AGILE TESTING

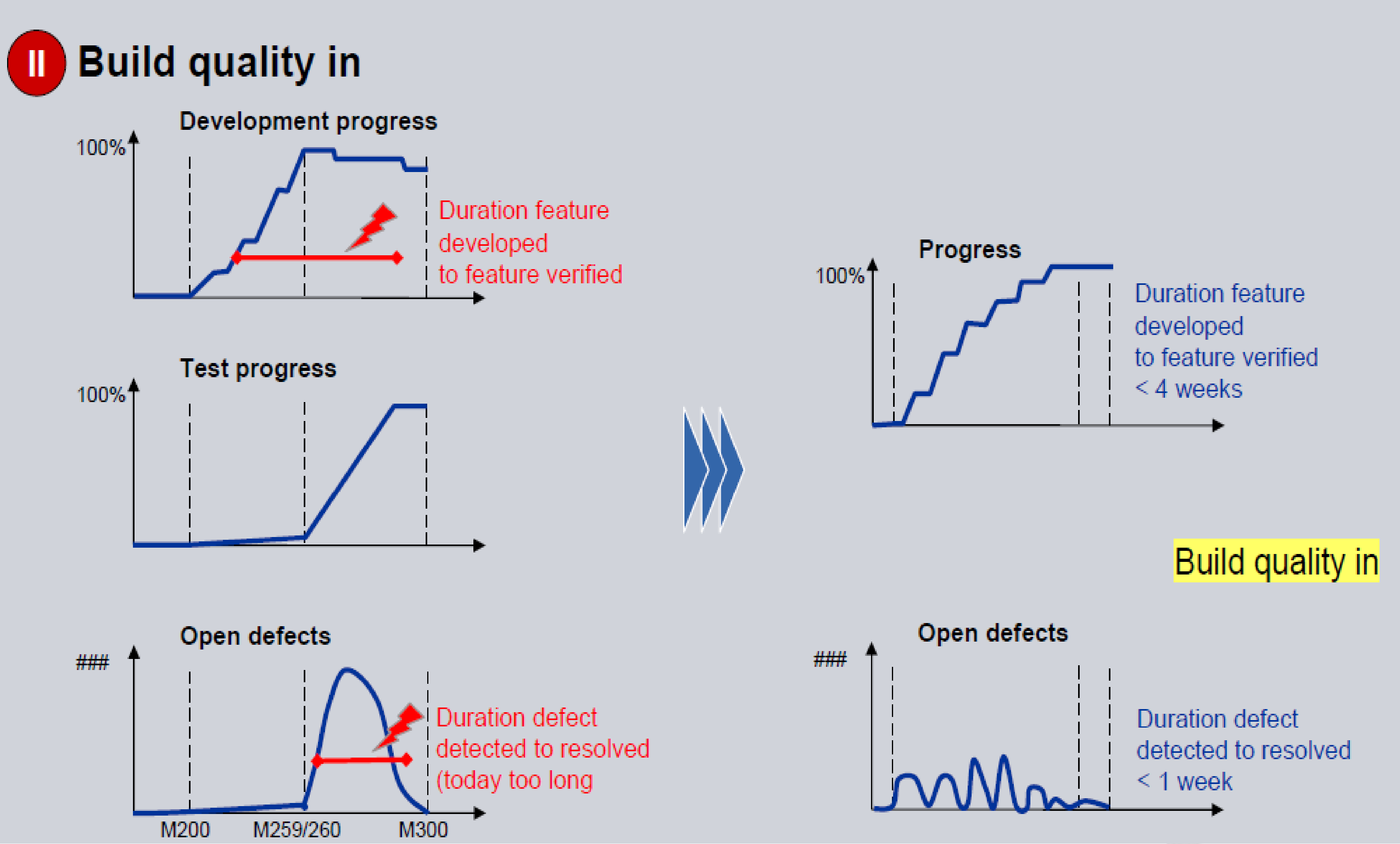
A.K.A DEV-QA -THE NEXT GENERATION

So We've Gone Kanban w/ some Feature Teams…

So we're talking about…

- Whole Team Approach – WIP Limits keep us together focused on delivering Working Tested Clean Software
- Delivering and testing smaller stories much more frequently

Agile is all about early feedback – why?



II Build quality in

Duration feature developed to feature verified

Duration defect detected to resolved (today too long

Duration feature developed to feature verified < 4 weeks

Build quality in

Duration defect detected to resolved < 1 week

http://less2010.leanssc.org/wp-content/uploads/2010/10/Siemens_Rudolf_Paulisch.pdf

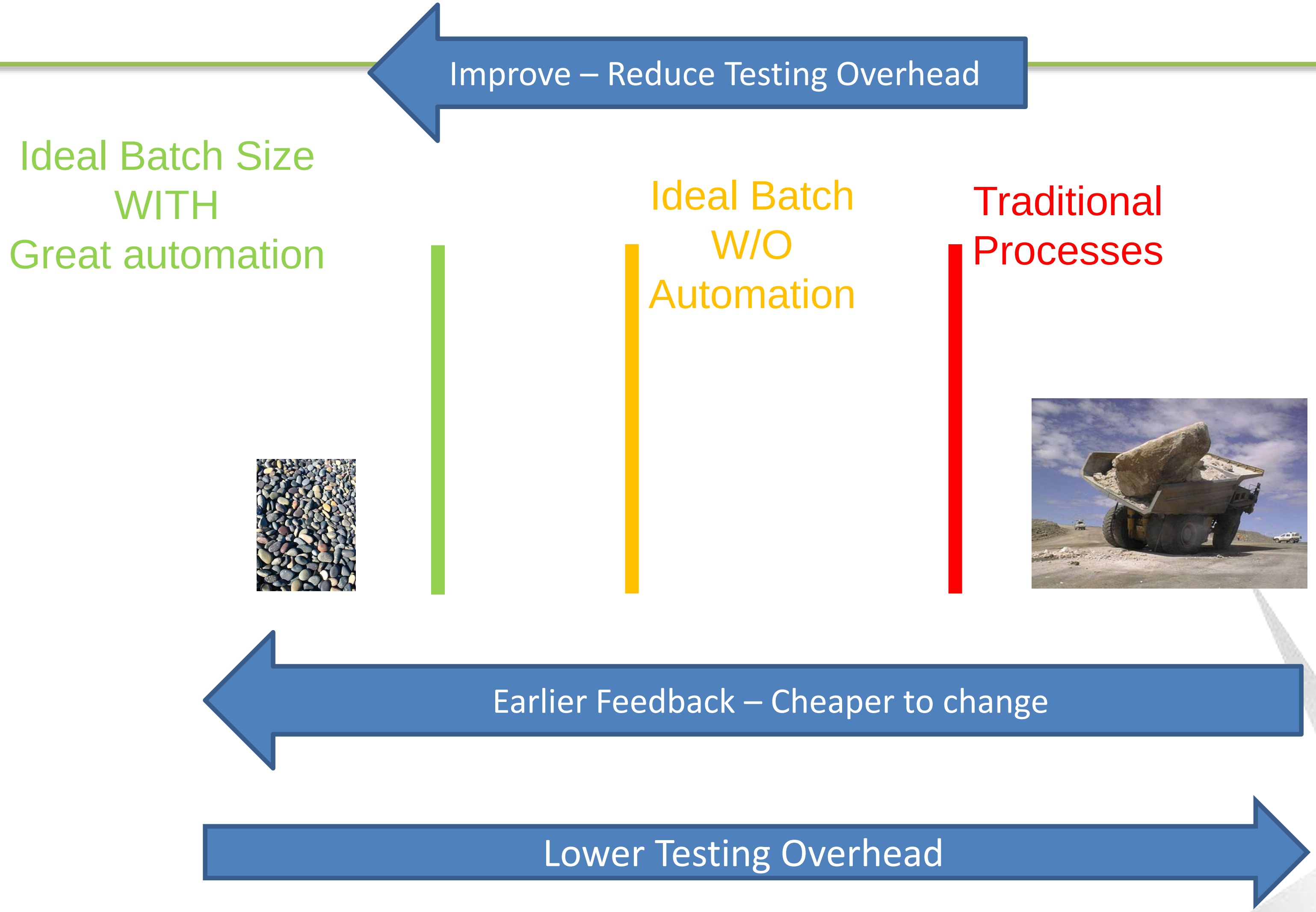
How WILL WE get early feedback ?

BIG Features

That take very long to get to testing…

test

Early Feedback – The Goal and the conflict…



Improve – Reduce Testing Overhead

Ideal Batch Size
WITH
Great automation

Ideal Batch
W/O
Automation

Traditional
Processes

Earlier Feedback – Cheaper to change

Lower Testing Overhead

- Even without reducing testing overhead it is usually more cost-effective to reduce batch size
- Aim to reduce testing overhead to reduce batch size even more and be even more cost-effective

Continuous Integration

"Works on my machine"?!

This is Phyllis and she doesn't care if the build works on your machine. As you can see, she's a busy woman with a jam-packed social calendar. She doesn't have time for brittle builds from the likes of you; she needs a build that just plain works. You hear?!

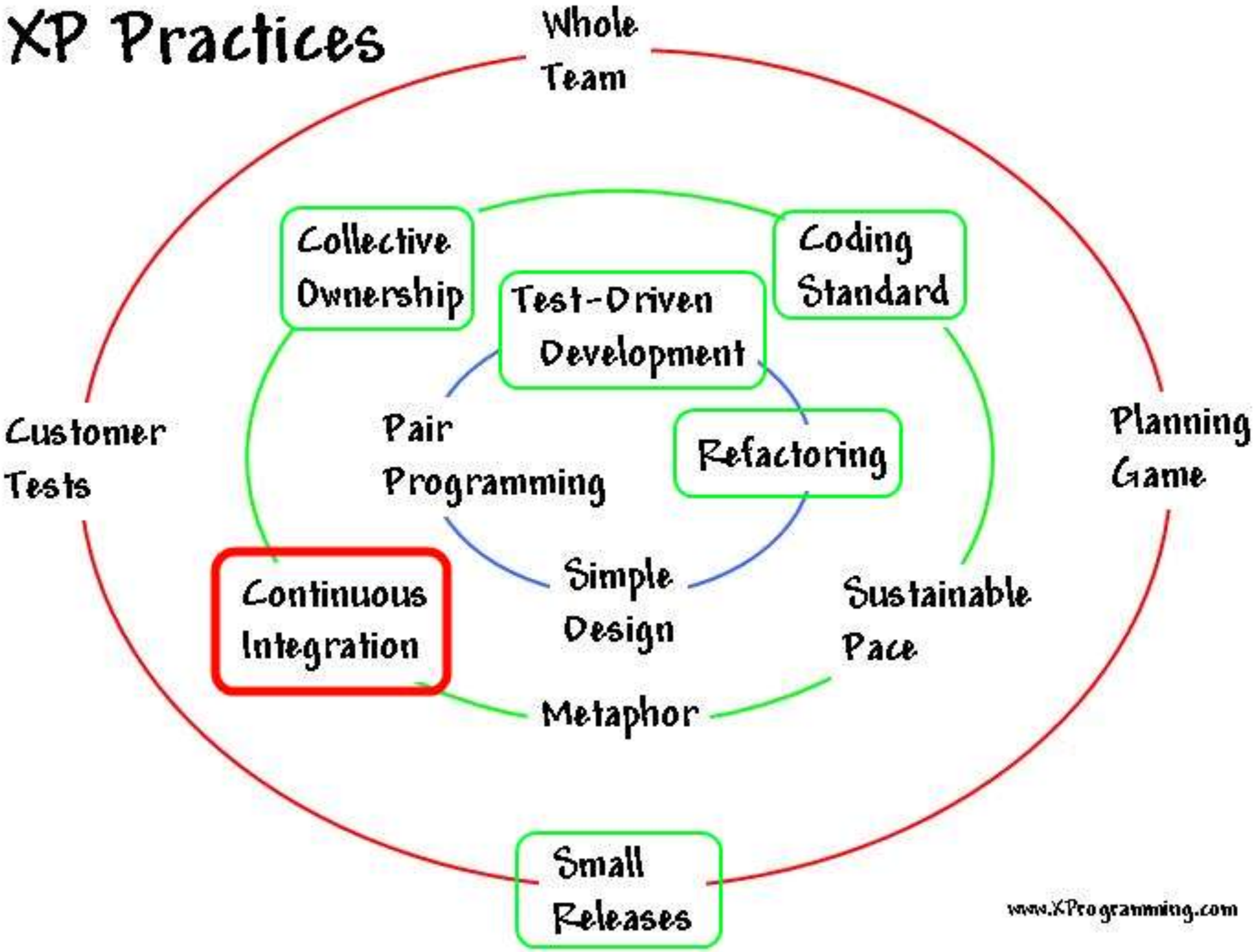
If you start using continuous integration, Phyllis can get back to gettin' her groove-thing on.

www.BuildsOnMyMachine.com

Continuous Integration (CI) – The backbone for Agile testing

- the practice of integrating early and often
  - avoid the pitfalls of "integration hell"
- “stop and fix”
- Always have a working system based on latest code

Continuous integration in the XP framework



But how do we get enough coverage for the Continuous Integration system?

We automate tests as part of Definiton of Done

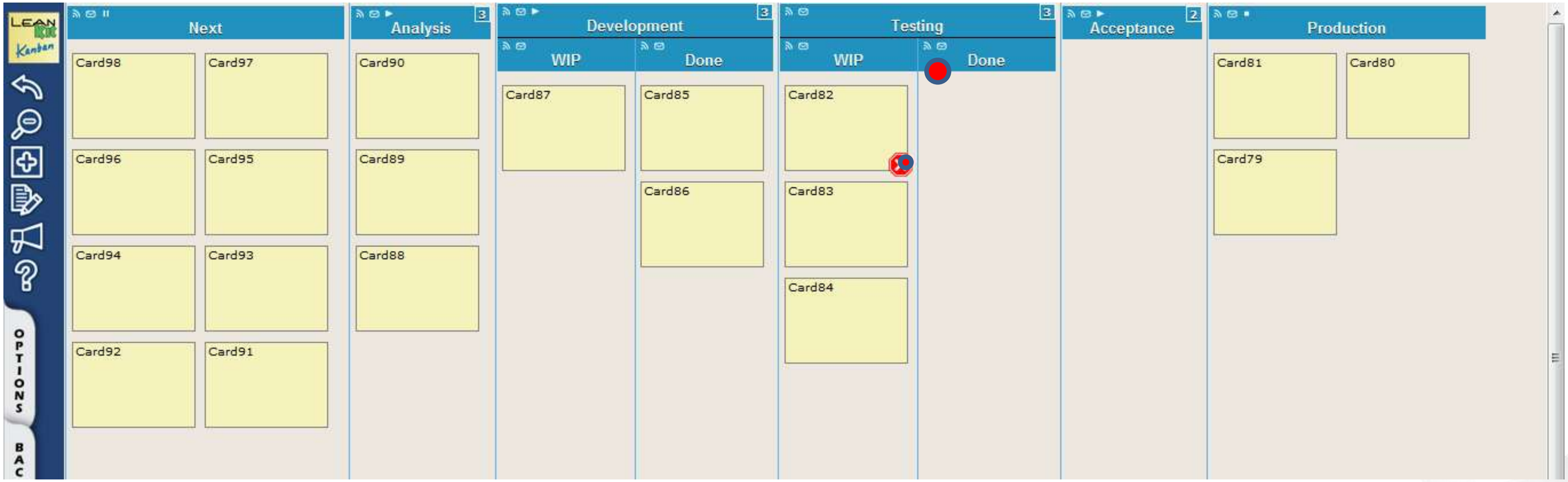
Well, Dah…

But what might happen then?

Pop Quiz

- What does this mean?

Blocked/ Impeded Card



LEAN KIT Kanban

OPTIONS

BAC

Next: Card98, Card97, Card96, Card95, Card94, Card93, Card92, Card91

Analysis 3: Card90, Card89, Card88

Development 3

- WIP: Card87
- Done: Card85, Card86

Testing 3

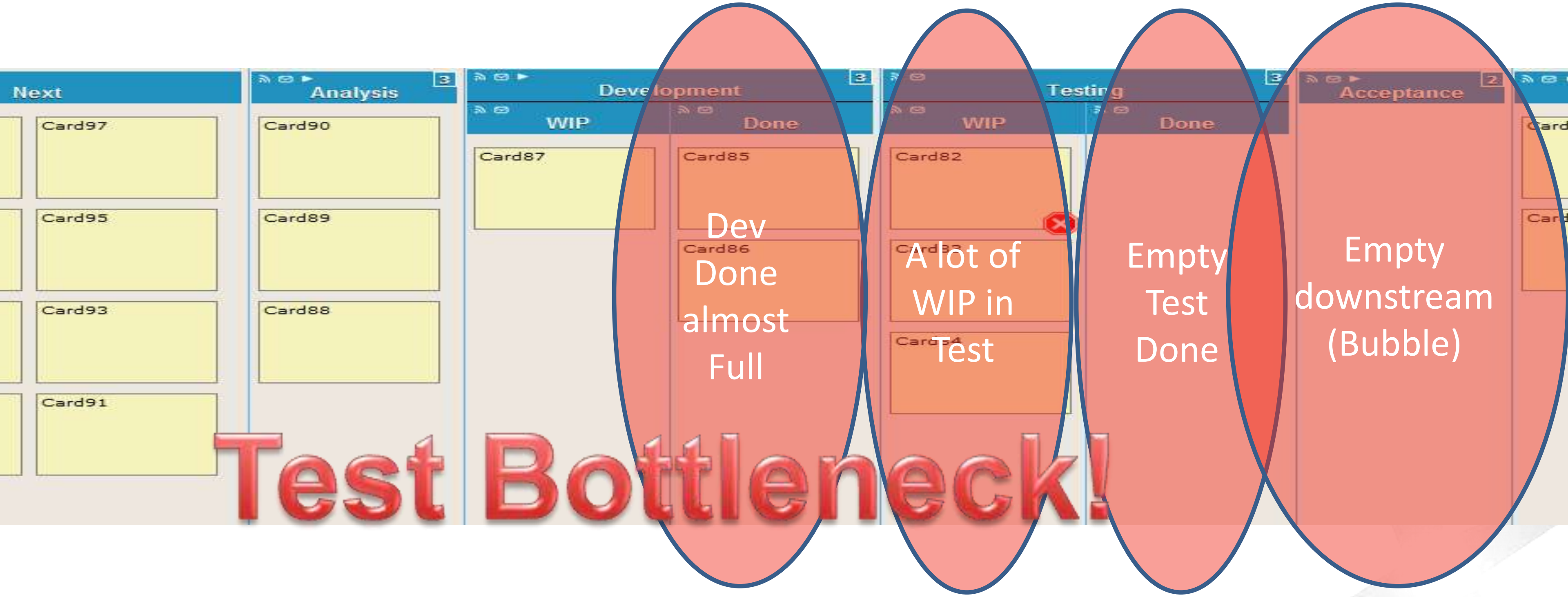
- WIP: Card82, Card83, Card84
- Done

Acceptance 2

Production: Card81, Card80, Card79

Full story at http://yuvalyeret.com/2010/08/03/finding-the-right-dev-to-test-ratio-when-working-in-kanban/

Pop Quiz



Dev Done almost Full

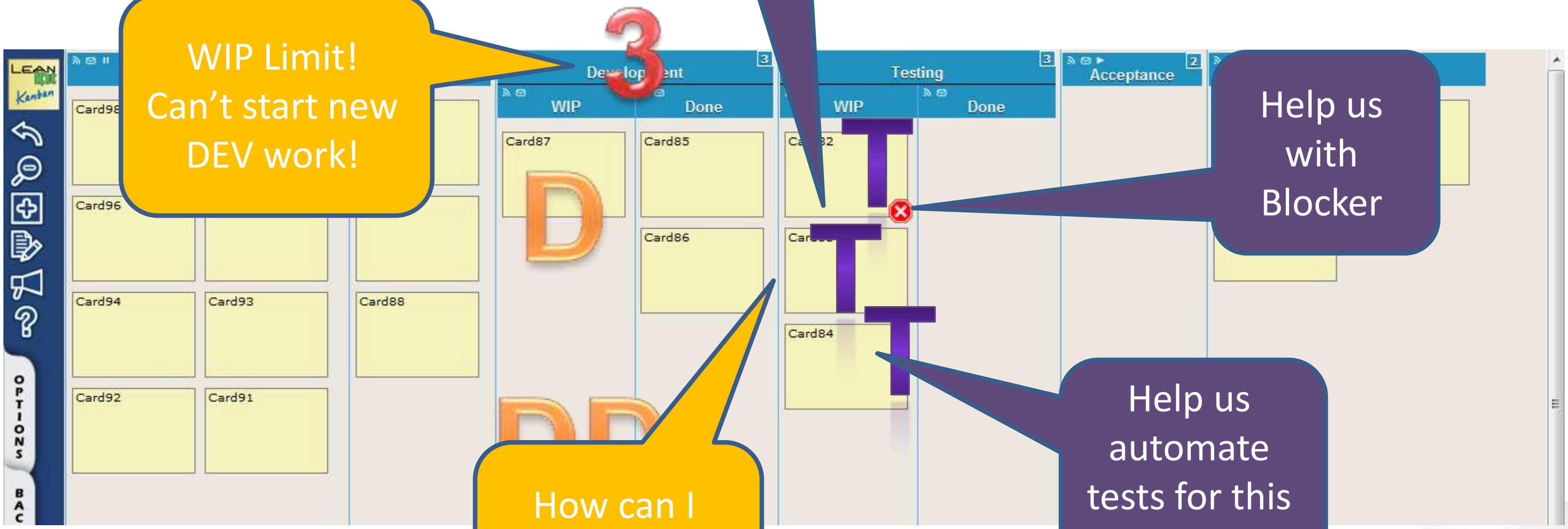
A lot of WIP in Test

Empty Test Done

Empty downstream (Bubble)

Test Bottleneck!

What LIMITED WIP Does



Fix open defects on our Stories

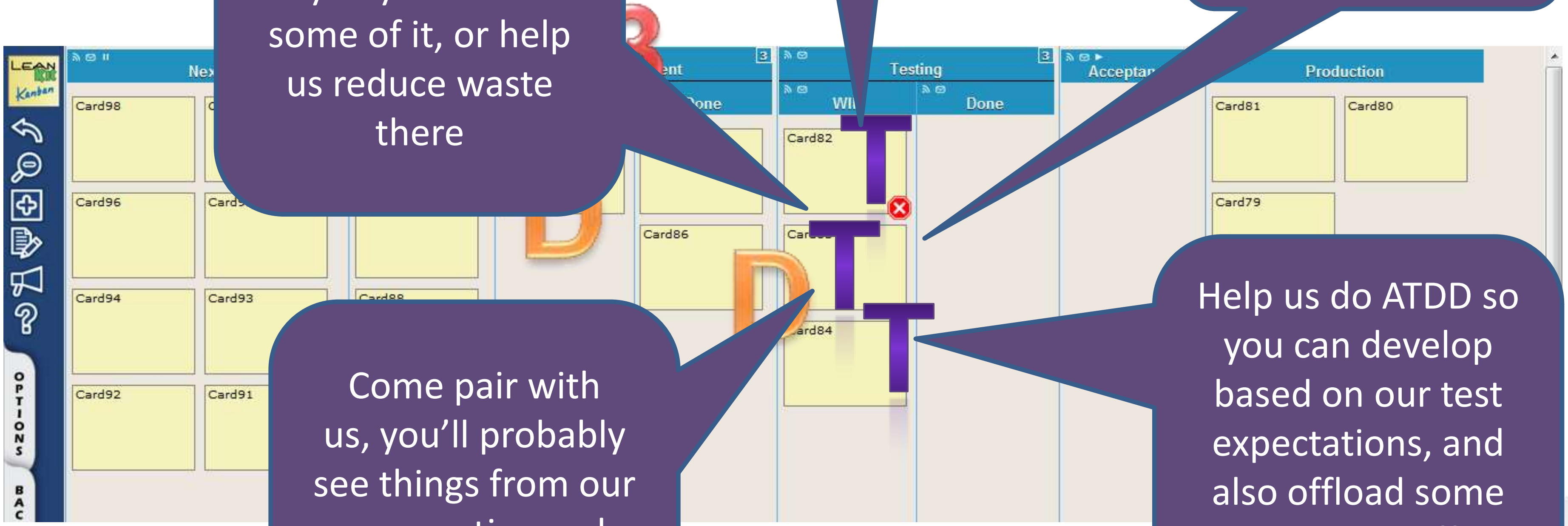
WIP Limit! Can't start new DEV work!

Help us with Blocker

How can I help current stories?

Help us automate tests for this story

What LIMITED WIP Does



Half of our work is not core test work. Maybe you can take some of it, or help us reduce waste there

Automate Setups and Test Data

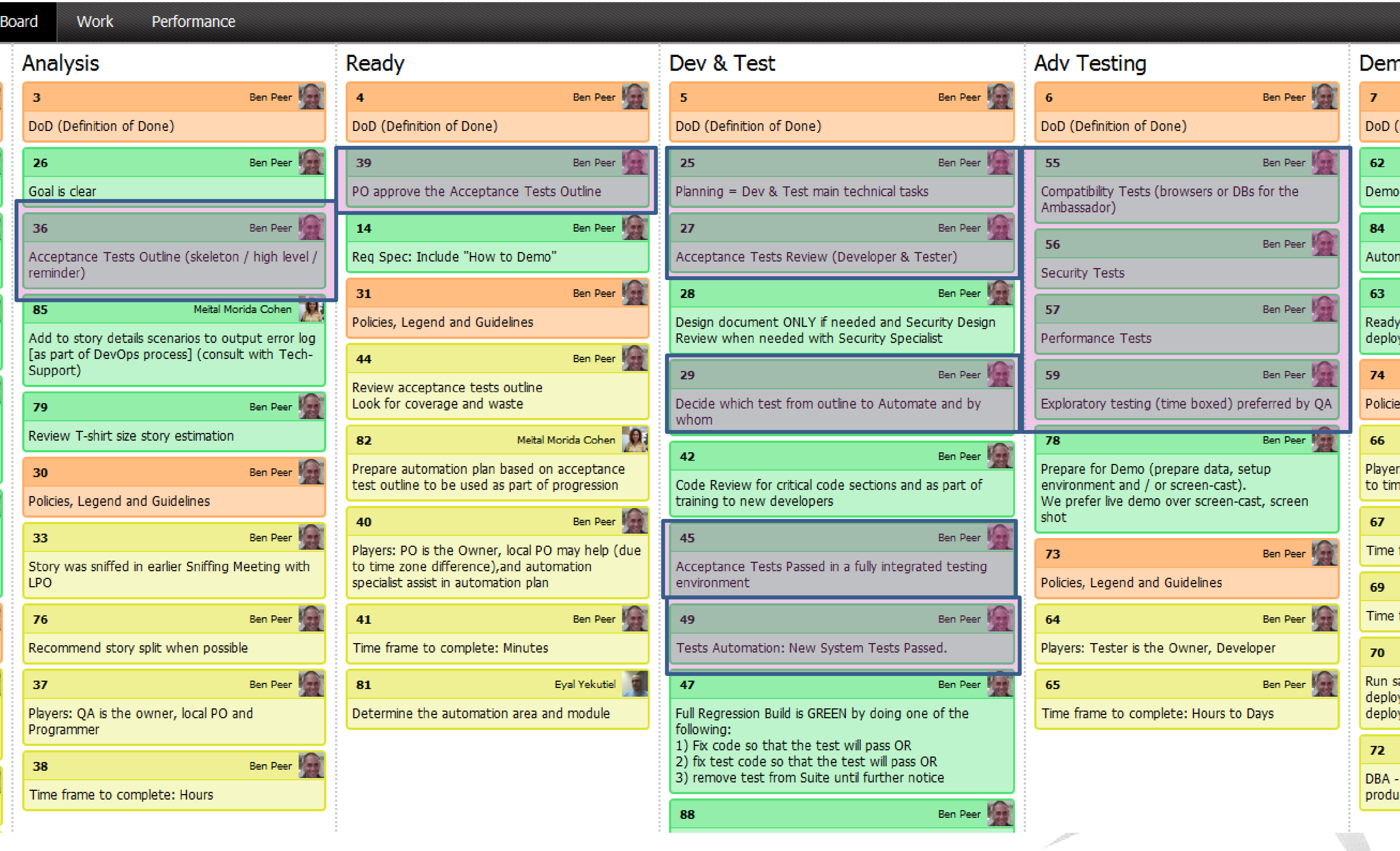
Improve Dev Done quality! – less retesting for us

Come pair with us, you'll probably see things from our perspective and have some ideas how to help!

Help us do ATDD so you can develop based on our test expectations, and also offload some automation effort from us

How can I help you be more efficient?

Agile Testing driven by Flow



Analysis

3 Ben Peer
DoD (Definition of Done)

26 Ben Peer
Goal is clear

36 Ben Peer
Acceptance Tests Outline (skeleton / high level / reminder)

85 Meital Morida Cohen
Add to story details scenarios to output error log [as part of DevOps process] (consult with Tech-Support)

79 Ben Peer
Review T-shirt size story estimation

30 Ben Peer
Policies, Legend and Guidelines

33 Ben Peer
Story was sniffed in earlier Sniffing Meeting with LPO

76 Ben Peer
Recommend story split when possible

37 Ben Peer
Players: QA is the owner, local PO and Programmer

38 Ben Peer
Time frame to complete: Hours

Ready

4 Ben Peer
DoD (Definition of Done)

39 Ben Peer
PO approve the Acceptance Tests Outline

14 Ben Peer
Req Spec: Include "How to Demo"

31 Ben Peer
Policies, Legend and Guidelines

44 Ben Peer
Review acceptance tests outline
Look for coverage and waste

82 Meital Morida Cohen
Prepare automation plan based on acceptance test outline to be used as part of progression

40 Ben Peer
Players: PO is the Owner, local PO may help (due to time zone difference),and automation specialist assist in automation plan

41 Ben Peer
Time frame to complete: Minutes

81 Eyal Yekutiel
Determine the automation area and module

Dev & Test

5 Ben Peer
DoD (Definition of Done)

25 Ben Peer
Planning = Dev & Test main technical tasks

27 Ben Peer
Acceptance Tests Review (Developer & Tester)

28 Ben Peer
Design document ONLY if needed and Security Design Review when needed with Security Specialist

29 Ben Peer
Decide which test from outline to Automate and by whom

42 Ben Peer
Code Review for critical code sections and as part of training to new developers

45 Ben Peer
Acceptance Tests Passed in a fully integrated testing environment

49 Ben Peer
Tests Automation: New System Tests Passed.

47 Ben Peer
Full Regression Build is GREEN by doing one of the following:
1) Fix code so that the test will pass OR
2) fix test code so that the test will pass OR
3) remove test from Suite until further notice

88 Ben Peer

Adv Testing

6 Ben Peer
DoD (Definition of Done)

55 Ben Peer
Compatibility Tests (browsers or DBs for the Ambassador)

56 Ben Peer
Security Tests

57 Ben Peer
Performance Tests

59 Ben Peer
Exploratory testing (time boxed) preferred by QA

78 Ben Peer
Prepare for Demo (prepare data, setup environment and / or screen-cast).
We prefer live demo over screen-cast, screen shot

73 Ben Peer
Policies, Legend and Guidelines

64 Ben Peer
Players: Tester is the Owner, Developer

65 Ben Peer
Time frame to complete: Hours to Days

Dem

7
DoD (

62
Demo

84
Auton

63
Ready
deploy

74
Policie

66
Player
to tim

67
Time f

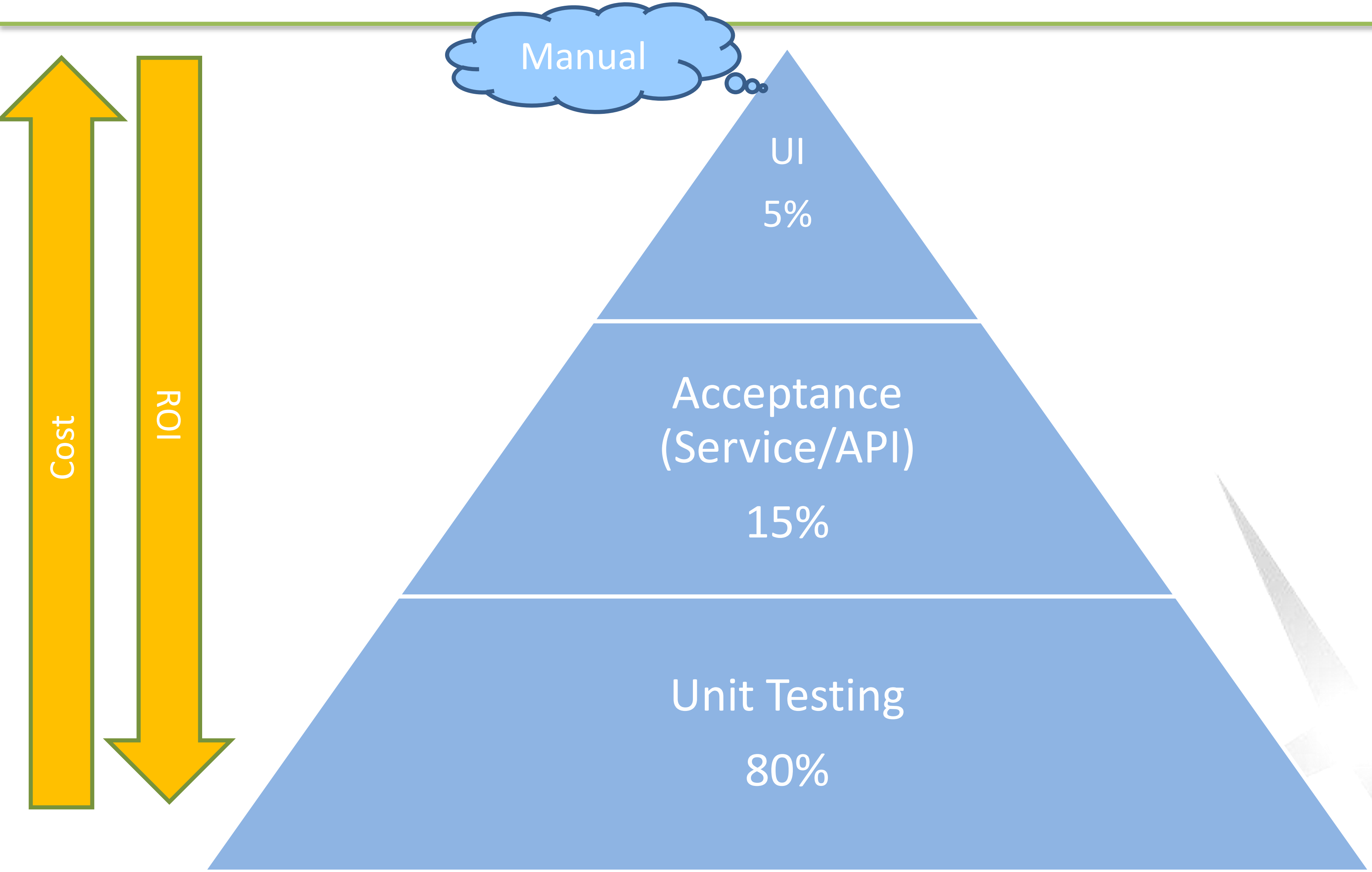
69
Time f

70
Run sa
deploy
deploy

72
DBA -
produ

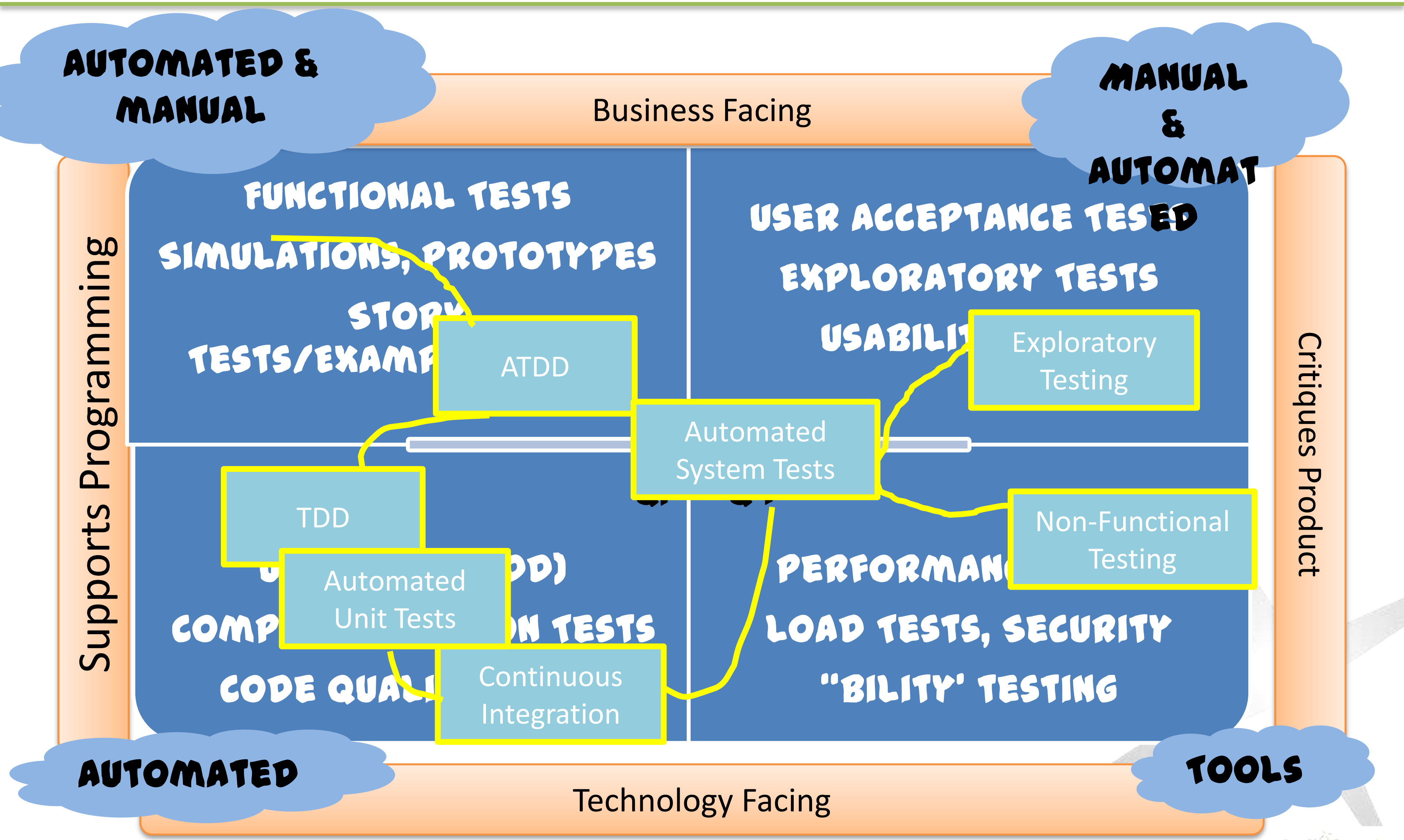
AgileSparks

Automate at the right level



http://www.mountaingoatsoftware.com/blog/the-forgotten-layer-of-the-test-automation-pyramid

If we only had limited QA capacity, what would we focus on? What would we enable Devs to do?



AUTOMATED & MANUAL

Business Facing

MANUAL & AUTOMATED

Supports Programming

Critiques Product

FUNCTIONAL TESTS

SIMULATIONS, PROTOTYPES

STORY TESTS/EXAMP

USER ACCEPTANCE TESTS

EXPLORATORY TESTS

USABILIT

ATDD

Exploratory Testing

Automated System Tests

TDD

Automated Unit Tests

Non-Functional Testing

U DD)

COMP N TESTS

CODE QUAL

Continuous Integration

PERFORMAN

LOAD TESTS, SECURITY

"BILITY' TESTING

AUTOMATED

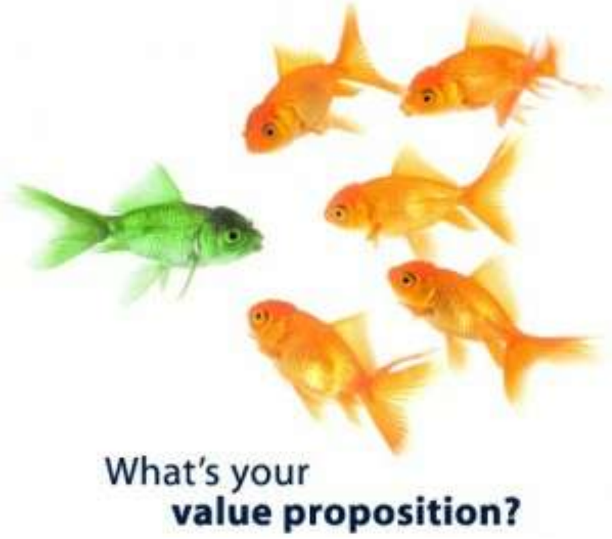
Technology Facing

TOOLS

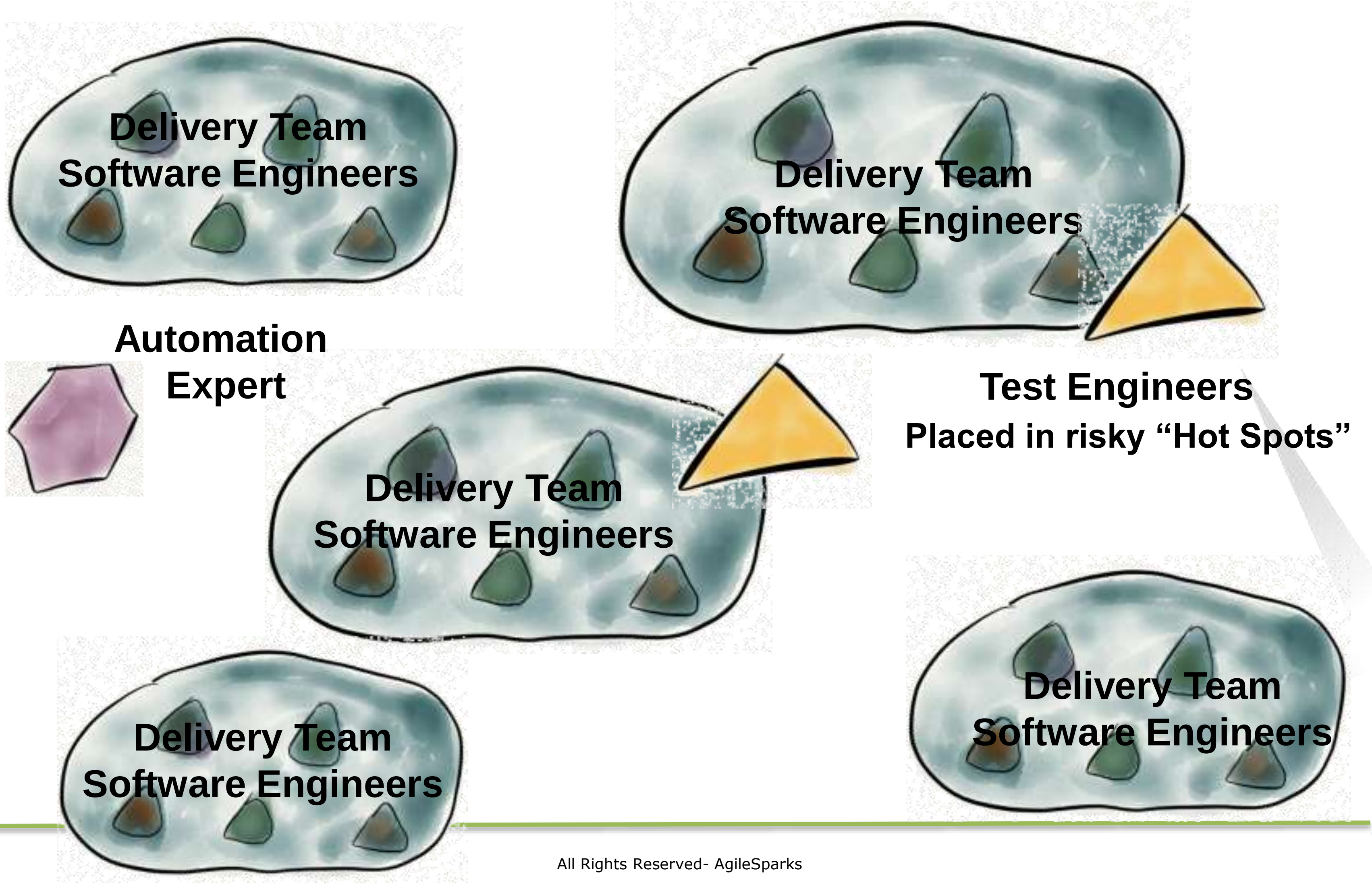
AgileSparks

The UNIQUE role of the QA engineers

- Being Champions of the Product and the Customer/User.
- Specializing in Performance/ Security/Load/etc.
- Shining light on where to focus quality efforts by analyzing risk probability and Impact.



Quality OVER Quantity – QA expertise SUPPORTING delivery



Delivery Team
Software Engineers

Delivery Team
Software Engineers

Automation
Expert

Test Engineers
Placed in risky “Hot Spots”

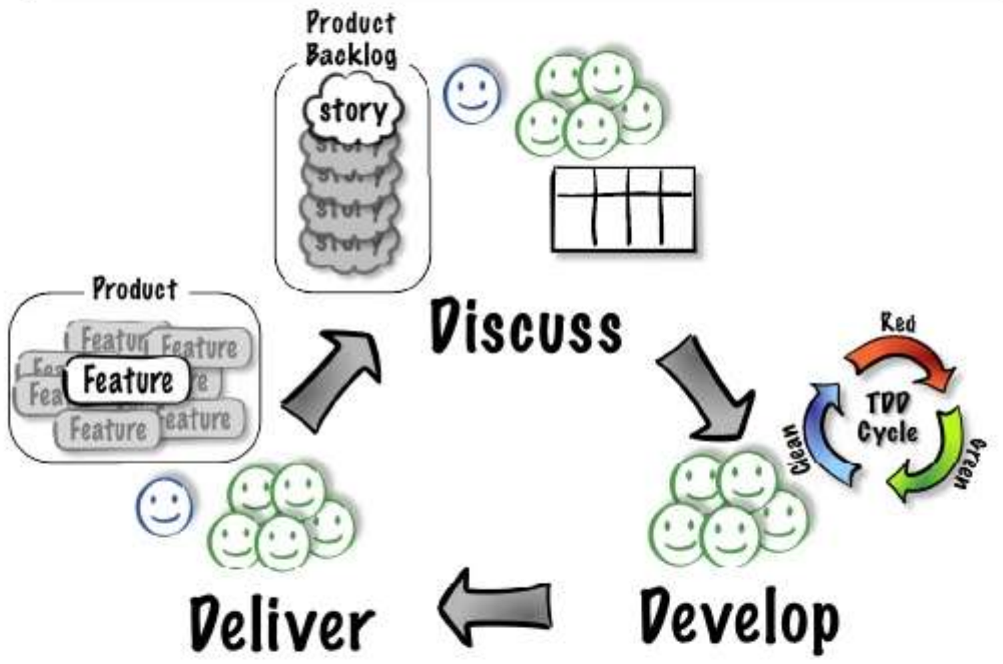
Delivery Team
Software Engineers

Delivery Team
Software Engineers

Delivery Team
Software Engineers

Driving from Behaviour/Examples/Acceptance Tests

Acceptance-Test Driven Development (ATDD) Cycle



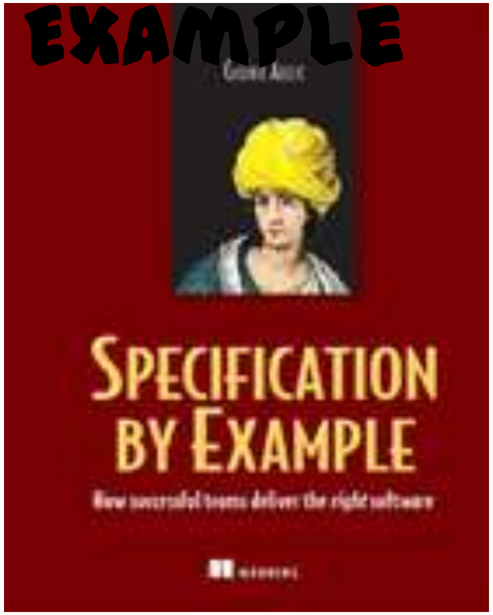
(Model developed with Pekka Klärck, Bas Vodde, and Craig Larman.)

Copyright © 2010 Quality Tree Software, Inc.

http://slidesha.re/LqQRa3 Intro to ATDD - Elisabeth Hendrickson

Meet the family

SPECIFICATION BY EXAMPLE



Behavior Driven Development

http://dannorth.net/introducing-bdd/

Very important – “Step away from the tools”

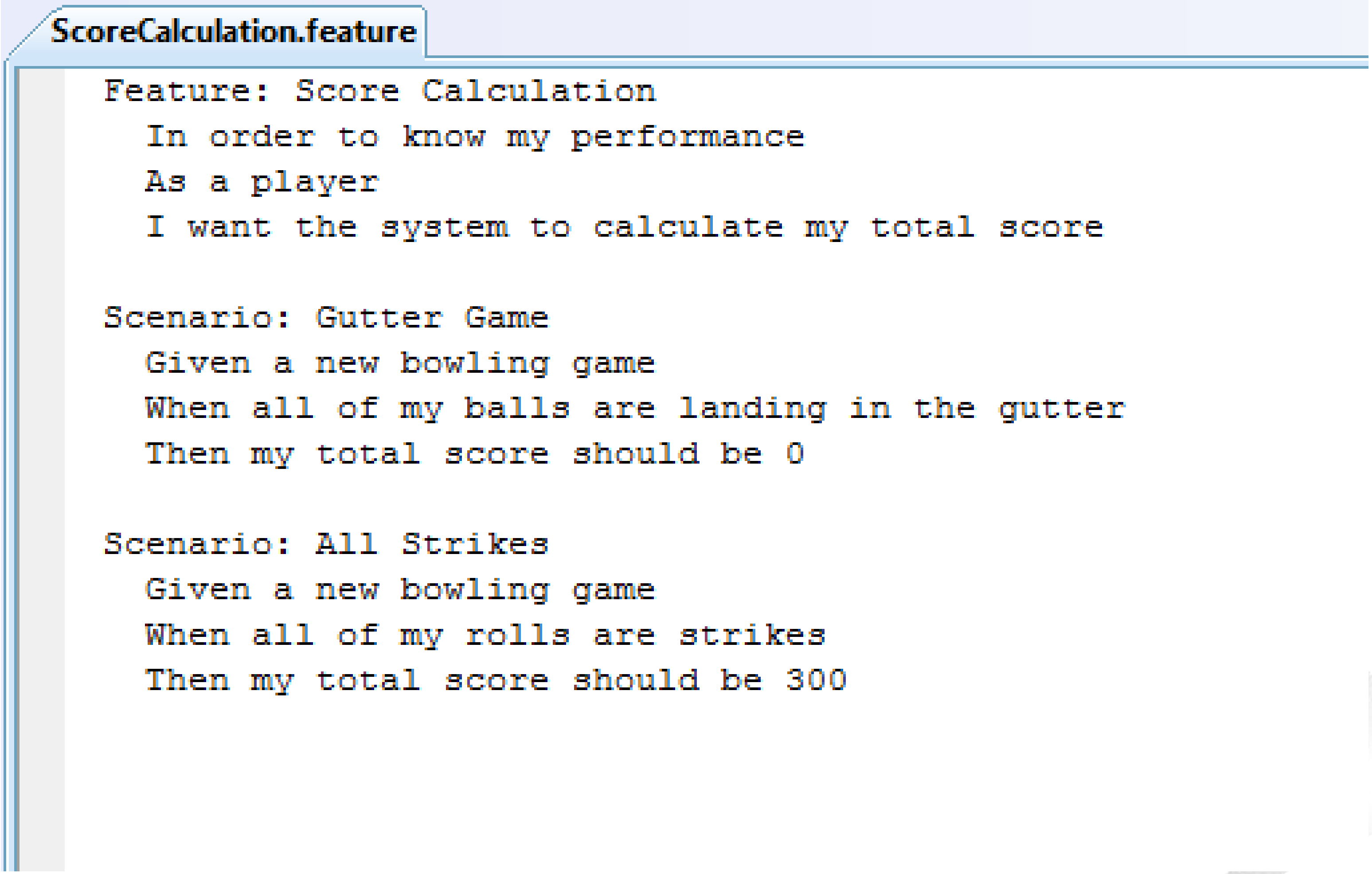
http://lizkeogh.com/2011/03/04/step-away-from-the-tools/

Let's look at a concrete workflow using SpecFlow

(Pragmatic BDD for .Net)

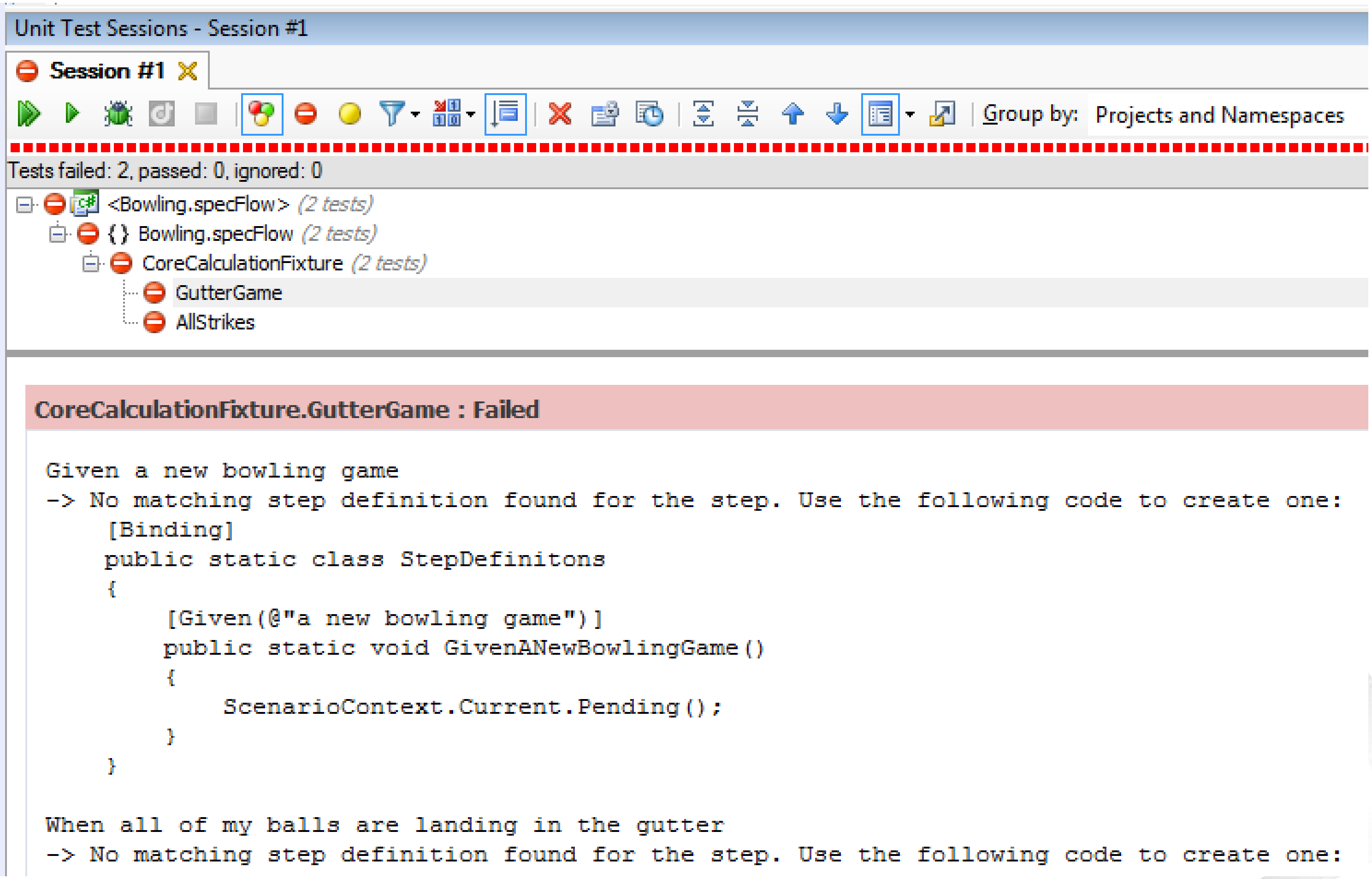
http://www.specflow.org/specflow/workflow.aspx

Step 1 - Write a Feature (using Gherkin language)



```
Feature: Score Calculation
  In order to know my performance
  As a player
  I want the system to calculate my total score

Scenario: Gutter Game
  Given a new bowling game
  When all of my balls are landing in the gutter
  Then my total score should be 0

Scenario: All Strikes
  Given a new bowling game
  When all of my rolls are strikes
  Then my total score should be 300
```

Step 2 - Watch it Fail



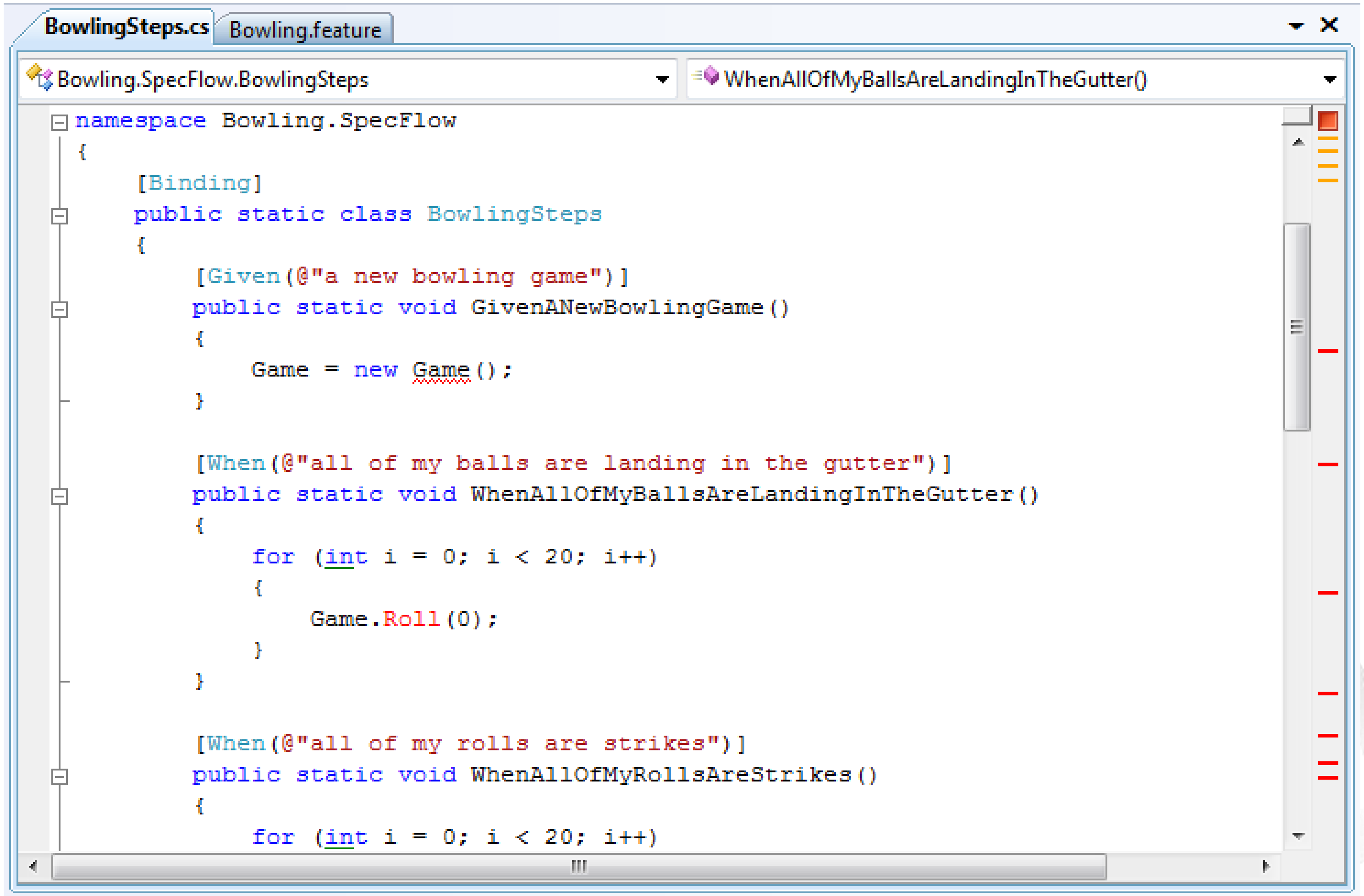
Tests failed: 2, passed: 0, ignored: 0

```
<Bowling.specFlow> (2 tests)
  { } Bowling.specFlow (2 tests)
    CoreCalculationFixture (2 tests)
      GutterGame
      AllStrikes
```

CoreCalculationFixture.GutterGame : Failed

```
Given a new bowling game
-> No matching step definition found for the step. Use the following code to create one:
    [Binding]
    public static class StepDefinitons
    {
        [Given(@"a new bowling game")]
        public static void GivenANewBowlingGame()
        {
            ScenarioContext.Current.Pending();
        }
    }

When all of my balls are landing in the gutter
-> No matching step definition found for the step. Use the following code to create one:
```

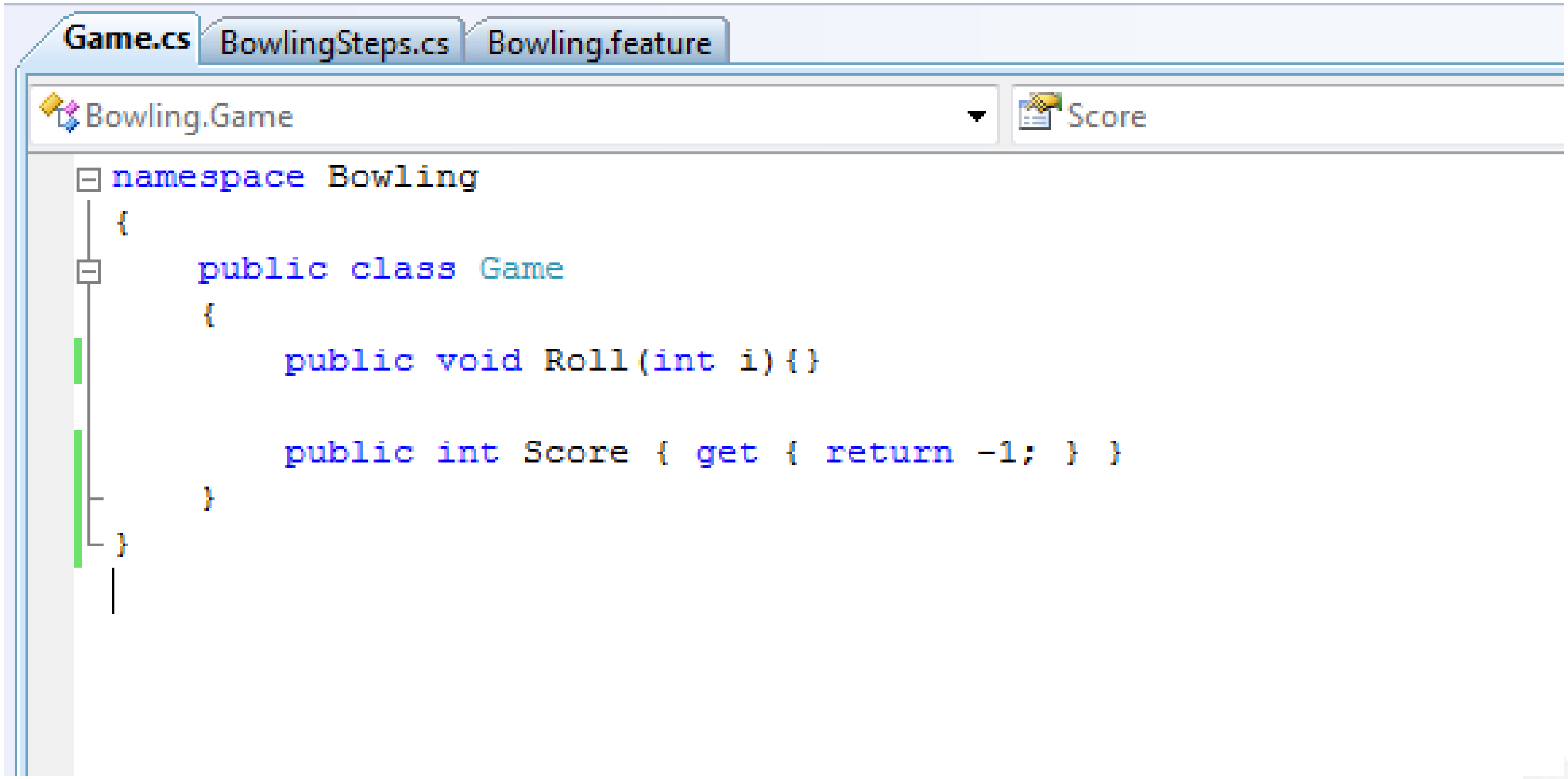
Step 3 – Implement Step definitions



BowlingSteps.cs
Bowling.feature
Bowling.SpecFlow.BowlingSteps
WhenAllOfMyBallsAreLandingInTheGutter()

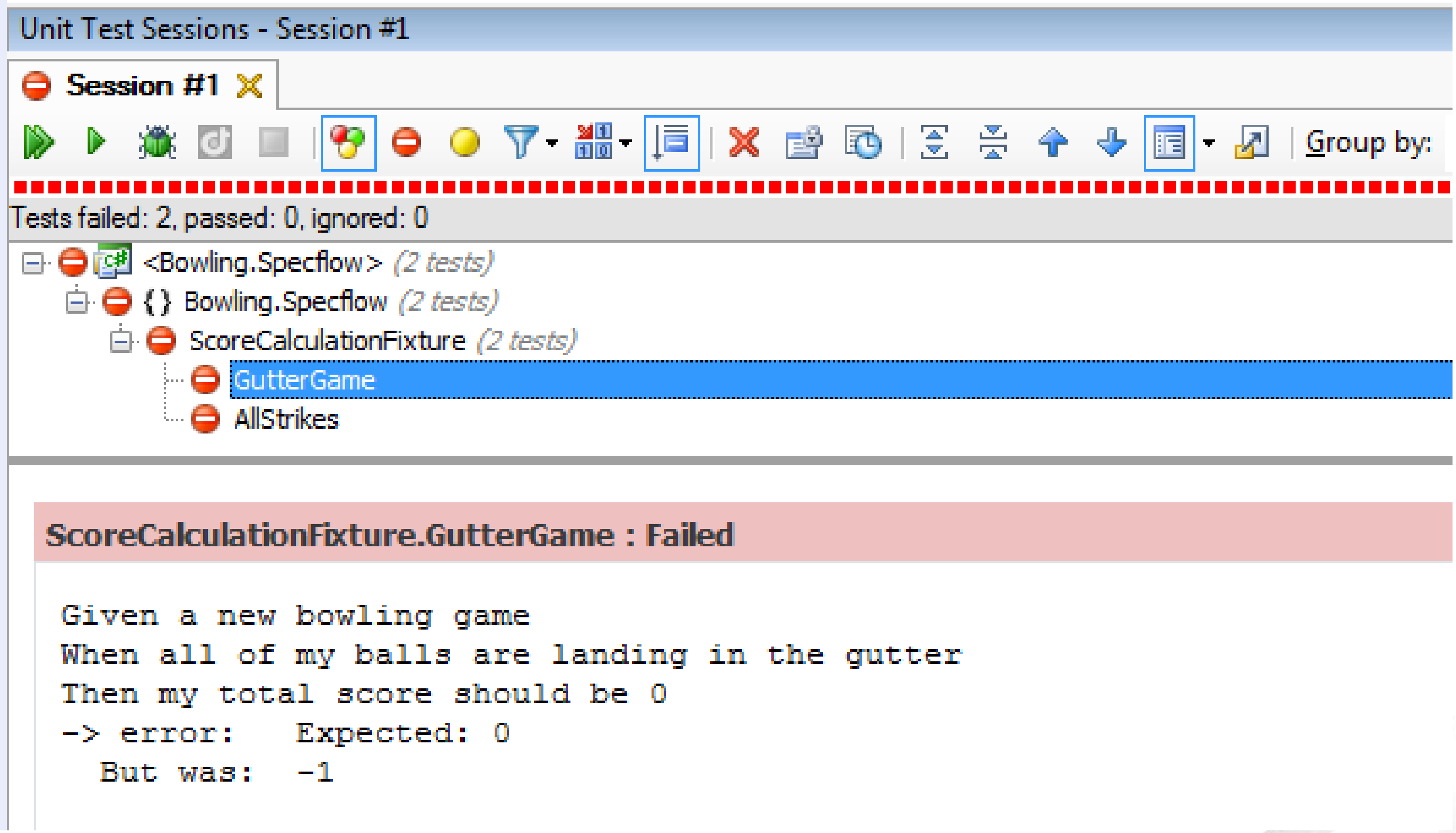
```
namespace Bowling.SpecFlow
{
    [Binding]
    public static class BowlingSteps
    {
        [Given(@"a new bowling game")]
        public static void GivenANewBowlingGame()
        {
            Game = new Game();
        }

        [When(@"all of my balls are landing in the gutter")]
        public static void WhenAllOfMyBallsAreLandingInTheGutter()
        {
            for (int i = 0; i < 20; i++)
            {
                Game.Roll(0);
            }
        }

        [When(@"all of my rolls are strikes")]
        public static void WhenAllOfMyRollsAreStrikes()
        {
            for (int i = 0; i < 20; i++)
```

Step 4 – Create Domain Skeleton



```
namespace Bowling
{
    public class Game
    {
        public void Roll(int i){}

        public int Score { get { return -1; } }
    }
}
```

Step 5 - Watch it fail


Unit Test Sessions - Session #1
Session #1
Group by:

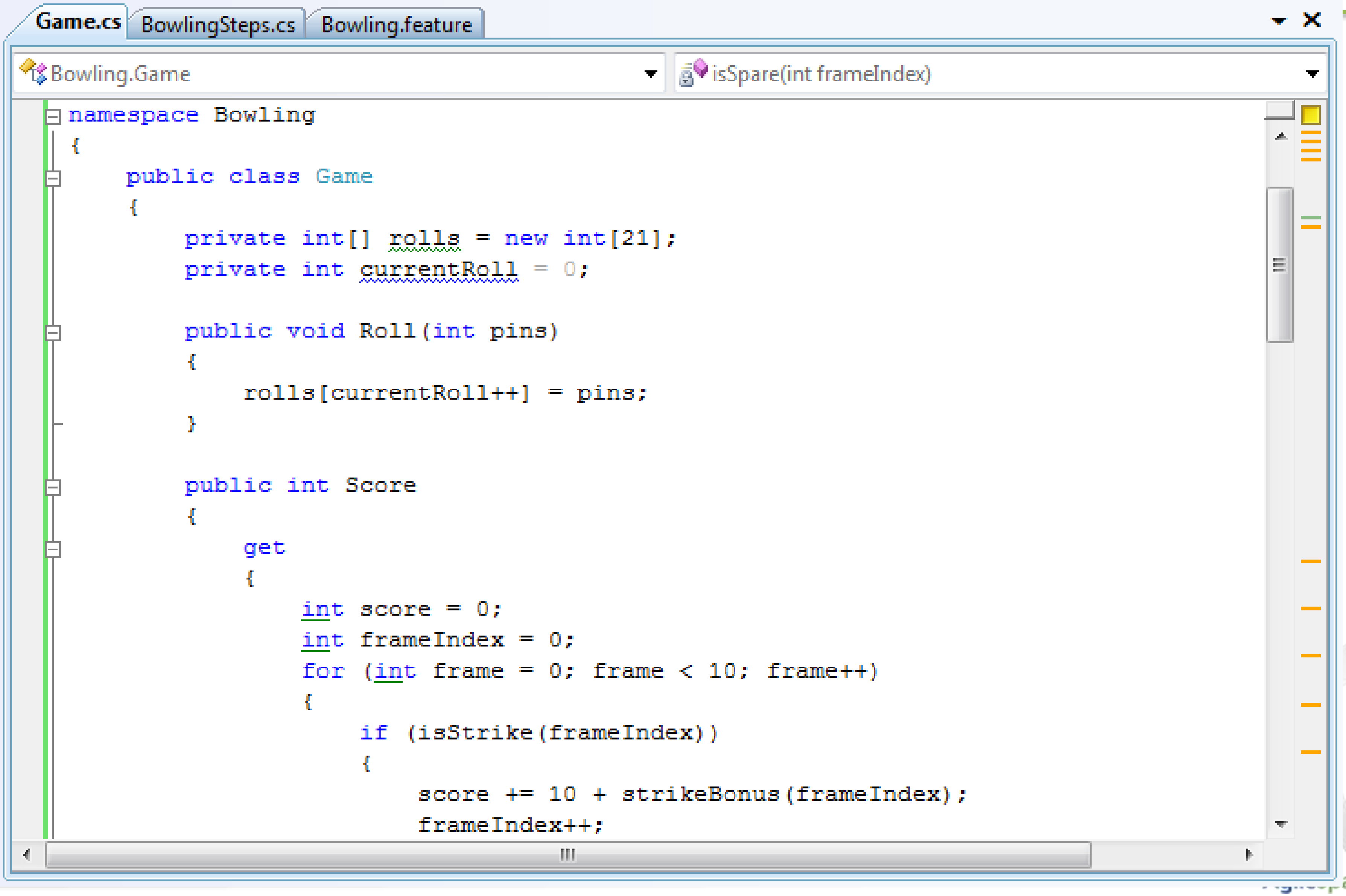
Tests failed: 2, passed: 0, ignored: 0

<Bowling.Specflow> (2 tests)
{} Bowling.Specflow (2 tests)
ScoreCalculationFixture (2 tests)
GutterGame
AllStrikes

ScoreCalculationFixture.GutterGame : Failed

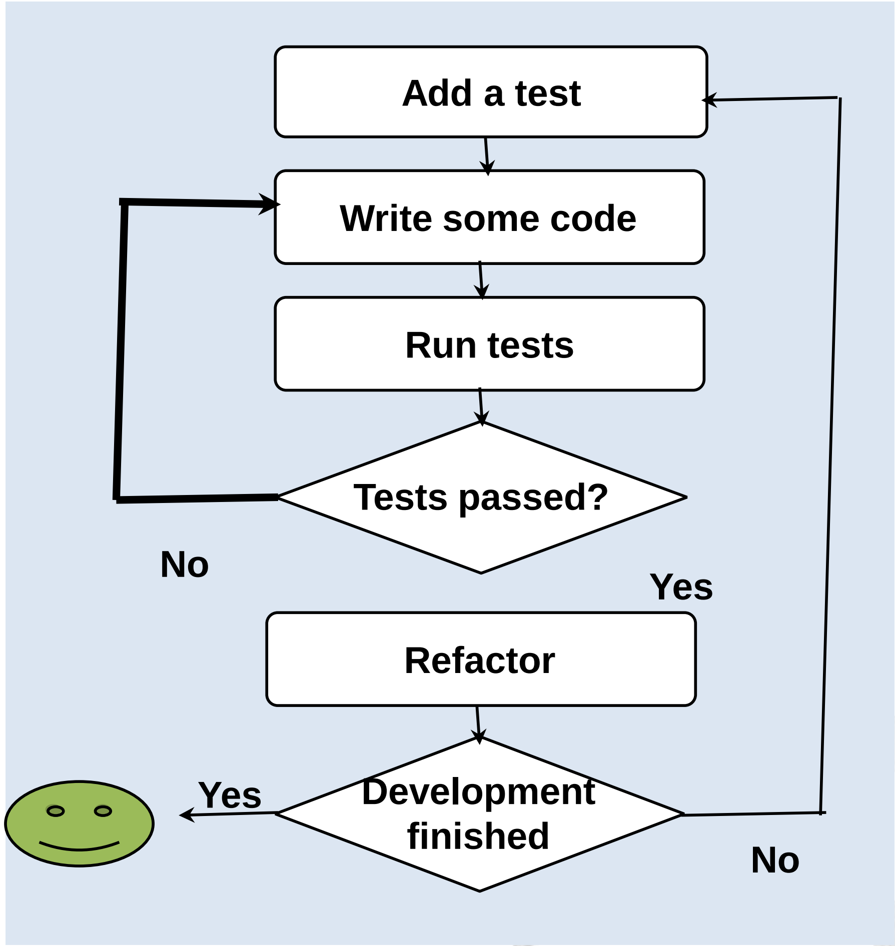
```
Given a new bowling game
When all of my balls are landing in the gutter
Then my total score should be 0
-> error:   Expected: 0
  But was:  -1
```

Step 6 - Implement Domain Functionality

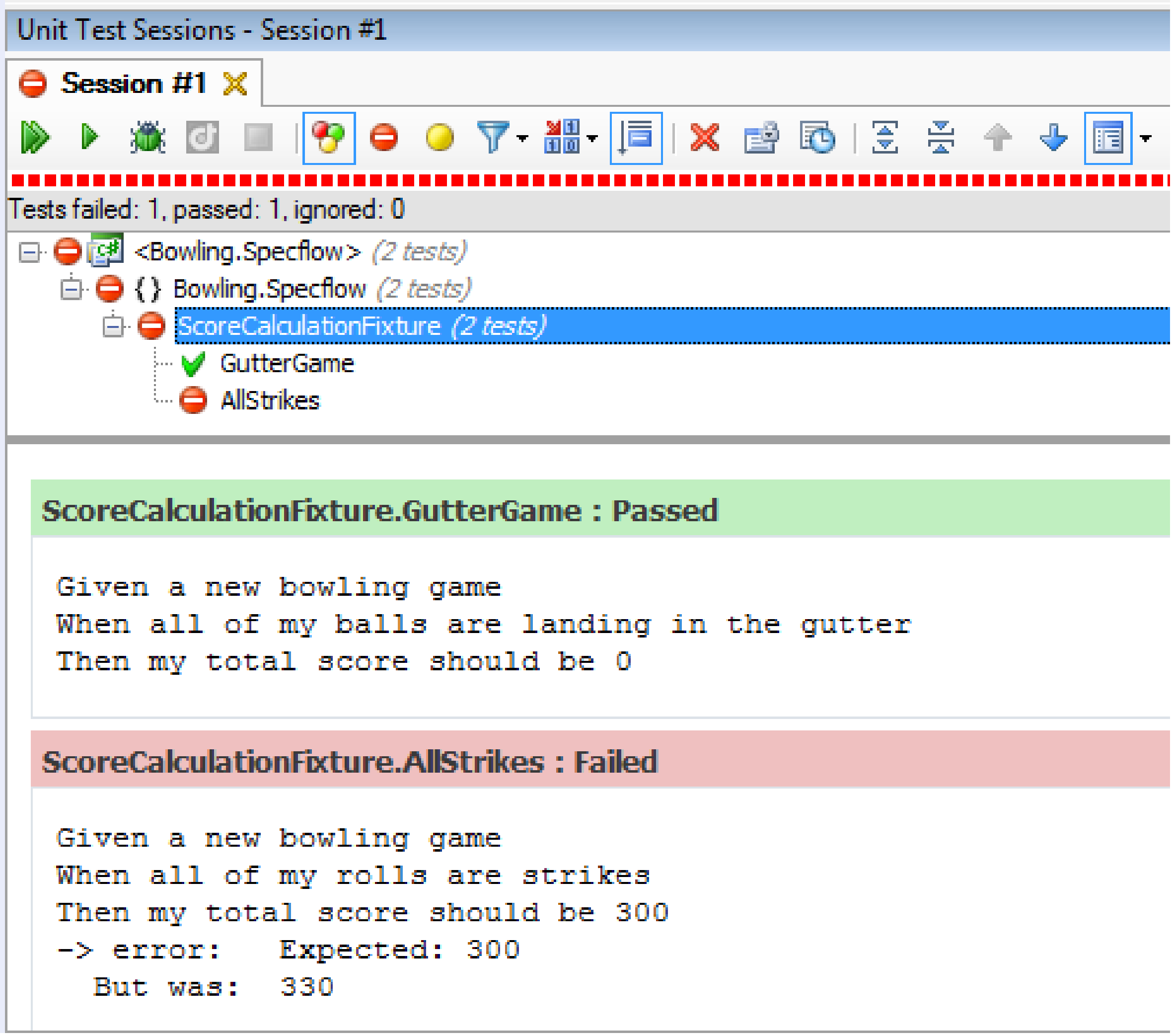


```
namespace Bowling
{
    public class Game
    {
        private int[] rolls = new int[21];
        private int currentRoll = 0;

        public void Roll(int pins)
        {
            rolls[currentRoll++] = pins;
        }

        public int Score
        {
            get
            {
                int score = 0;
                int frameIndex = 0;
                for (int frame = 0; frame < 10; frame++)
                {
                    if (isStrike(frameIndex))
                    {
                        score += 10 + strikeBonus(frameIndex);
                        frameIndex++;
```

Optional Step 6a - Unit-level TDD



Step 7 - Iterate steps 5+6 until scenario passes



Unit Test Sessions - Session #1

Session #1

Tests failed: 1, passed: 1, ignored: 0

- <Bowling.Specflow> *(2 tests)*
  - { } Bowling.Specflow *(2 tests)*
    - ScoreCalculationFixture *(2 tests)*
      - ✓ GutterGame
      - AllStrikes

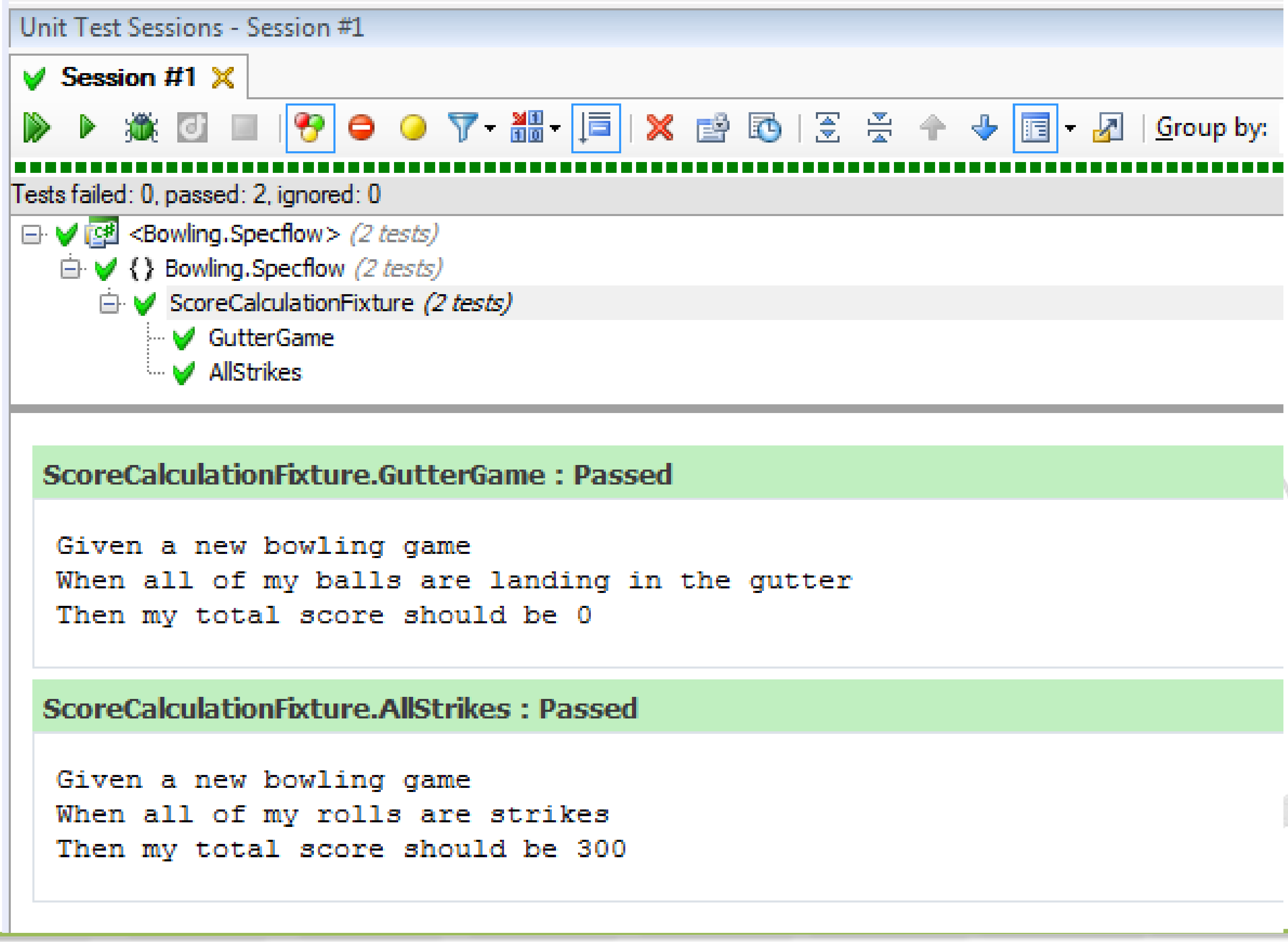
ScoreCalculationFixture.GutterGame : Passed

```
Given a new bowling game
When all of my balls are landing in the gutter
Then my total score should be 0
```

ScoreCalculationFixture.AllStrikes : Failed

```
Given a new bowling game
When all of my rolls are strikes
Then my total score should be 300
-> error:   Expected: 300
  But was:  330
```

Step 8 - Iterate steps 2-7 until Feature Passes



Tests failed: 0, passed: 2, ignored: 0

- <Bowling.Specflow> *(2 tests)*
  - Bowling.Specflow *(2 tests)*
    - ScoreCalculationFixture *(2 tests)*
      - GutterGame
      - AllStrikes

ScoreCalculationFixture.GutterGame : Passed

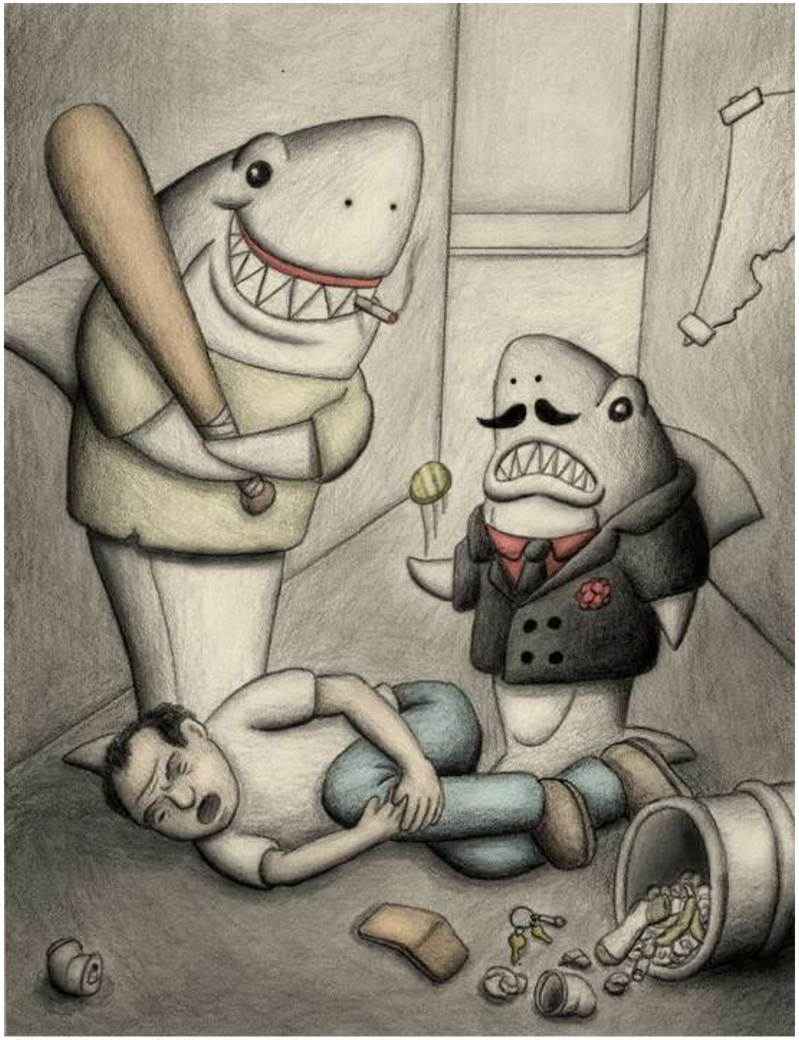
```
Given a new bowling game
When all of my balls are landing in the gutter
Then my total score should be 0
```

ScoreCalculationFixture.AllStrikes : Passed

```
Given a new bowling game
When all of my rolls are strikes
Then my total score should be 300
```

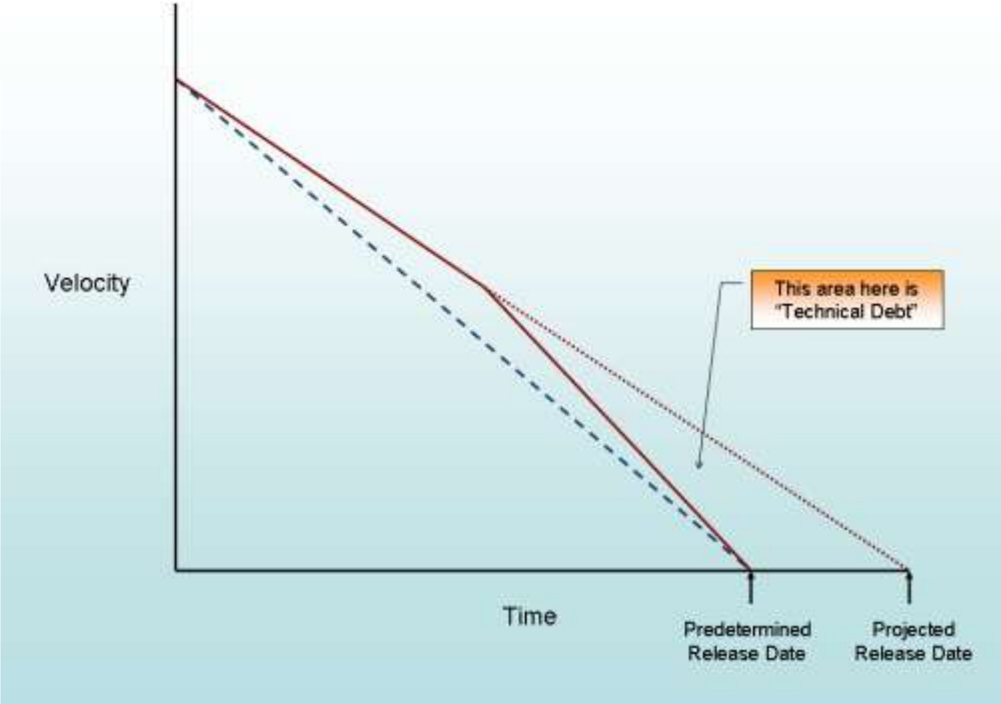
But how do we get started, we have so much legacy?

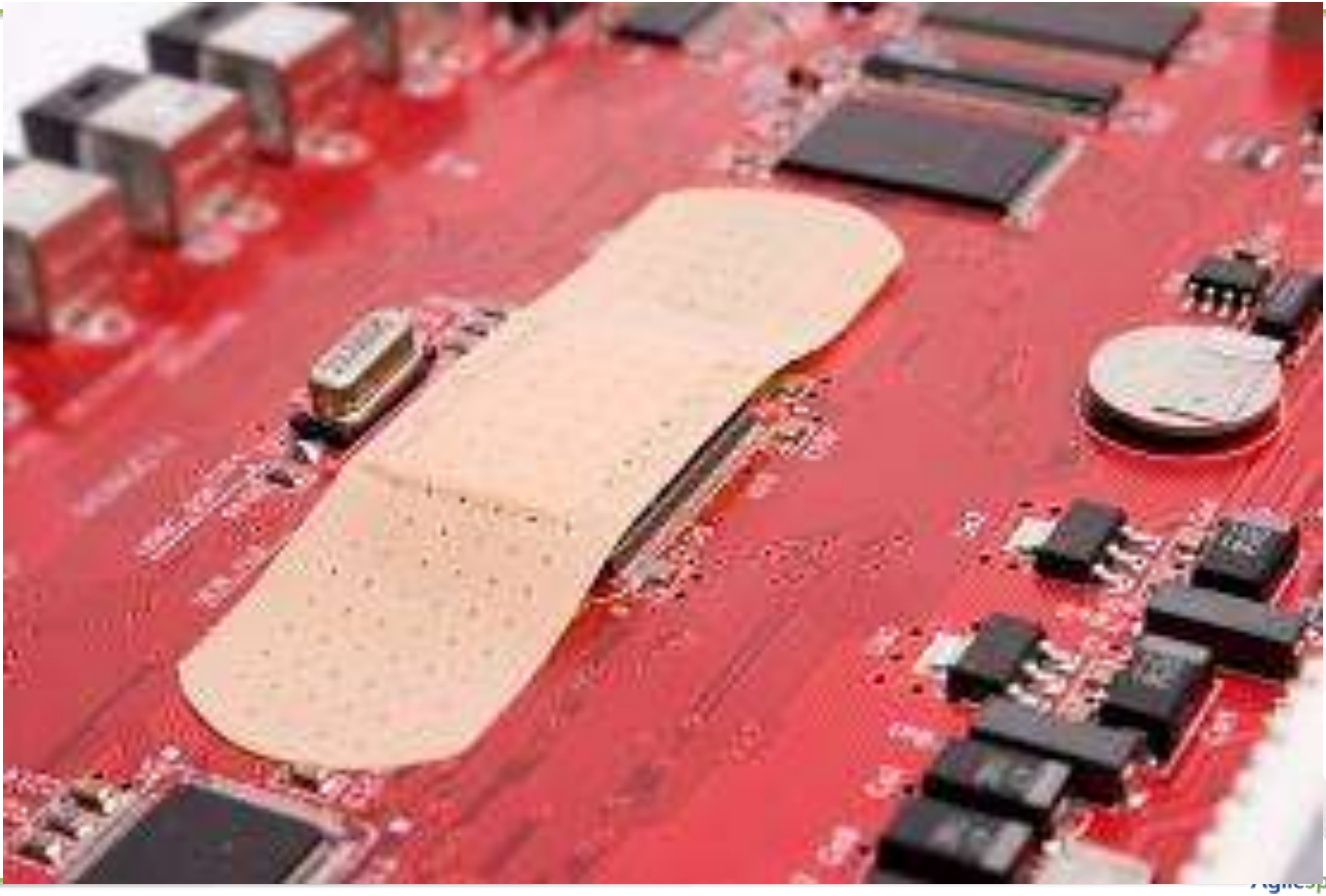
Time to talk about Technical Debt…



http://www.christrefz.com/

Chris Trefz ©2005





Check out a recent great presentation…

#LFMF

(and others who shall remain anonymous even if they don't deserve it)

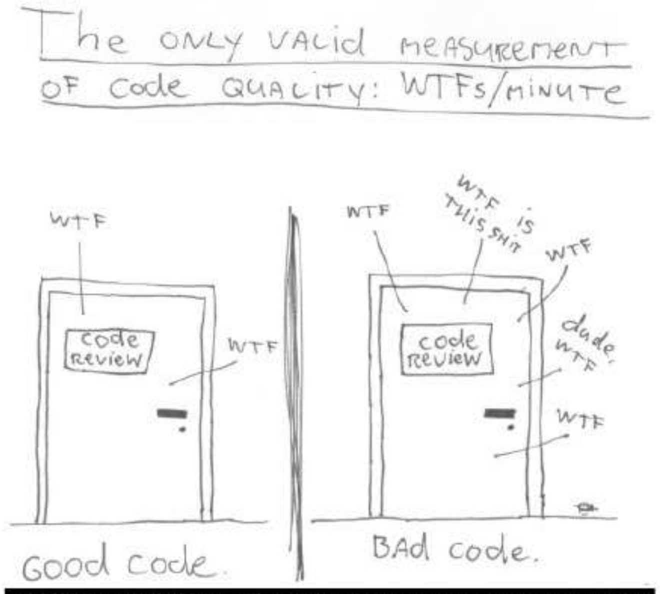
Learn from My Fail:

Tales of Test Automation Gone Wrong

(Plus a Little Bit about How to Do It Right)

Elisabeth Hendrickson
Quality Tree Software, Inc.
@testobsessed

Flooded basement image courtesy of carlpenergy
Showerhead image courtesy of deusx
Images from Flickr licensed through Creative Commons Attribution

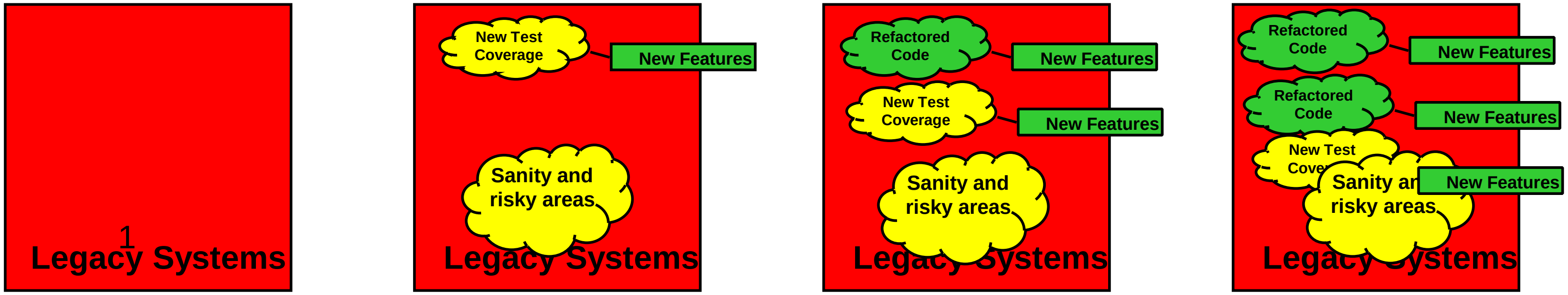


(c) 2008 Focus Shift/OSNews/Thom Holwerda · http://www.osnews.com/comics

Copyright © 2012 Elisabeth Hendrickson, Quality Tree Software, Inc.

http://www.slideshare.net/ehendrickson/lfmf-tales-of-test-automation-fa

How to Start



🟥 Frightening code

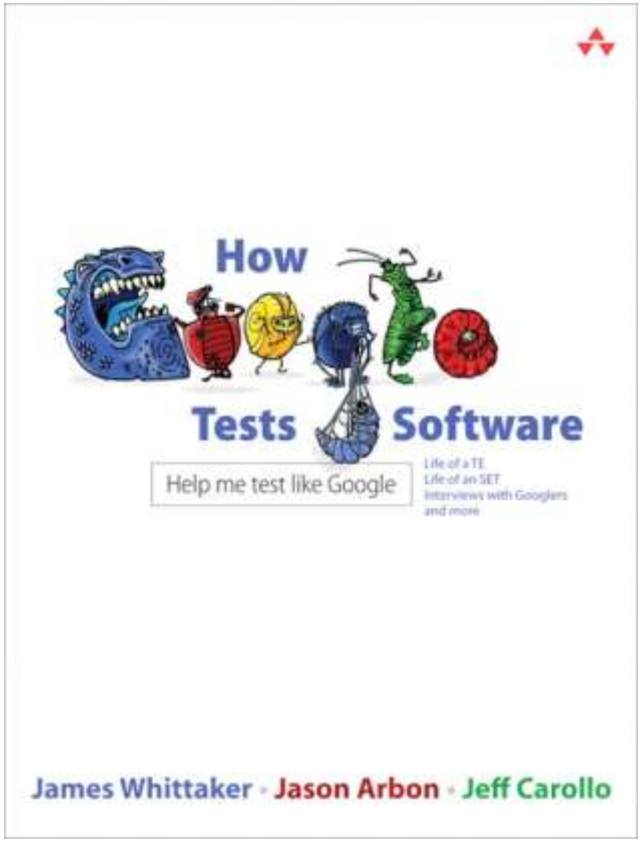
🟨 Low quality code covered by automated tests

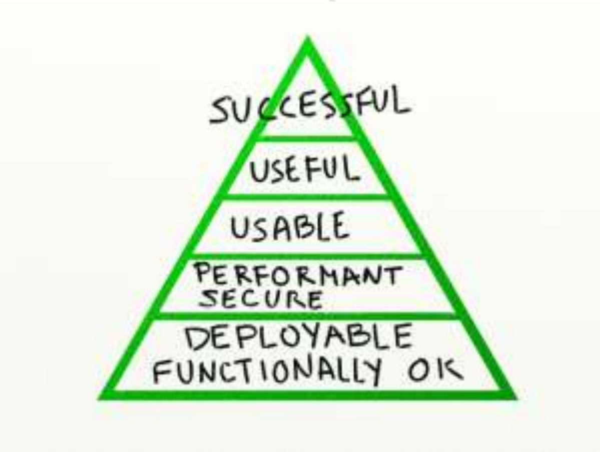
🟩 High quality code covered by automated tests

- Apply automation to incrementally repair touch points as new features added.
- Manual affected regression testing only after risk analysis.
- Automate Sanity and risky areas by independent team

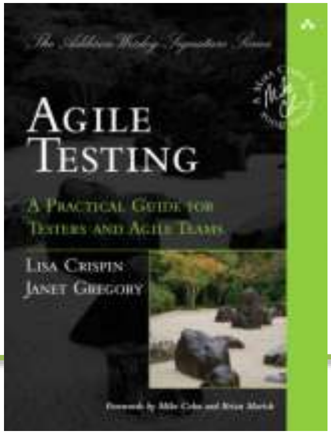
References

http://bit.ly/testisdeadGTAC11





http://gojko.net/2012/05/08/redefining-software-quality/


The Addison-Wesley Signature Series
AGILE TESTING
A PRACTICAL GUIDE FOR TESTERS AND AGILE TEAMS
LISA CRISPIN
JANET GREGORY