Behavior Driven Development

Test First Development II

In This Module

- **The Essence of BDD**

- **Homemade Structure Alternatives**

- **Verification Styles**

- **Machine Specifications**

Behavior Driven Development

The Essence of BDD

Dan North

- **The natural evolution of TDD**
- **Outside in**
- **Pull based**
- **Multi-stakeholder**
- **High automation**
- **Agile**

dannorth.net/introducing-bdd

The Latest From Dan

… is a recurring topic in BDD (or **acceptance test-driven development**, **specification-by-example**, or **whatever** you choose to call the thing where you write acceptance criteria, automate them and then make the application match).
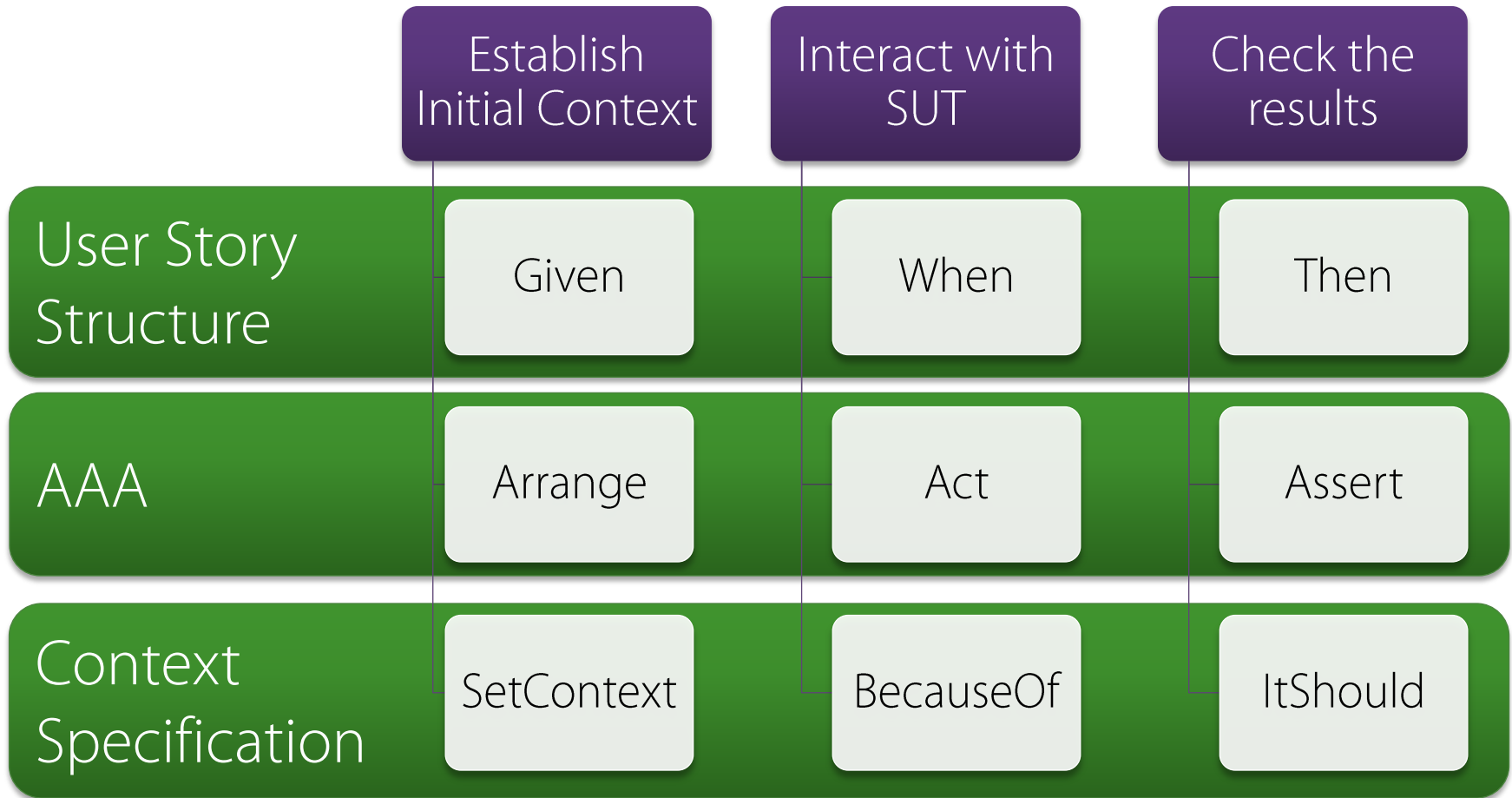
BDD Recommends

- **Test method names should be sentences**

- **A simple sentence template keeps test methods focused**

- **An expressive test name is helpful when a test fails**

- **Acceptance criteria should be executable**

- **BDD provides a "ubiquitous language" for analysis**
  - Given
  - When
  - Then

Behavior Driven Development

Homemade Structure Alternatives

Variations on a Theme

	Establish Initial Context	Interact with SUT	Check the results
User Story Structure	Given	When	Then
AAA	Arrange	Act	Assert
Context Specification	SetContext	BecauseOf	ItShould

pluralsight
see what you can learn

Behavior Driven Development

Machine Specifications

An MSpec Test

```csharp
public class When_adding_5_and_1
{
    static int x;

    Establish context = () => x = 5;
    Because of = () => x = x + 1;
    It Should_be_6 = () => x.ShouldEqual(6);

}
```

When adding 5 and 1 *(1 test)*
    Should be 6

Behavior Driven Development

Testing Styles

Inside Out	Outside In
▪ **Start with a low level part of the system**	▪ **Start with a high level abstraction**
▪ **Assemble final SUT from tested components**	▪ **Evolve the final SUT as components are added**
▪ **Ensures only one test may fail at a time for true TDD**	▪ **More than one test may be failing at a time**

State	Interaction
▪ **Focuses on results of SUT invocation**	▪ **Focuses on behavioral details of SUT invocation**
▪ **Coupled to the external interface of the SUT**	▪ **Coupled to the method implementation**
▪ **Verify what was returned**	▪ **Verify what the SUT did**

www.martinfowler.com/articles/mocksArentStubs.html
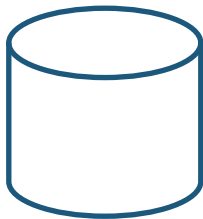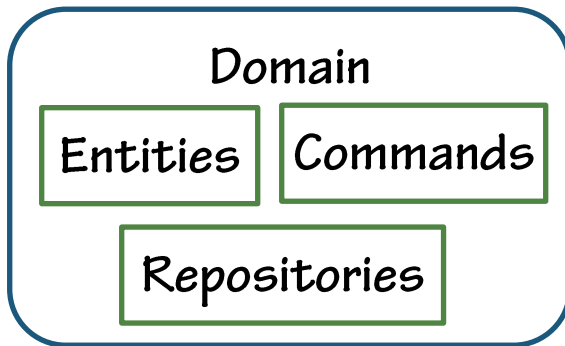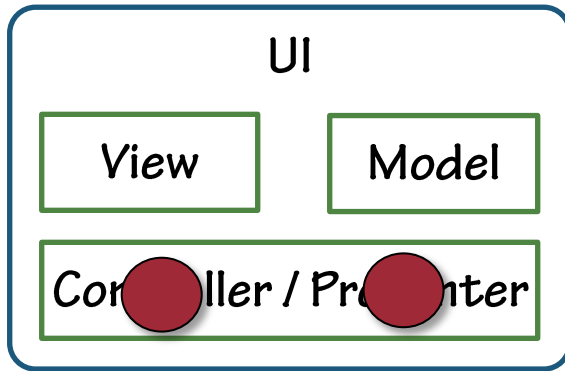
Testability and Entity Framework 4.0

by Scott Allen

http://bit.ly/9ogHkH

- **Reuse the Test implementations**

- **Forces separation of concerns between classes**

- **We are able to isolate the SUT completely for testing**

Summary

- **The Essence of BDD**

- **Homemade Structure Alternatives**

- **Verification Styles**

- **Machine Specifications**

References

- **Mocks Aren't Stubs**
  www.martinfowler.com/articles/mocksArentStubs.html

- **Testability and Entity Framework 4.0**
  http://bit.ly/9ogHkH