Automated Testing with MSTest V2

GETTING STARTED

Jason Roberts
.NET MVP

@robertsjason   dontcodetired.com

Overview

The benefits of automated tests

An overview of MSTest
- High level components
- NuGet packages
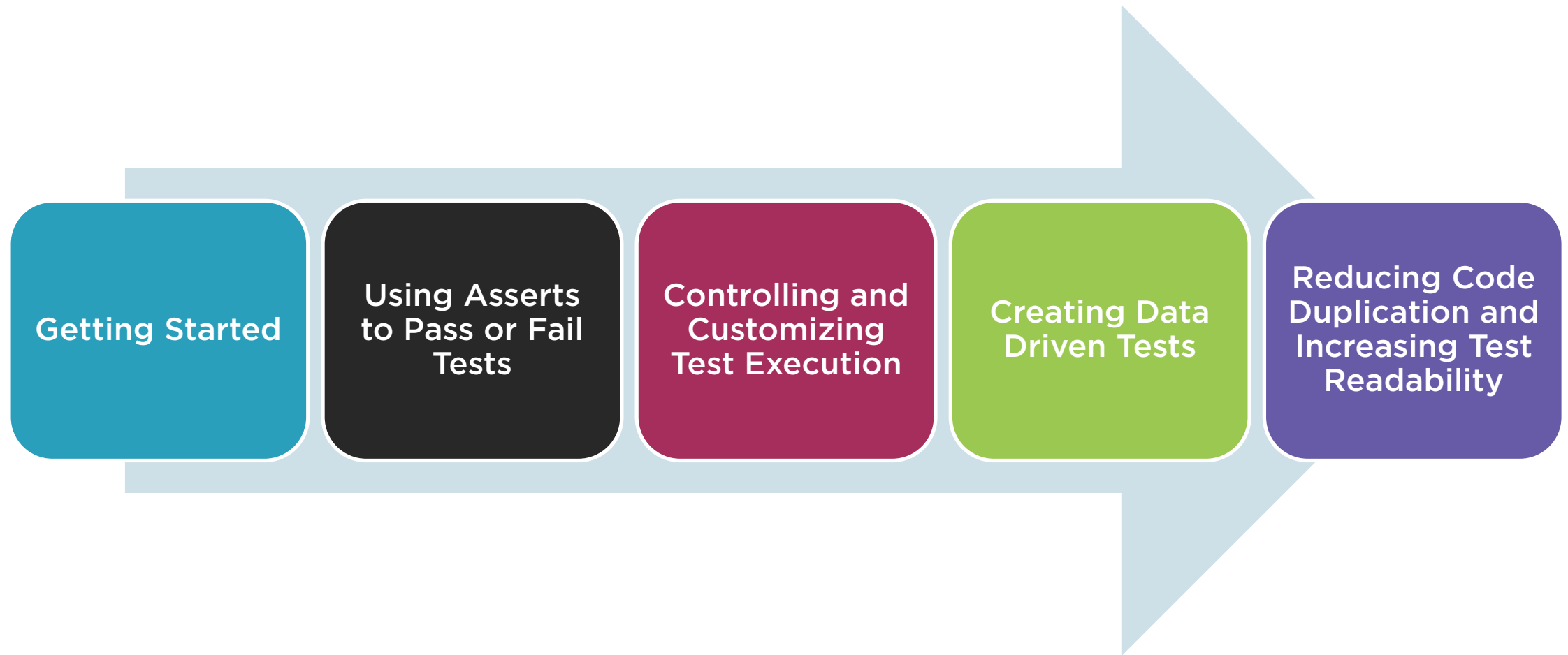- Features

Creating an MSTest test project

Writing an initial test

Running tests in Visual Studio

Running tests at the command line

Course Outline

Getting Started

Using Asserts to Pass or Fail Tests

Controlling and Customizing Test Execution

Creating Data Driven Tests

Reducing Code Duplication and Increasing Test Readability

The Benefits of Automated Tests

Find (and fix) errors sooner

Free to execute as often as required

Quick to execute (versus manual testing)

Generally more repeatable (versus manual)

Execution flexibility:
- Run on demand
- As part of continuous integration
- On a schedule (e.g. overnight)

Happier:
- Developers & teams
- Happier end users
- Happier business owners

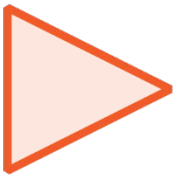# Testing Automation: The Big Picture

An Overview of MSTest

Attributes

Asserts

Test runners

NuGet Packages

MSTest.TestFramework

Test framework
Attributes
Assert methods
Data driven tests
Extensibility points

MSTest.TestAdapter

Discover MSTest-based tests
[TestMethod] etc.
Execute tests

Feature Overview

Asserts
- Basic asserts
- Additional string asserts
- Additional collection asserts

Test categories

Test execution lifecycle hooks

Data driven tests

Extensibility
- Custom assert methods
- Custom attributes
- Custom test data sources

Summary

The benefits of automated tests

An overview of MSTest
- Attributes, asserts, and runners
- NuGet packages
- Test execution lifecycle hooks

Created an MSTest test project

Wrote an initial test
- [TestClass]
- [TestMethod]

Run tests in Visual Studio Test Explorer

Run tests at the command line

Next:

Using Asserts to Pass or Fail Tests