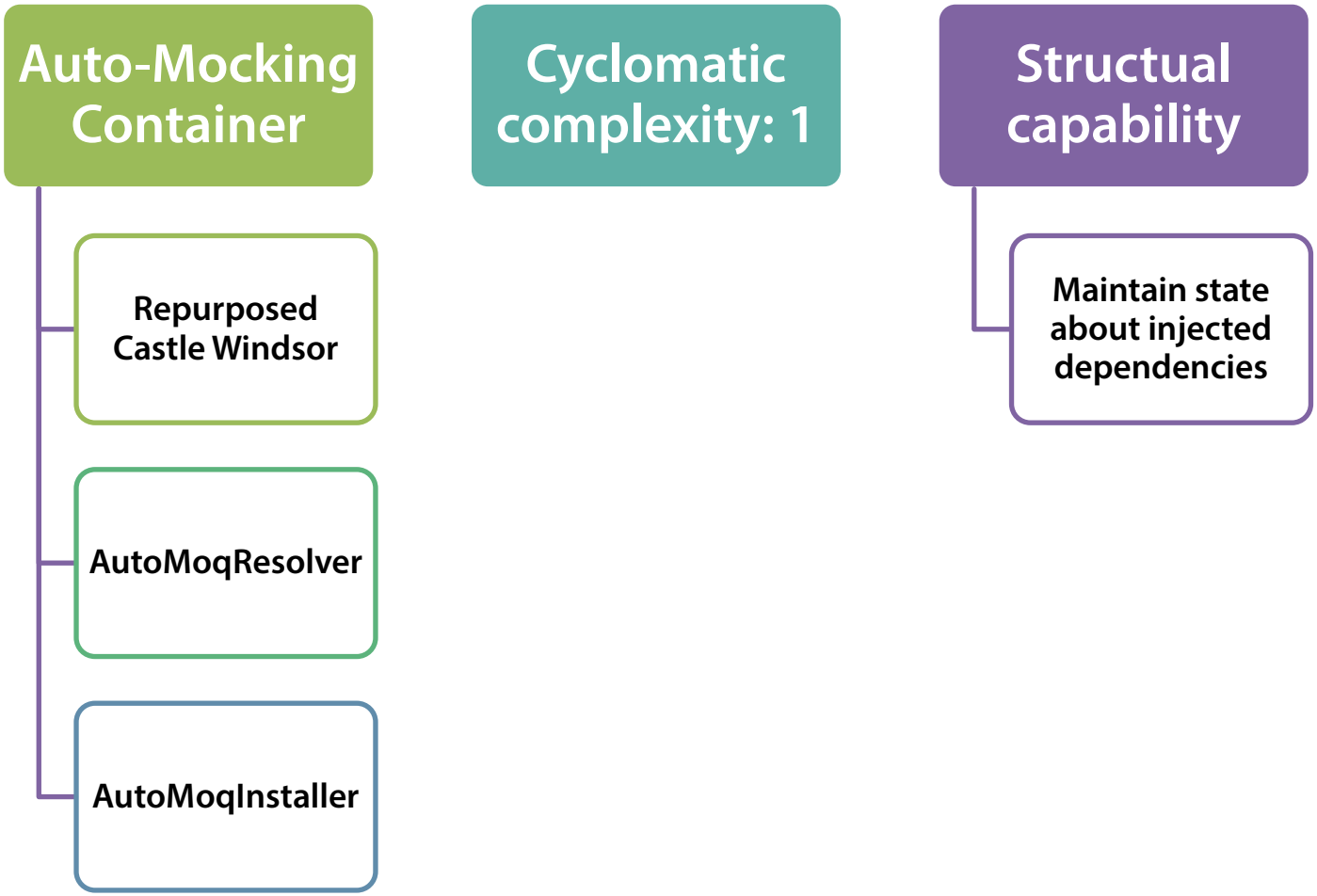
Demo

Auto-Mocking Container

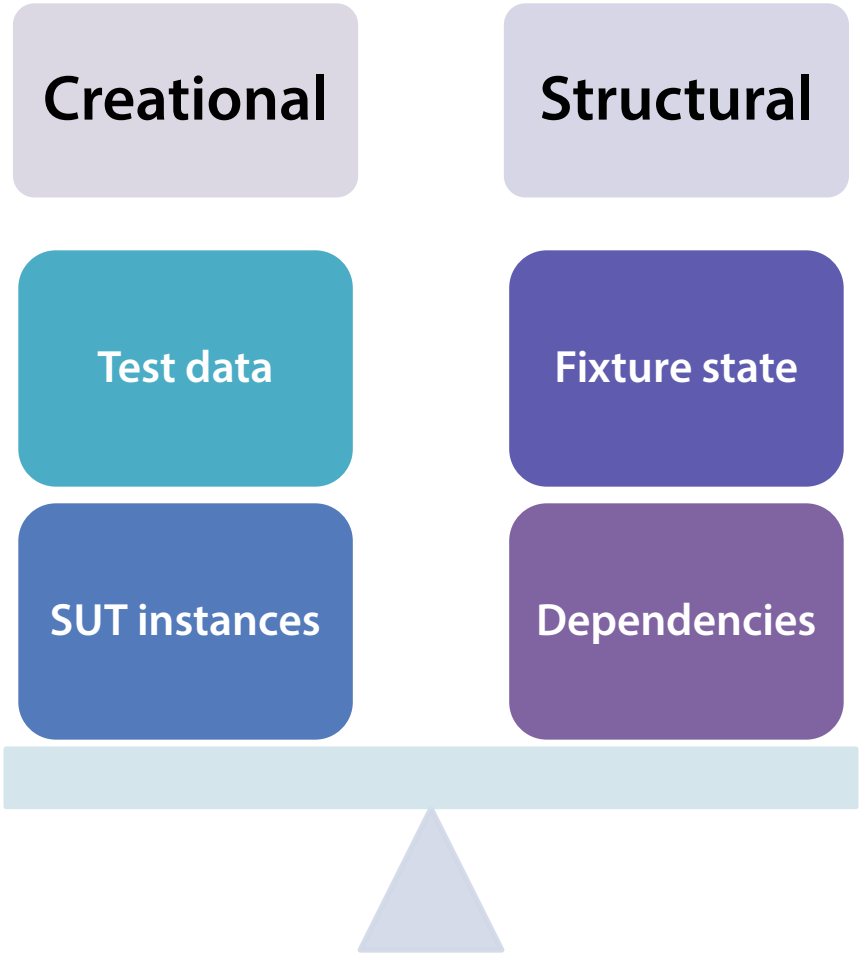
Repurposing Castle Windsor



Demo recap


Auto-Mocking Container
Repurposed Castle Windsor
AutoMoqResolver
AutoMoqInstaller
Cyclomatic complexity: 1
Structual capability
Maintain state about injected dependencies

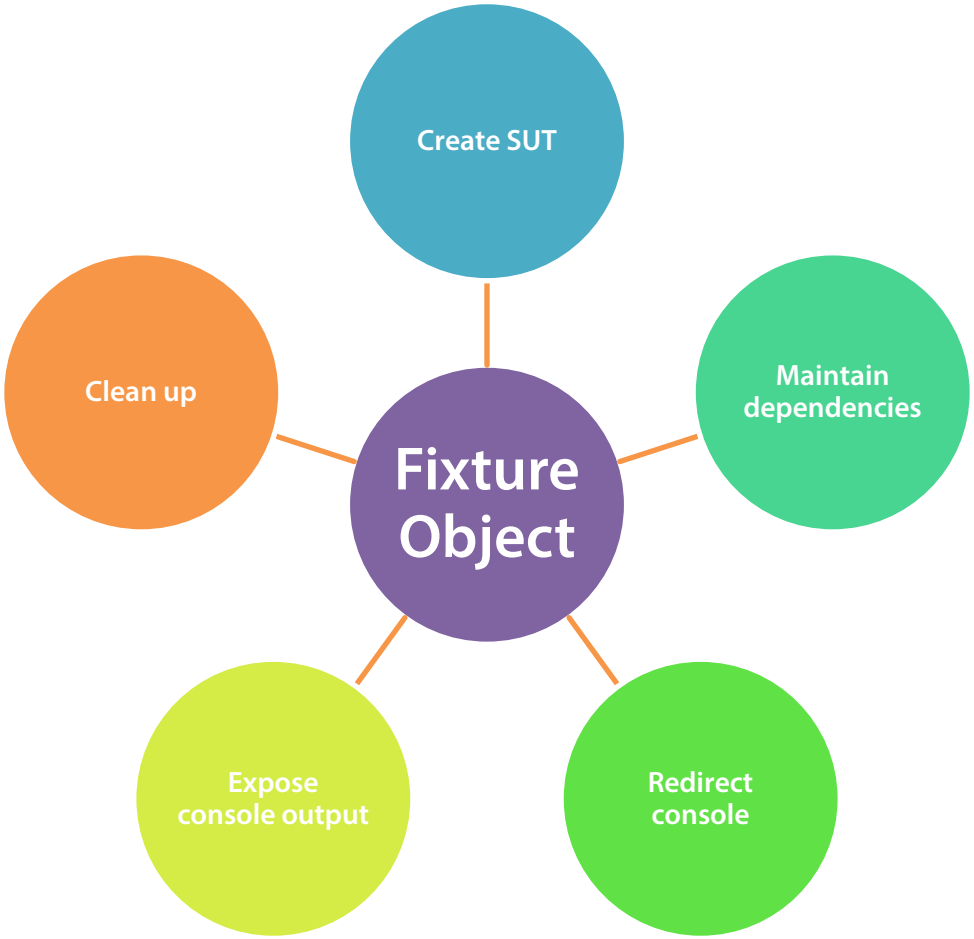
Fixture Object



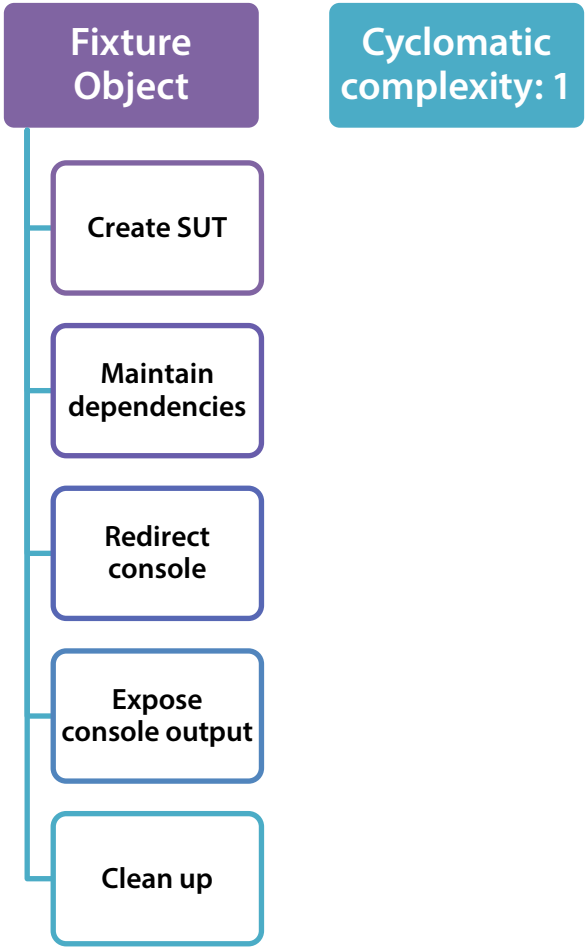
Fixture Object

```
var fixture = new BasketControllerFixture();
var sut = fixture.CreateSut();


var basket =
    fixture.Basket.WithSmallDiscount().Build();


var channelMock = fixture.channel.AsMock();
```

Demo


Create SUT
Clean up
Maintain dependencies
Fixture Object
Expose console output
Redirect console

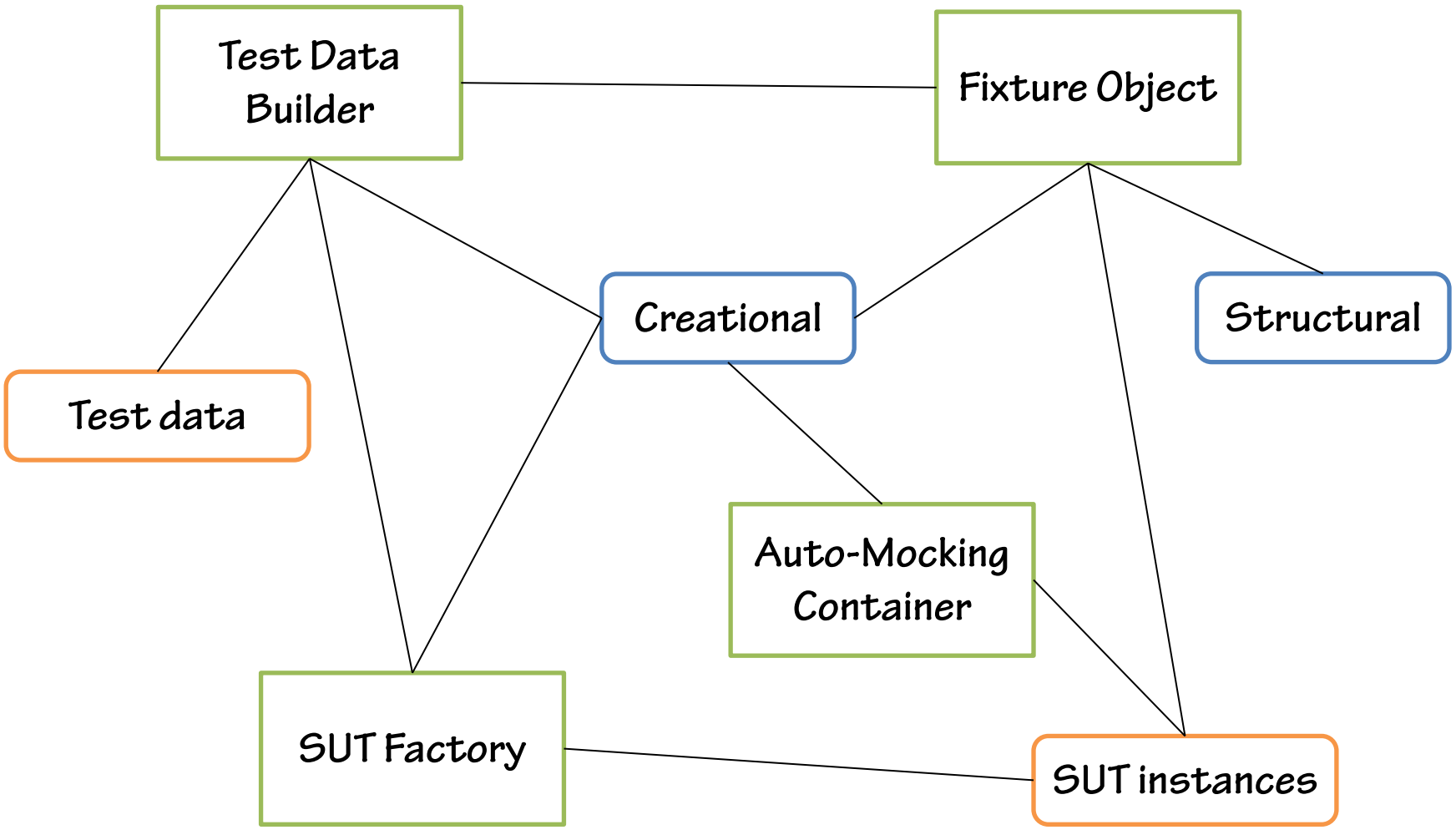
Demo recap


Fixture Object
Cyclomatic complexity: 1
Create SUT
Maintain dependencies
Redirect console
Expose console output
Clean up

Fixture Object and Test Data Builder combined

```
[Fact]
public void InvalidInputCorrectlyRendersValidationMessage()
{
    // Fixture setup
    var fixture = new SecurityControllerFixture();
    var sut = fixture.CreateSut();

    UserProfileInput invalidInput = new UserProfileInputBuilder()
        .WithNonMatchingPasswords();
    // Exercise system
    sut.CreateUser(invalidInput);
    // Verify outcome
    var expected = invalidInput.Validate();
    fixture.renderer.AsMock().Verify(r => r.Render(expected));
    // Teardown
}
```

Summary


Test Data Builder
Fixture Object
Creational
Structural
Test data
Auto-Mocking Container
SUT Factory
SUT instances