

Better .NET Unit Tests with AutoFixture: Get Started

Introducing AutoFixture



Jason Roberts

@robertsjason | dontcodetired.com

```
[Fact]
public void Test()
{
    var sut = new Calculator();
    sut.Subtract(1);
    Assert.True(sut.Value < 0);
}
```

```
[Theory, AutoData]
public void Test(int aNumber, Calculator sut)
{
    sut.Subtract(aNumber);
    Assert.True(sut.Value < 0);
}
```

Benefits of AutoFixture

Improve
Readability

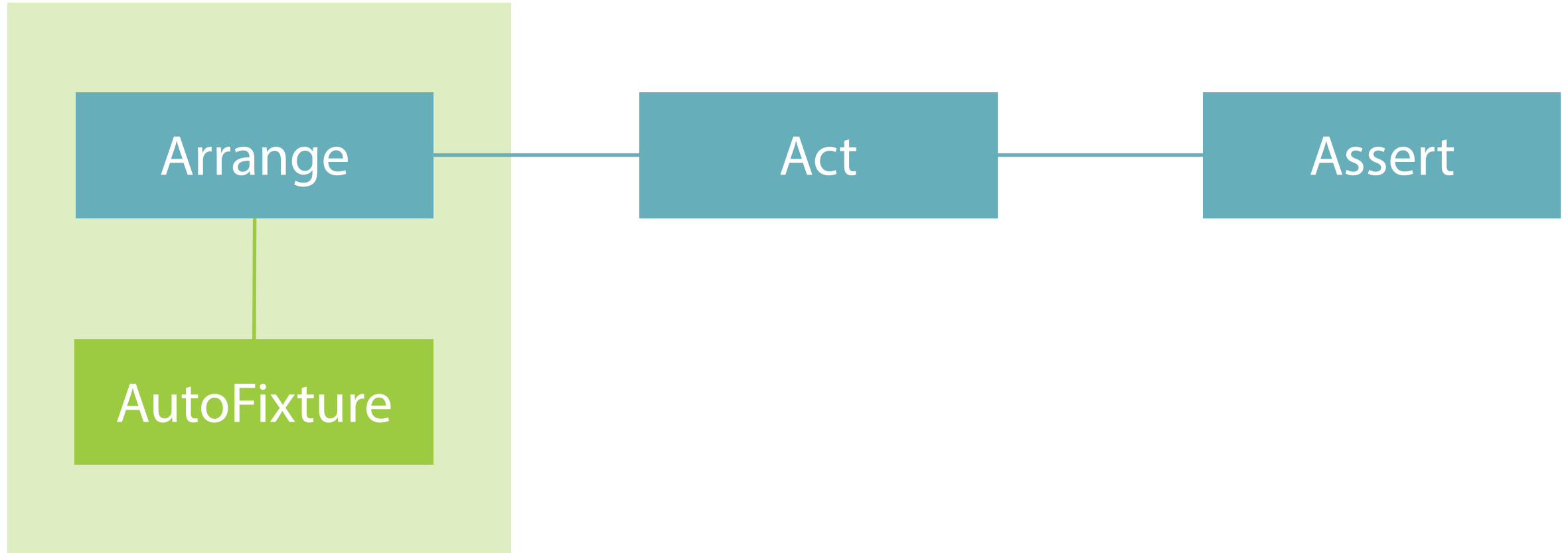
Improve
Productivity

Reduce Test
Maintenance

Reduce TDD
Friction

Reduces “Arrange” code

Arrange, Act, Assert



Anonymous Test Data

Data that is required to be present for the test to be able to execute, but where the value itself is unimportant.

```
[Fact]
public void Test()
{
    var sut = new Calculator();
    sut.Subtract(1);
    Assert.True(sut.Value < 0);
}
```

Known test
data



```
[Theory, AutoData]
public void Test(int aNumber, Calculator sut)
{
    sut.Subtract(aNumber);
    Assert.True(sut.Value < 0);
}
```

Anonymous
test data



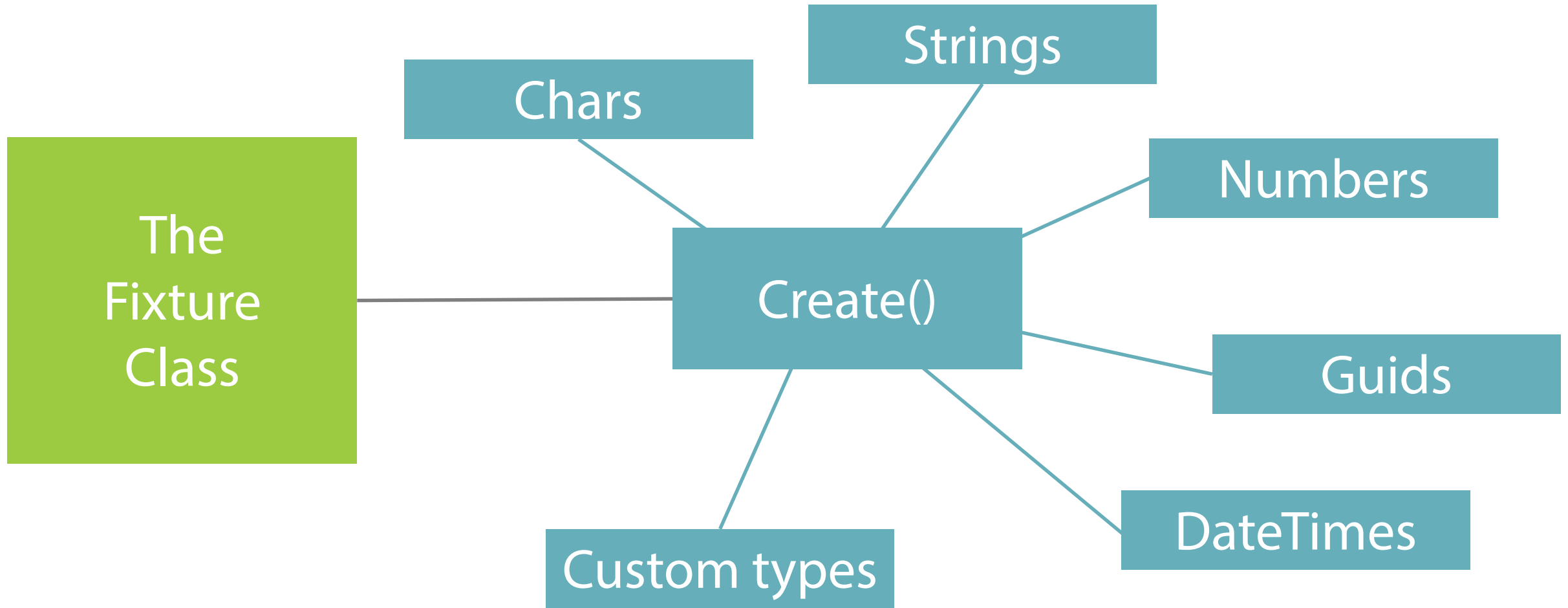
Supported Test Frameworks

NUnit

MSTest

xUnit.net

Fixie



Getting Started in Visual Studio

Create a “traditional” test

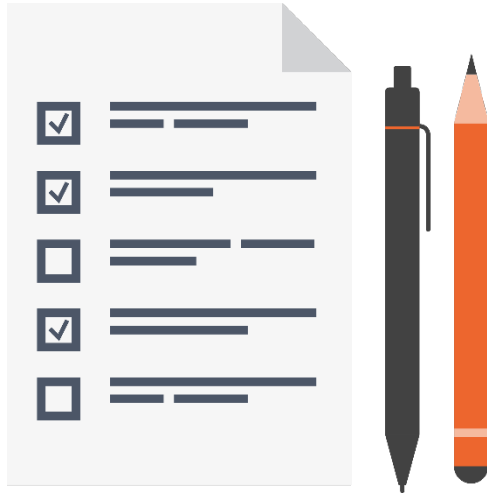
More descriptive test

Create an AutoFixture test

Using the Fixture class



Summary



Arrange phase

Anonymous test data

Fixture class

Many test frameworks

Open source NuGet package

Next: Creating Anonymous Test Data