Your First Unit Tests

Paul D. Sheriff
Business/IT Consultant PDS Consulting

psheriff@pdsa.com www.pdsa.com

Module Overview

- Create class/method to test
- Create unit test project
- Write unit test stubs
- Write unit tests
- Run tests and see results

Getting Started

Getting Started

Class library project

FileProcess class
- FileExists()

Unit test project

FileProcessTest class
- FileNameDoesExist()
- FileNameDoesNotExist()
- FileNameNullOrEmpty_ ThrowsArgumentNullExcepti on

```
public bool FileExists(string fileName) {
    if (string.IsNullOrEmpty(fileName)) {
        throw new
          ArgumentNullException("fileName");
    }
    return File.Exists(fileName);
}
```

- FileExists method
- Check for valid parameter
- Throws exception if nothing passed in
- Return true/false based on file existence

Getting Started

Step 1	Step 2
Try to think of as many tests as possible	Create test for each method
Write method stubs for each test	Create test for each overload of a method
Assert.Inconclusive() indicates no test has been written	Make sure write tests to cover all if, else, switch, loop statements

Getting Started

Step 3

Too many test methods?

Refactor original method

Step 4

Run the tests

See all inconclusive results

Getting Started

Step 5

Write tests

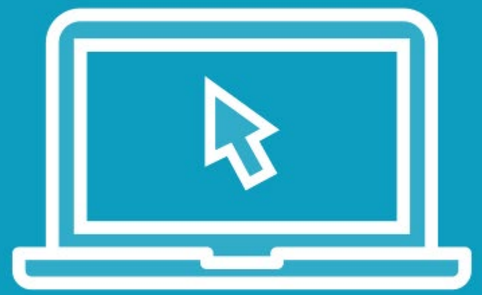
Refactor original method as needed

Repeat until all test methods are working

```
[TestMethod]
public void FileNameDoesExist() {
  Assert.Inconclusive();
}

[TestMethod]
public void FileNameDoesNotExist() {
  Assert.Inconclusive();
}

[TestMethod]
public void FileNameNullOrEmpty
       _ThrowsArgumentNullException() {
  Assert.Inconclusive();
}
```

- Create 3 method stubs
- Test for good file name
- Test for bad file name
- Test for no file name passed

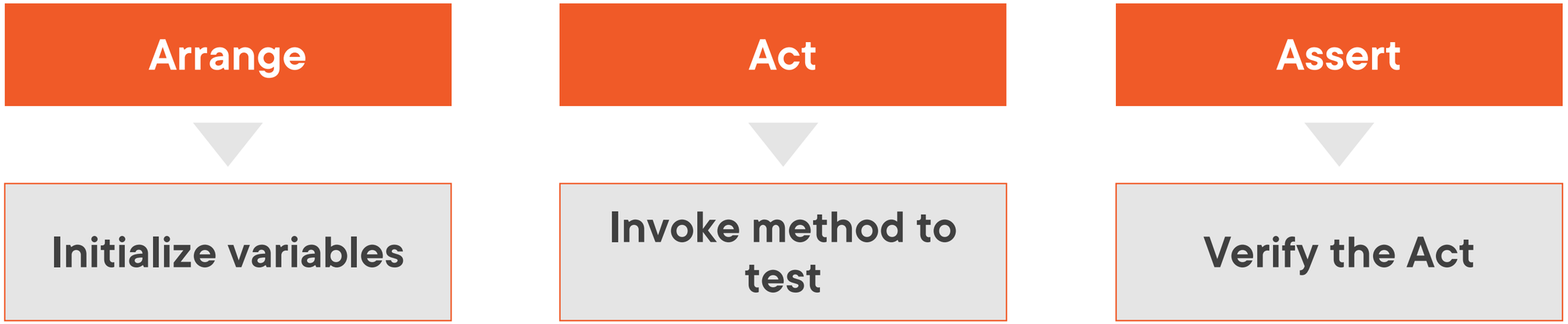
Demo



- Create class library project
- Create unit test project
- Write stubs

Write Tests

Use AAA


Arrange
Initialize variables
Act
Invoke method to test
Assert
Verify the Act

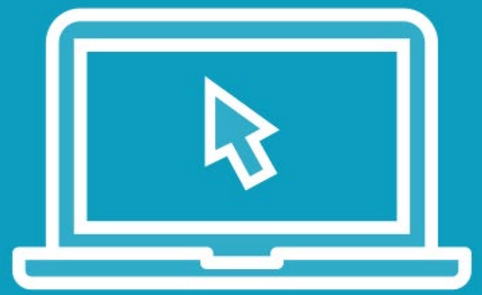
```
[TestMethod]
public void FileNameDoesExist() {
  FileProcess fp = new FileProcess();
  bool fromCall;

  fromCall =
    fp.FileExists(@“GoodFile.txt");

  Assert.IsTrue(fromCall);
}
```

◀ Arrange

◀ Act

◀ Assert

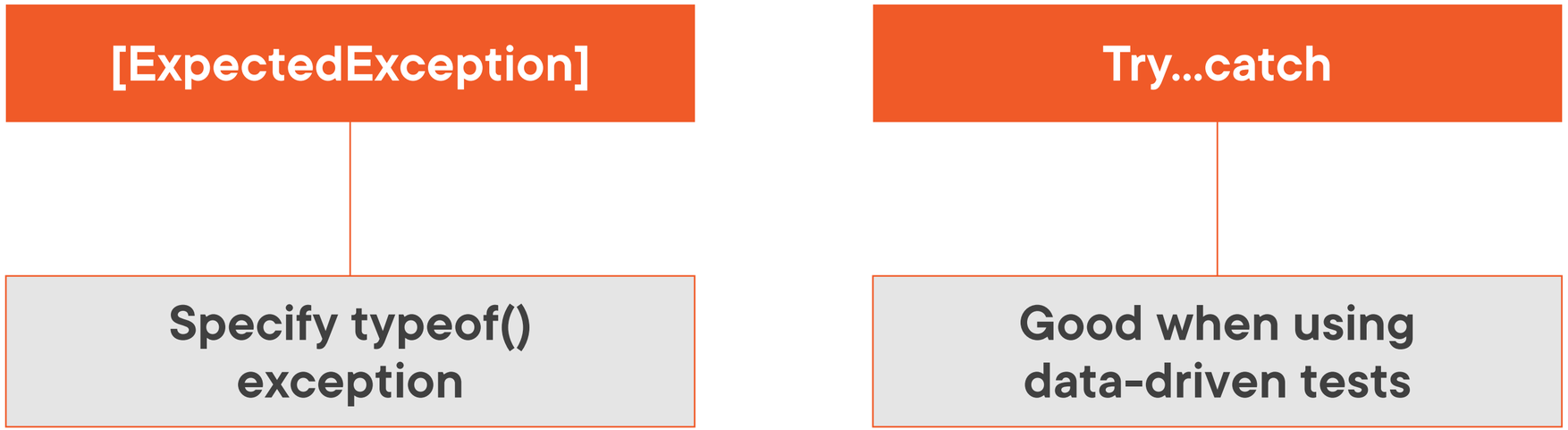
Demo



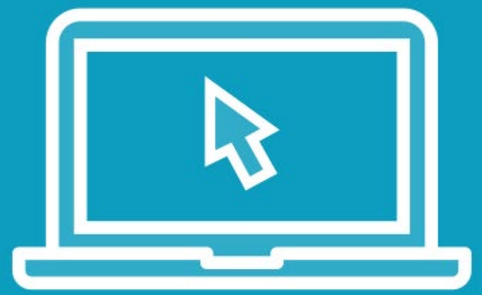
- **Write tests**

Handling Exceptions

Exception Handling



Demo



- **Exception handling**

Code Coverage

Code Coverage

Have you covered all code with test(s)?

Is all code tested?

Helps you determine what else you need to test

Tool built into Visual Studio

Enterprise Edition only

Module Summary

- Created unit test project
- Wrote test stubs
- Wrote tests using AAA
- Handle exceptions
- Code coverage

Up Next:
Avoid Hard-Coding in Unit Tests