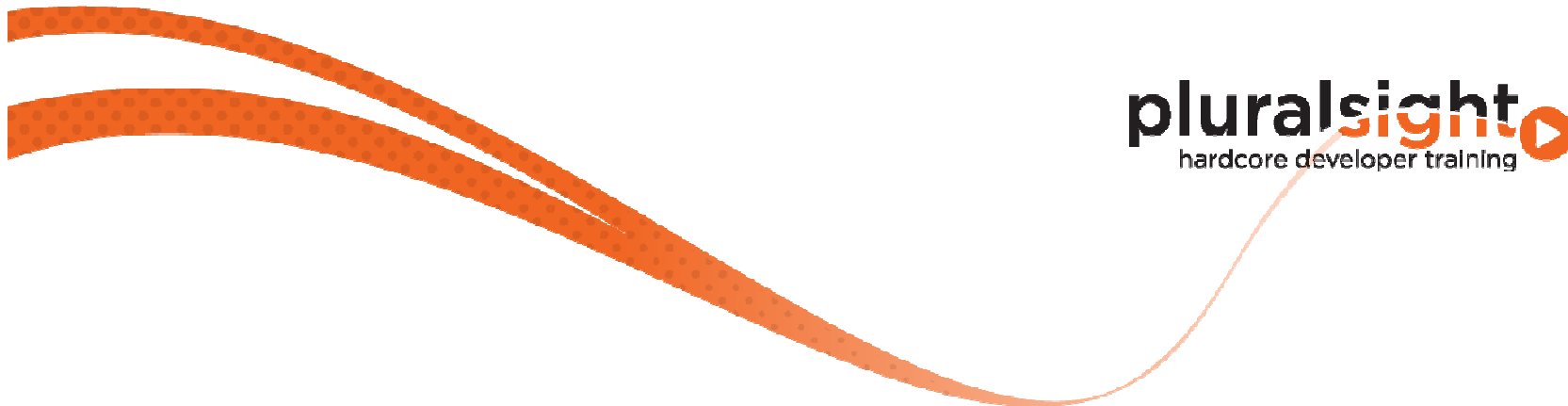


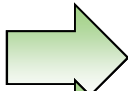


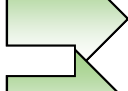

Advanced Mocking with Moq

Donald Belcham

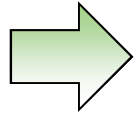
@dbelcham



Outline

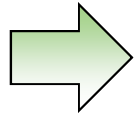
-  **Strict and Loose Mocking**
-  **Base Class Implementations**
-  **Recursive Mocking**
-  **Centralized Mock Creation**
-  **Protected Members**

Strict vs Loose Mocking



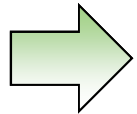
Strict:

- raises exception for anything on a mock object that doesn't have an explicitly declared expectation



Loose:

- no exception raised
- returns default values when no expectation is explicitly declared

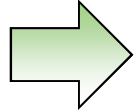


Loose is the Moq default behavior

Base Class Implementations

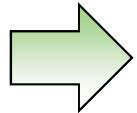
- ➡ aka "Partial Mocks"
- ➡ Allows for invocation of base class implementation if no expectation has been set on the member
- ➡ Necessary for mocking Web/Html controls in System.Web

Recursive Mocking



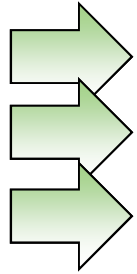
A mock object that returns a new mock object for every member

- If the member is mockable (i.e. not a value type)
- If there is no expectation set on the member



The same mock object is returned on all calls

Centralizing Mock Creation



MockRepository

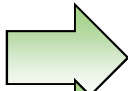

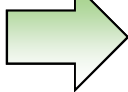
Configure all mock objects centrally

Verify all mock objects in one call

Protected Members

- ➡ Mocking members that use the 'protected' keyword
- ➡ No intellisense
- ➡ Must use 'ItExpr' instead of 'It'

Summary

-  **Complex scenarios are possible**
-  **The defaults of Moq allow for easy adoption**
-  **Complex features require configuration in tests**