Business Readable Automated Tests with SpecFlow 2.0

WHY USE SPECFLOW?

Jason Roberts
.NET MVP

@robertsjason dontcodetired.com

Overview

Why use SpecFlow?

Benefits of business readable automated tests

“Build the right thing and build it right”

Types of tests that can be written with SpecFlow

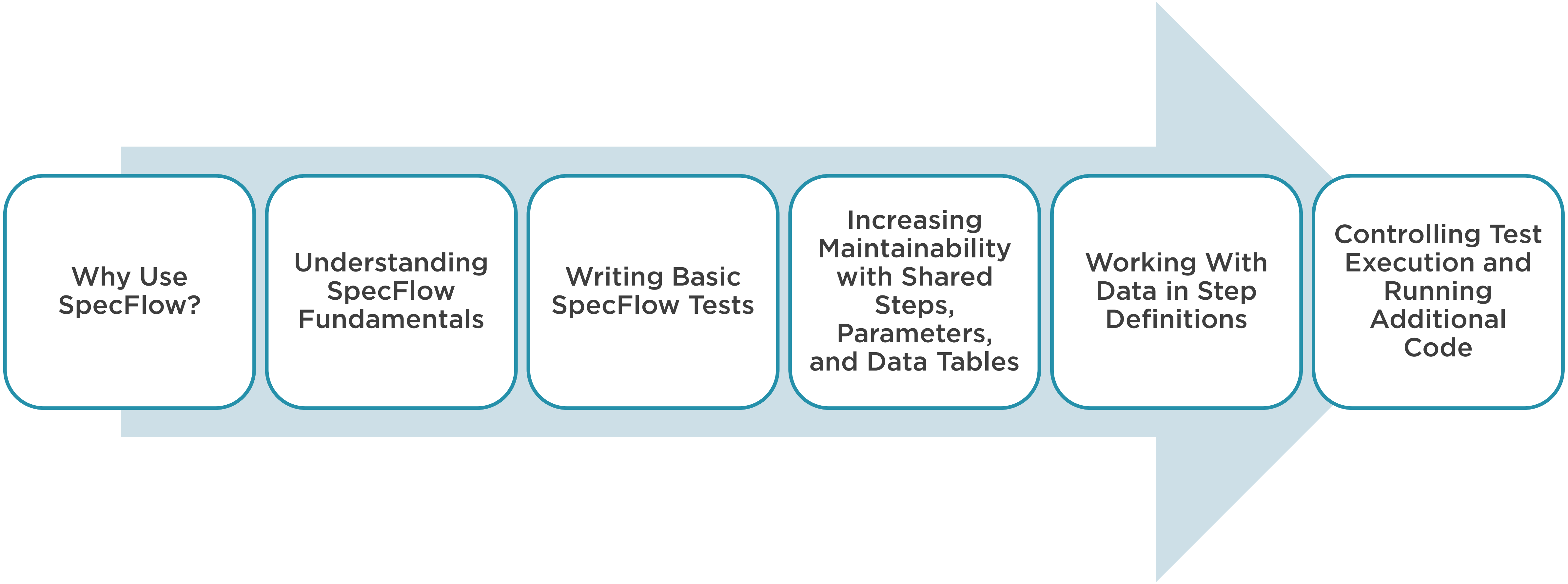
Possible SpecFlow workflows

Living documentation

High level overview of how SpecFlow works

SpecFlow technical features

Course Outline



Why Use SpecFlow?

Understanding SpecFlow Fundamentals

Writing Basic SpecFlow Tests

Increasing Maintainability with Shared Steps, Parameters, and Data Tables

Working With Data in Step Definitions

Controlling Test Execution and Running Additional Code

Business readable automated tests

Tests that run automatically to verify a system is working as expected, and that document the system in a way that non-technical people can understand and contribute to.

Why Business Readable Automated Tests?

Document

Execute

Communicate

Help document:

- What features does the system have?
- How do those features work?
- What different usage scenarios does the system support?
- What does the system not support?

Lives alongside source code

New developers

New business people

Audit

Executable

Checks the system is behaving as expected

Documentation is “living”

Stays in sync with actual code

Business readable documentation maps to executable test code

Enables better communication between development team and business

Common (high-level) language

Generally non-technical

Ensure correct features are being built

Ensure different usage scenarios are covered

> “Build the right thing and build it right.”

Types of Automated Tests

Unit

Integration

API

Functional UI

SpecFlow can sit on top of any type of automation test code you are currently writing

SpecFlow Workflows

BDD

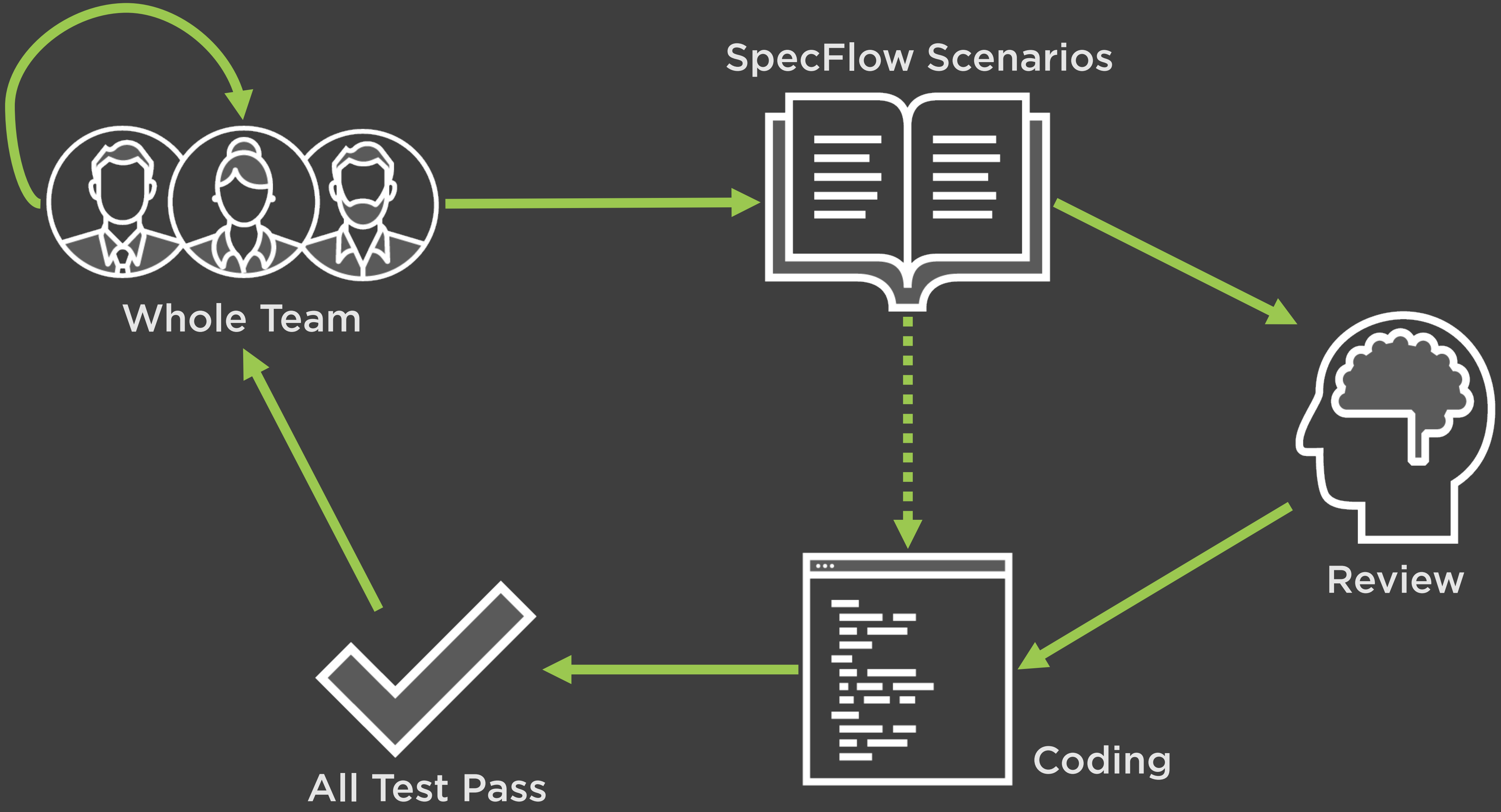
ATDD

Add SpecFlow tests to existing code

Not “all-or-nothing” approach

Start with communication “gaps”

Example Workflow



Source code exists in source control system

Truth of what the system does

Documentation in separate place

Word documents on network share

Documents in SharePoint

Uncertainty or ambiguity in requirements or specifications can lead to wasted time and effort of business people, developers and testers and may impact team happiness in a negative way.

Source code exists in source control system

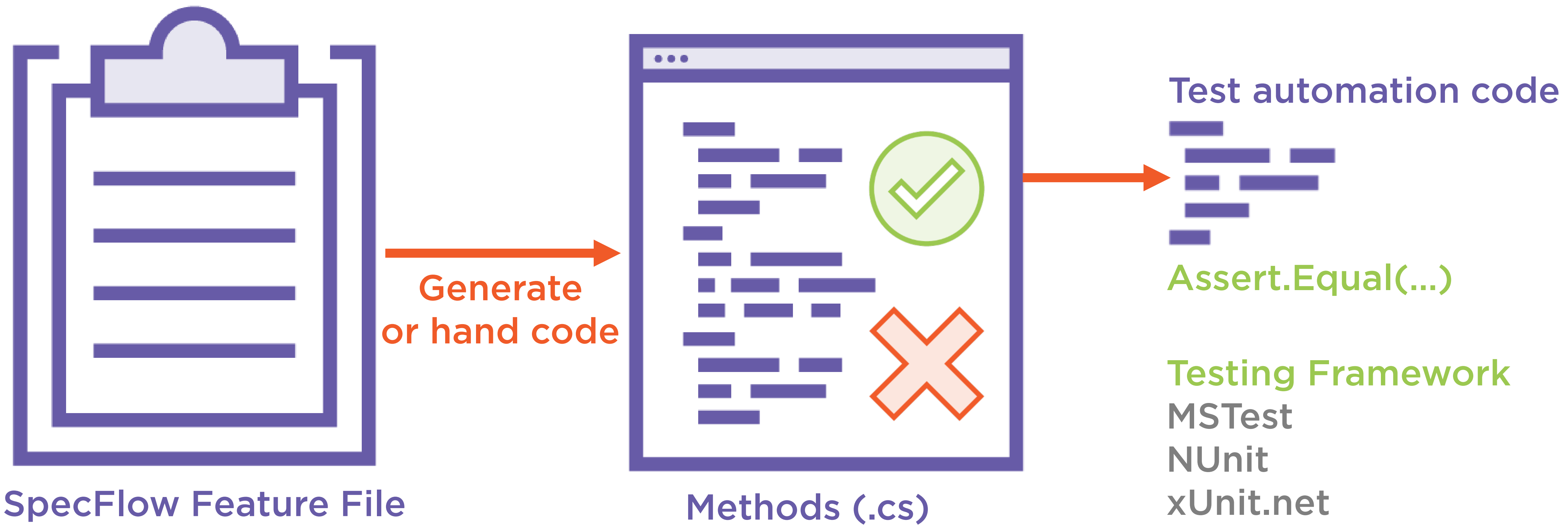
Specifications exist alongside source code

Single truth of what the system does

Specifications (tests) can be executed

May still have supporting documents

High Level SpecFlow Overview



Testing Framework

MSTest
NUnit
xUnit.net

SpecFlow Technical Features Overview

Organize scenarios into feature files

Support for many different spoken languages

Map automation code by method name convention or regular expressions

Share code steps between scenarios

Add arbitrary tags to organize and execute

Create reusable steps that use parameters

Replace repetitious scenarios with tables

Customizable step parameter data conversion

Use dynamic types to reduce code

Run additional setup/clean-up code

Summary

Document, Execute, Communicate

“Build the right thing and build it right”

Unit, integration, API, UI

ATTD, BDD, retrofit to existing code

Living documentation

High level overview of how SpecFlow works (feature file → step methods)

SpecFlow technical features

Next:

Understanding SpecFlow Fundamentals