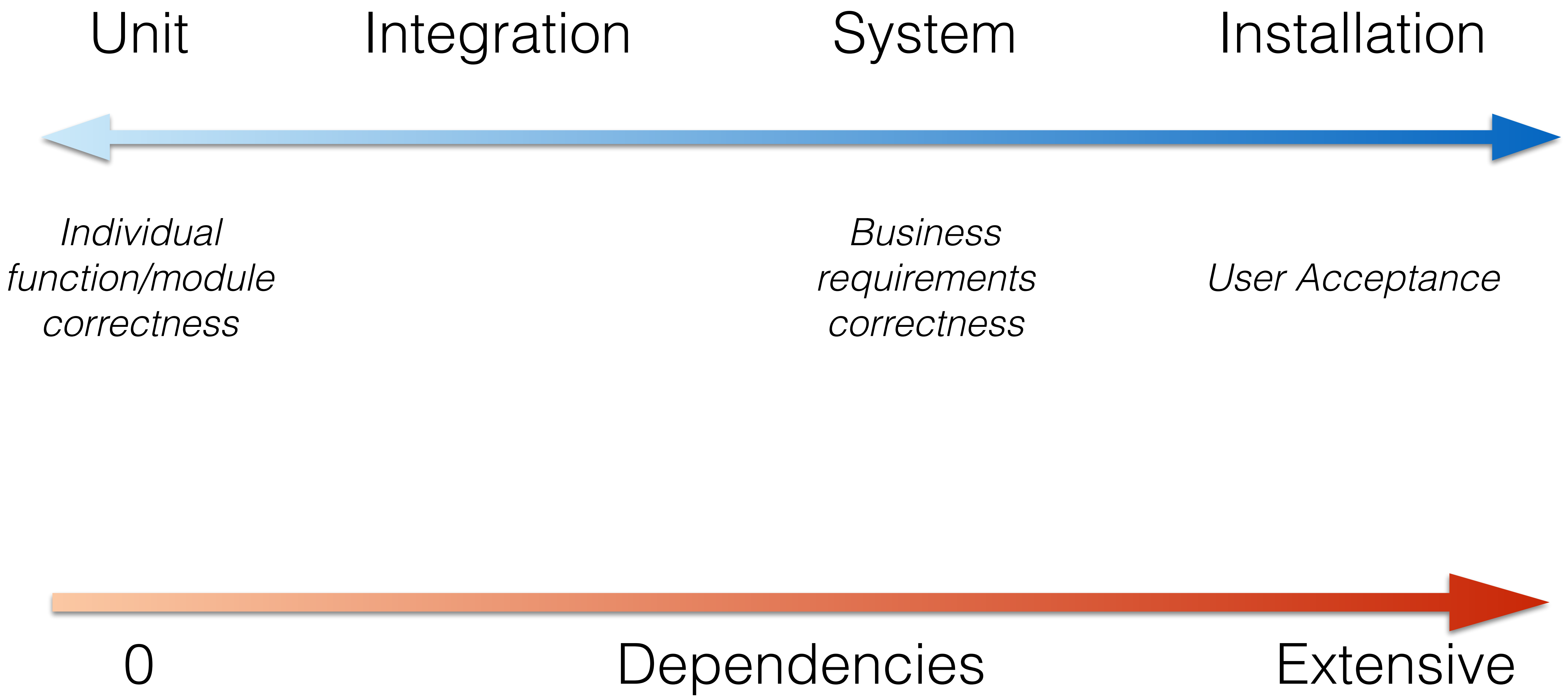
Integration & System Testing


Unit
Integration
System
Installation
Individual function/module correctness
Business requirements correctness
User Acceptance
0
Dependencies
Extensive

Integration Testing

"Integration testing is the process of verifying the interactions among software components. Classical integration testing strategies, such as top-down and bottom-up, are often used with hierarchically structured software.
Modern, systematic integration strategies are typically architecture-driven, which involves incrementally integrating the software components or subsystems based on identified functional threads.
Integration testing is often an ongoing activity at each stage of development during which software engineers abstract away lower-level perspectives and concentrate on the perspectives of the level at which they are integrating. For other than small, simple software, incremental integration testing strategies are usually preferred to putting all of the components together at once—which is often called "big bang" testing." SWEBOK 2.1.2

Extend Unit Testing Model

- Script tests, write test cases, test harnesses/drivers
- Use tools to manage, run regressions
- Does not necessarily test through the UI
- Individual modules have good unit test coverage
- Write tests to test their integration
- Dependencies are usually unavoidable
- Bottom-up is easiest to think about

Example for .NET: SpecFlow+

- specflow.org
- Supports Agile, TDD, ATDD, BDD
- Supports Regression Tests
- Automated UI testing (integration with Selenium)

Acceptance / Qualification Testing

“determines whether a system satisfies its **acceptance criteria**, usually by checking desired system behaviors against the customer’s requirements. The customer or a customer’s representative thus specifies or directly undertakes activities to check that their requirements have been met, or in the case of a consumer product, that the organization has satisfied the stated requirements for the target market. *This testing activity may or may not involve the developers of the system*.” SWEBOK 2.2.1

Acceptance Test/Criteria Form

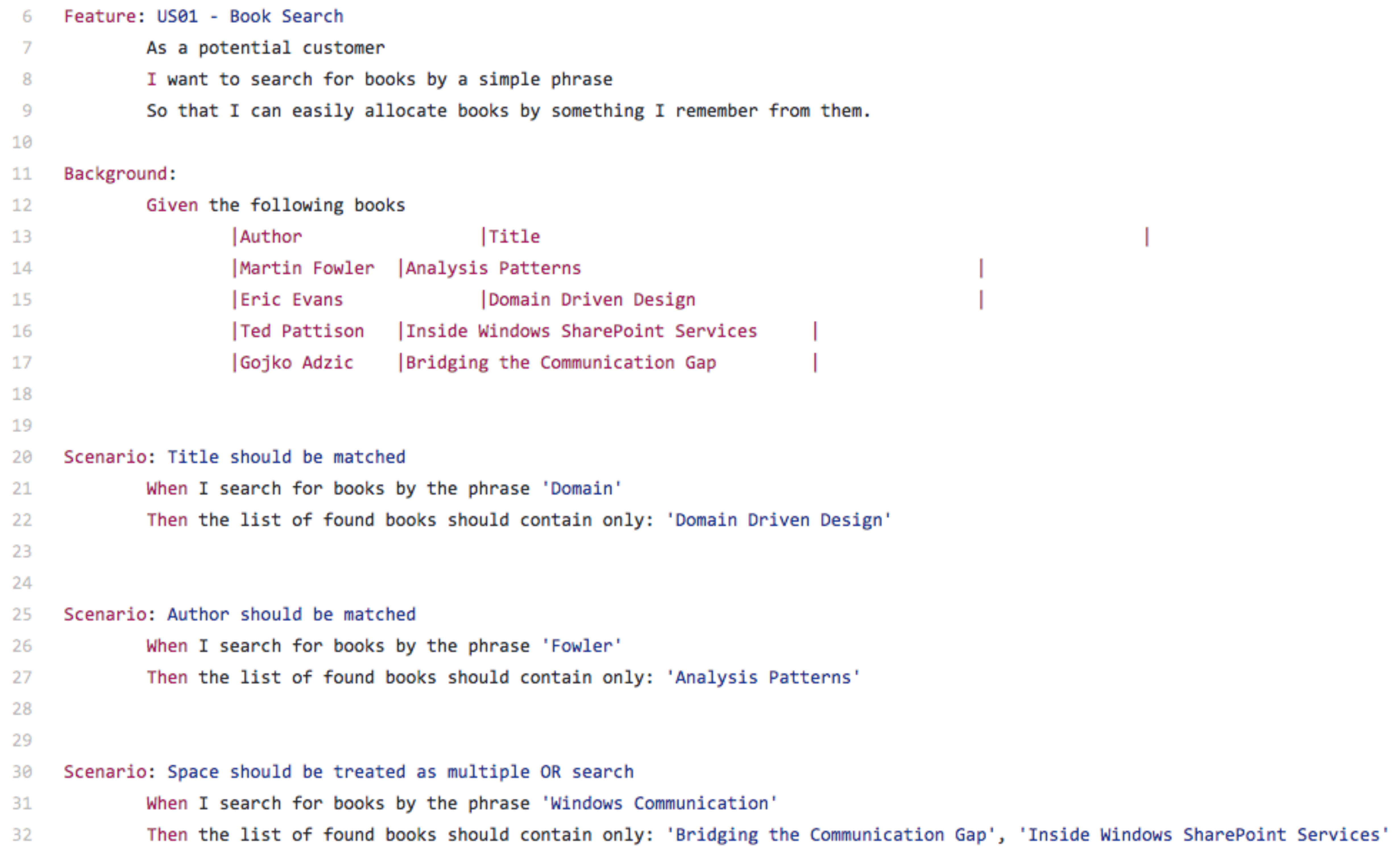
Givens — Whens — Thens

- Given: preconditions or state
- When: actions taken, data submitted
- Then: behavior expected, system results

https://github.com/cucumber/cucumber/wiki/Given-When-Then

https://sites.google.com/site/unclebobconsultingllc/the-truth-about-bdd

Example



```
Feature: US01 - Book Search
        As a potential customer
        I want to search for books by a simple phrase
        So that I can easily allocate books by something I remember from them.

Background:
        Given the following books
                |Author                 |Title                                                           |
                |Martin Fowler  |Analysis Patterns                                       |
                |Eric Evans             |Domain Driven Design                            |
                |Ted Pattison   |Inside Windows SharePoint Services      |
                |Gojko Adzic    |Bridging the Communication Gap          |


Scenario: Title should be matched
        When I search for books by the phrase 'Domain'
        Then the list of found books should contain only: 'Domain Driven Design'


Scenario: Author should be matched
        When I search for books by the phrase 'Fowler'
        Then the list of found books should contain only: 'Analysis Patterns'


Scenario: Space should be treated as multiple OR search
        When I search for books by the phrase 'Windows Communication'
        Then the list of found books should contain only: 'Bridging the Communication Gap', 'Inside Windows SharePoint Services'
```

Gherkin Syntax from Cucumber

https://github.com/cucumber/cucumber/wiki/Gherkin

```
Feature: Some terse yet descriptive text of what is desired
  Textual description of the business value of this feature
  Business rules that govern the scope of the feature
  Any additional information that will make the feature easier to understand

  Scenario: Some determinable business situation
    Given some precondition
      And some other precondition
    When some action by the actor
      And some other action
      And yet another action
    Then some testable outcome is achieved
      And something else we can check happens too

  Scenario: A different situation
      ...
```

More Examples

- SpecFlow BookShop Sample
  https://github.com/techtalk/SpecFlow-Examples/tree/master/ASP.NET-MVC/BookShop
    - Unit tests, Acceptance tests, Selenium tests of UI
- Test Automation
    - http://specflow.org/getting-started/

Selenium

- Web browser automation
- Selenium WebDriver (Java, C#, Python, Ruby, Perl, PHP, JavaScript)
- Selenium IDE (Firefox)
- Demo or YouTube (https://www.youtube.com/watch?v=gsHyDIyA3dg)

The Plan for Us

- Write Acceptance Tests for **Every** User Story in the Sprint
  - in Gherkin Syntax
  - 1 User Story == 1 Feature, >=2 Scenarios
  - need good coverage of the user story
  - each Feature in its own feature file in a testing folder in your repository, named with the user story ID from VSTS,i.e. `ID53.feature`, and containing all Scenarios
  - Use Selenium (by hand) to perform every test, recording it and saving in a test suite for regression tests. Save scripted tests to same folder in repo with same ID
  - Be able to run all tests and show green board during Sprint Review mtg.
  - Maintain spreadsheet file to show status of tests
  - The feature file replaces the acceptance tests in VSTS (i.e. don't enter acceptance tests in VSTS for new user stories)

Team Falcon Precision Development

Integration Test Panel

ID	Scenario Title	Tester	Date	Result	Notes
53	Home page has convenient link/button to view past reports	morses	4/18/17	PASS	
53	Home page has convenient link/button to create a new report	morses	4/18/17	PASS	
54	Meetings page shows past meetings in reverse chronological order	morses	4/18/17	PASS	
55	Admin user has all admin links on navbar, all work correctly	morses	4/18/17	FAIL	"Home" link is incorrect; shows "/Admin/Reports?Length=4", should be "/Admin/Index"

http://www.wou.edu/~morses/cs46X/resources/ITestPanel.xlsx