Agile Without Dedicated QA

James Shore

TWITTER: @jamesshore
EMAIL: jshore@jamesshore.com
WEB: jamesshore.com
GITHUB: github.com/jamesshore

Pacific Northwest Software Quality Conference
October 15, 2019

The Art of

Agile Development

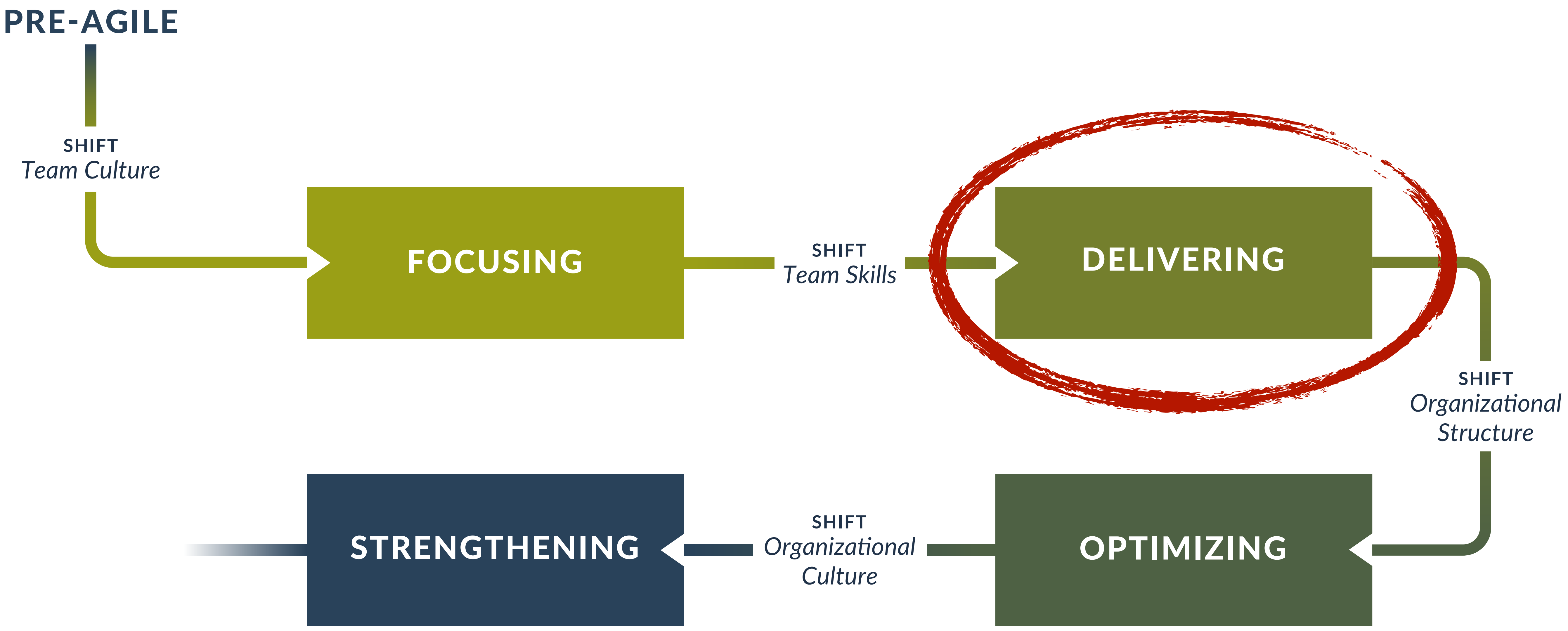
/Theory/In/Practice

O'REILLY®

James Shore & Shane Warden

THE AGILE FLUENCY™ MODEL

CHART YOUR AGILE PATHWAY



PRE-AGILE

SHIFT
Team Culture

FOCUSING

SHIFT
Team Skills

DELIVERING

SHIFT
Organizational Structure

OPTIMIZING

SHIFT
Organizational Culture

STRENGTHENING

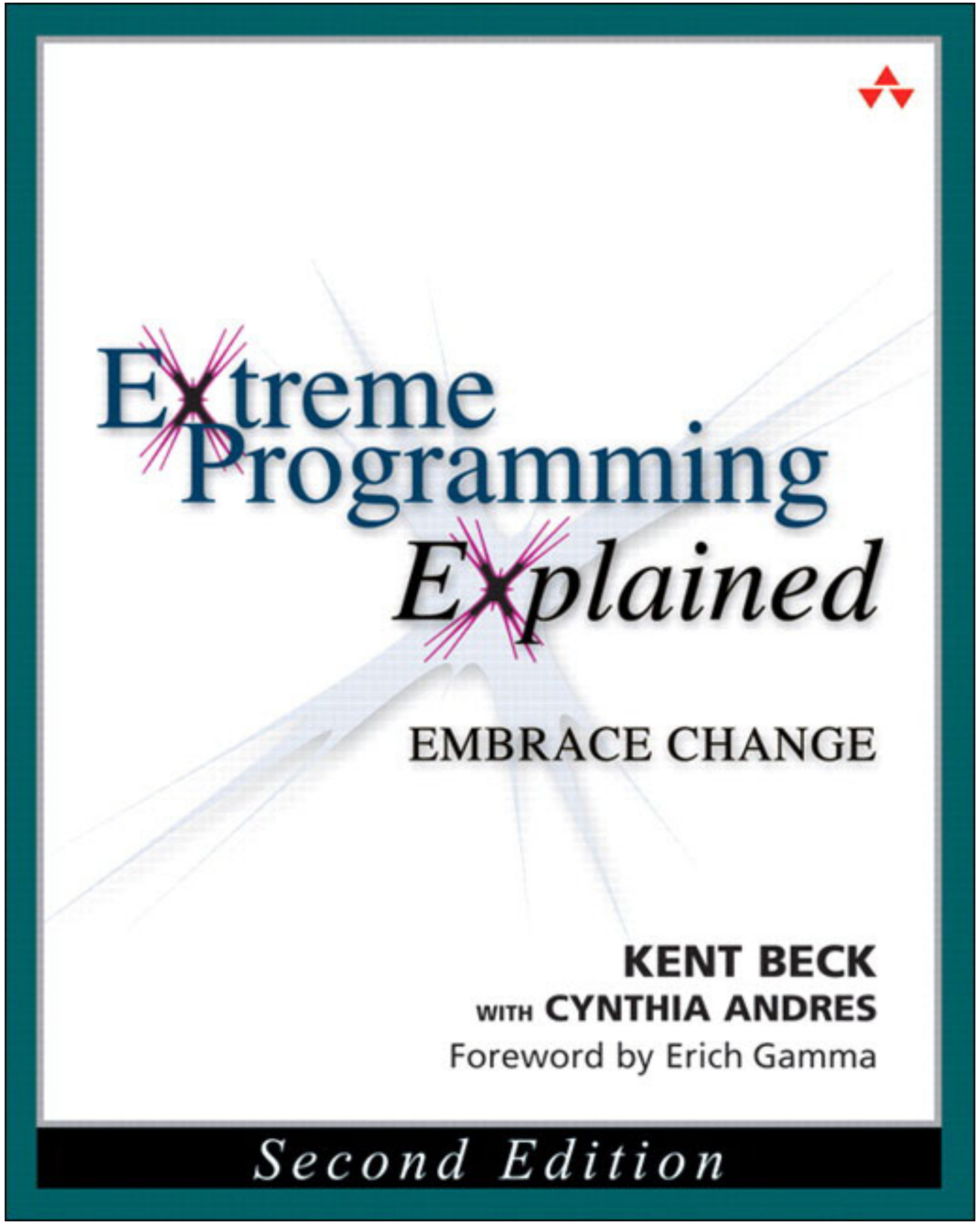
AGILE FLUENCY PROJECT

agilefluency.org

AN BO Y BUI

Enter the name of the TestCase class:
com.novell.devnet._TestAll$DeveloperSuite

Runs: 159 Errors: 0 Failures: 0



Extreme Programming Explained

EMBRACE CHANGE

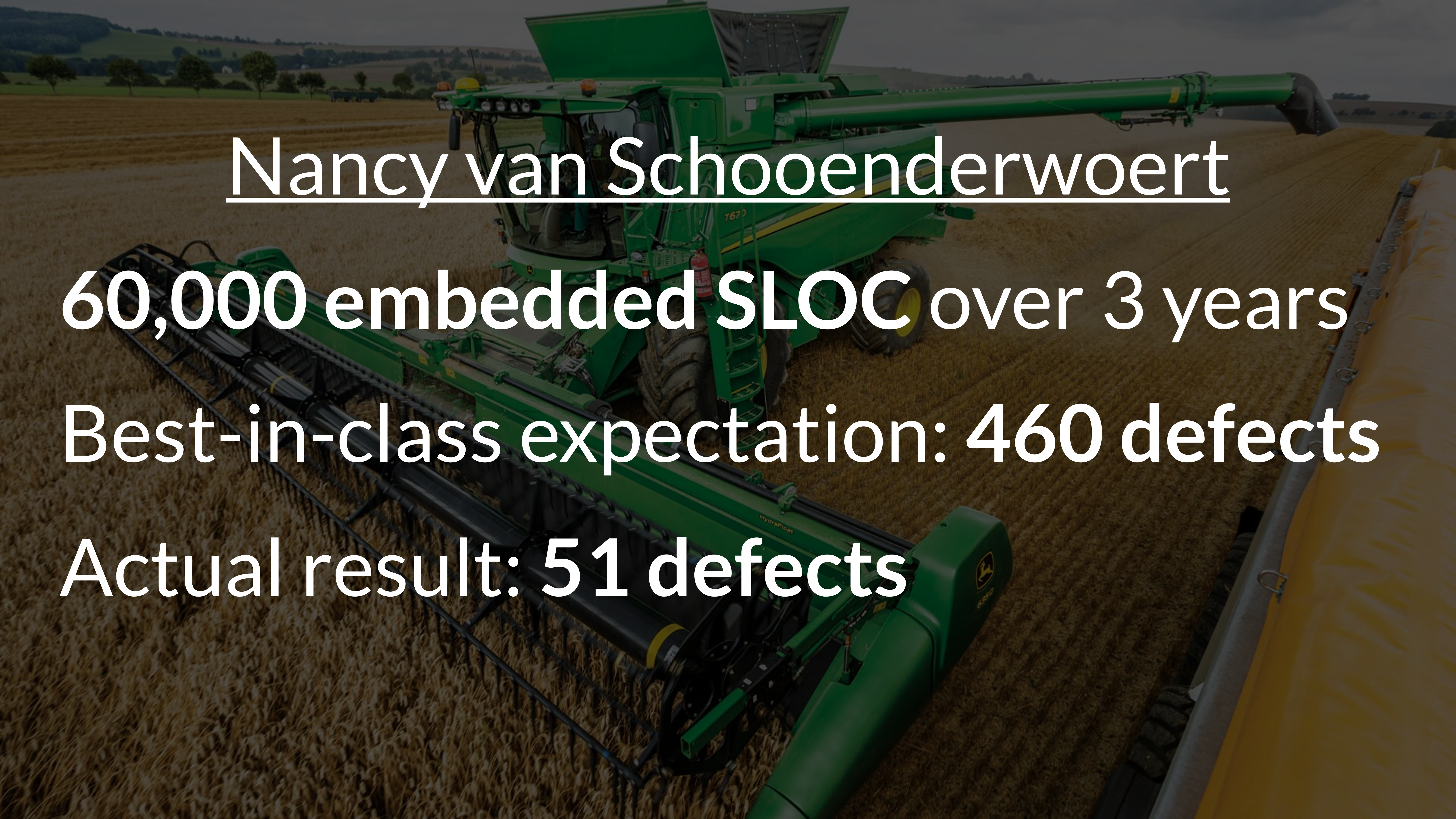
KENT BECK

WITH CYNTHIA ANDRES

Foreword by Erich Gamma

Second Edition

EXTREME PROGRAMMING
I CAN'T GIVE YOU ALL OF THESE FEATURES IN THE FIRST VERSION.
scottadams@aol.com
www.dilbert.com
AND EACH FEATURE NEEDS TO HAVE WHAT WE CALL A "USER STORY."
© 2002 United Feature Syndicate, Inc.
1/10/03
OKAY, HERE'S A STORY: YOU GIVE ME ALL OF MY FEATURES OR I'LL RUIN YOUR LIFE.



Nancy van Schooenderwoert

60,000 embedded SLOC over 3 years

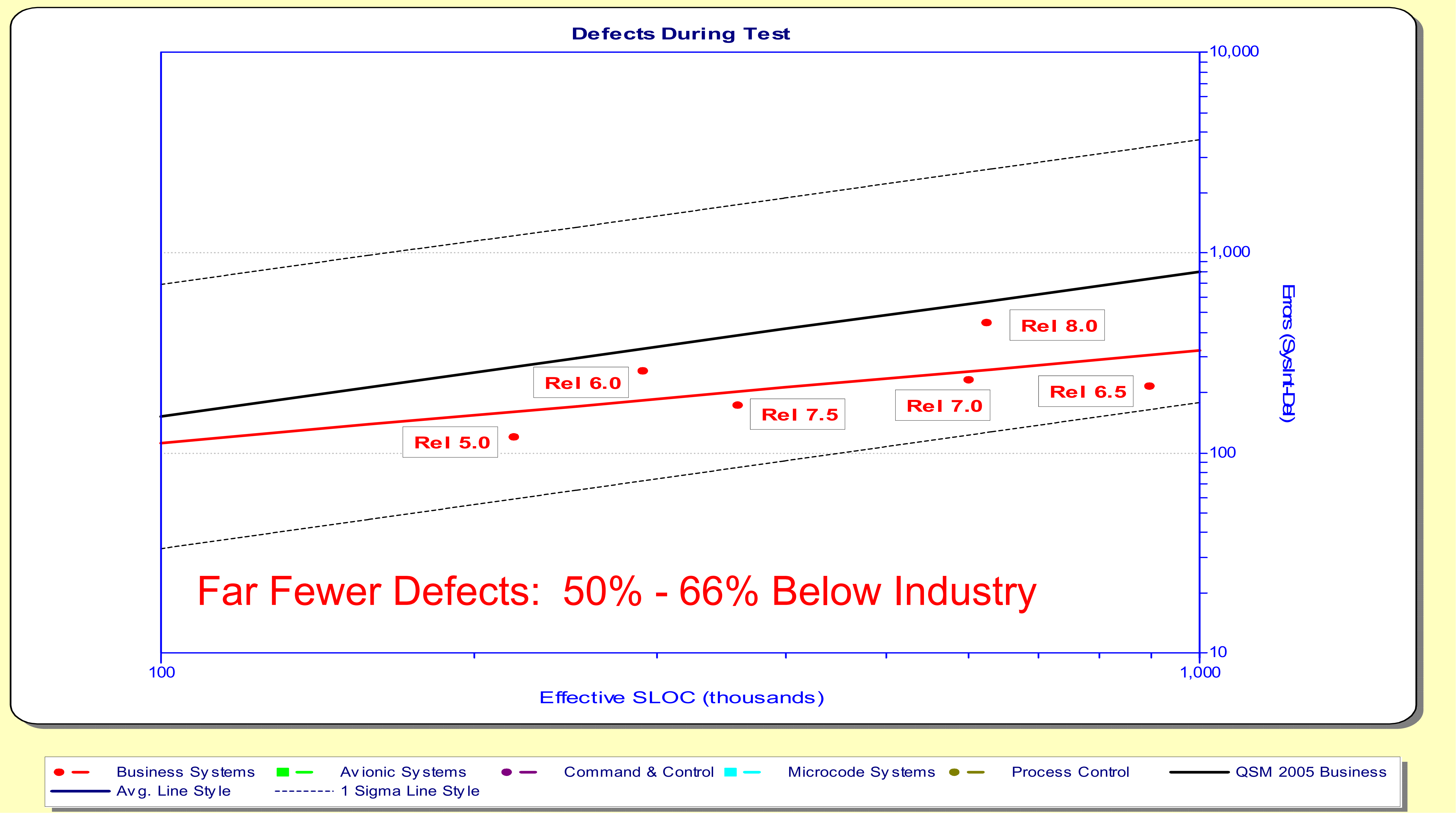
Best-in-class expectation: **460 defects**

Actual result: **51 defects**

QSM ASSOCIATES

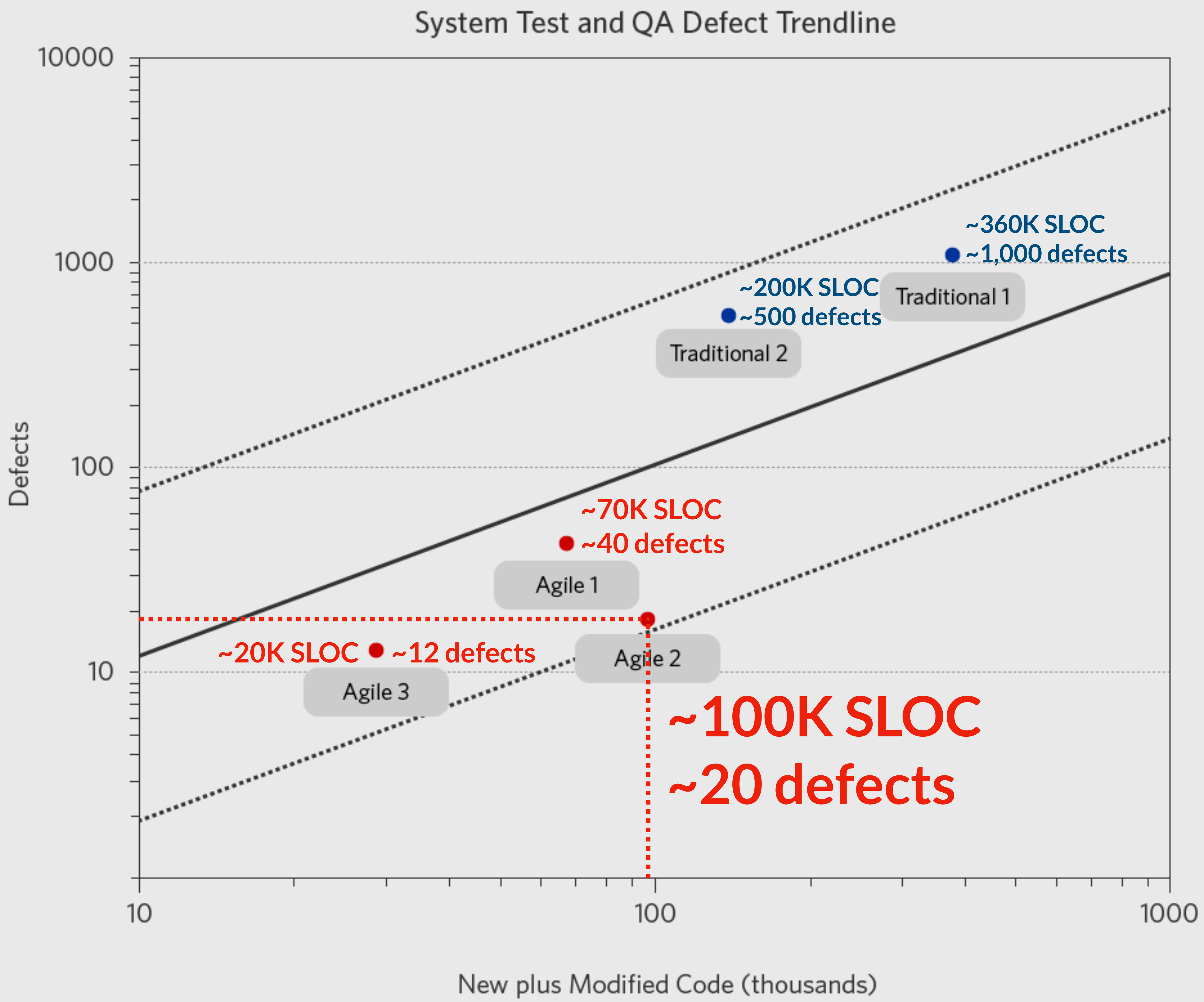
Project Management
That Learns From Experience

Trendline Assessment – Defects/Quality



Far Fewer Defects: 50% - 66% Below Industry

Excerpted from Michael Mah PNSQC 2010 presentation:
“The Good, the Bad, and the Puzzling: The Agile Experience at 5 Companies”



~360K SLOC
~1,000 defects

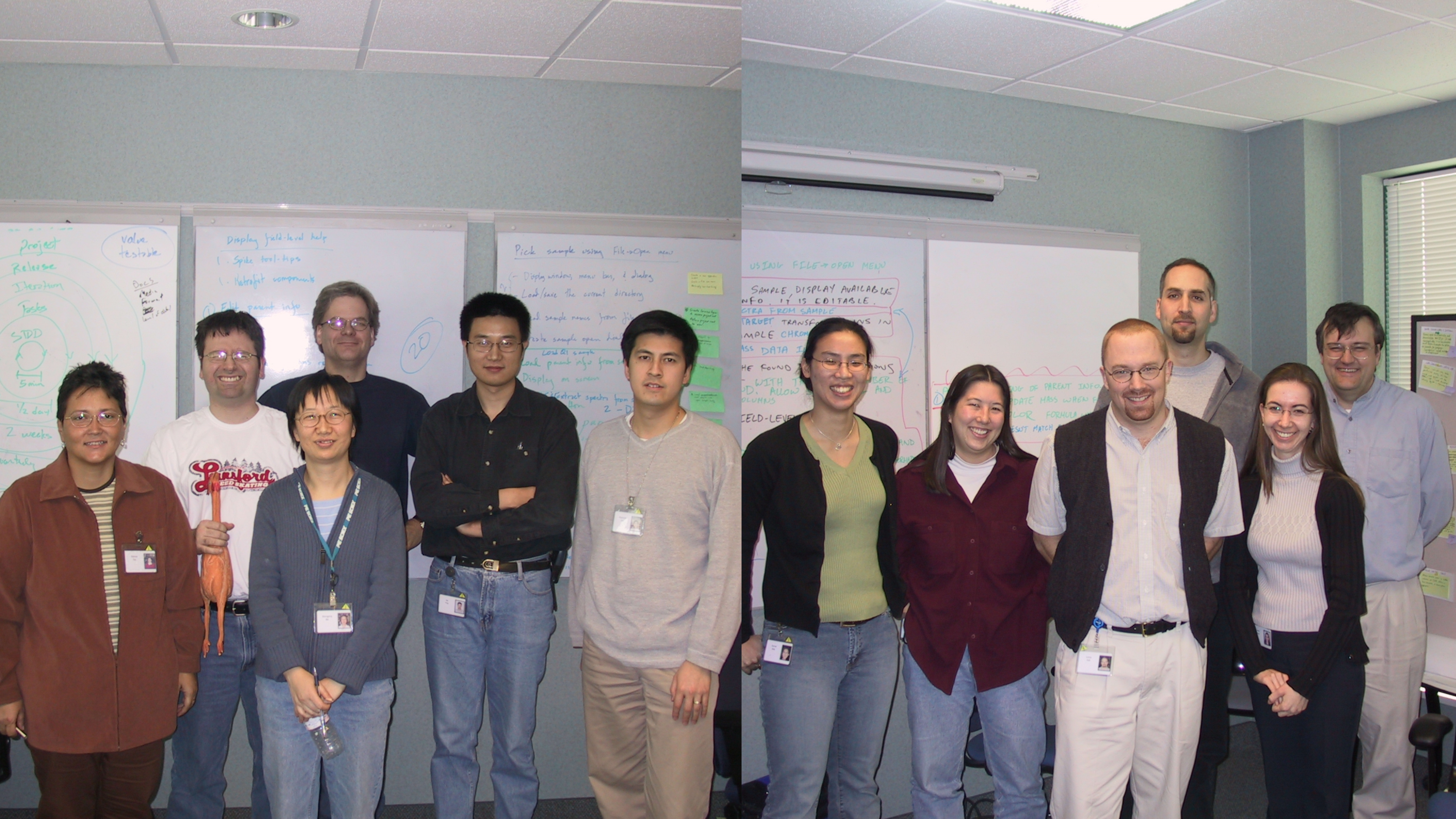
~200K SLOC
~500 defects

~70K SLOC
~40 defects

~20K SLOC ~12 defects

~100K SLOC
~20 defects

https://www.industriallogic.com/an-agile-xp-transition-continues-to-thrive/


Project
Release
Iteration
Tasks
STDD
5 min
1/2 day
2 weeks
Value testable
Docs
Display field-level help
1. Spike tool-tips
1. Retrofit components
Edit parent info
Pick sample using File→Open menu
Display window, menu bar, & dialog
Load/save the current directory
USING FILE→OPEN MENU
SAMPLE, DISPLAY AVAILABLE
INFO. IT IS EDITABLE.
CTRA FROM SAMPLE
ASS DATA
FIELD-LEVEL

Programmer Errors

Defect-Prone Designs

Requirements Misunderstandings

Systemic Blind Spots

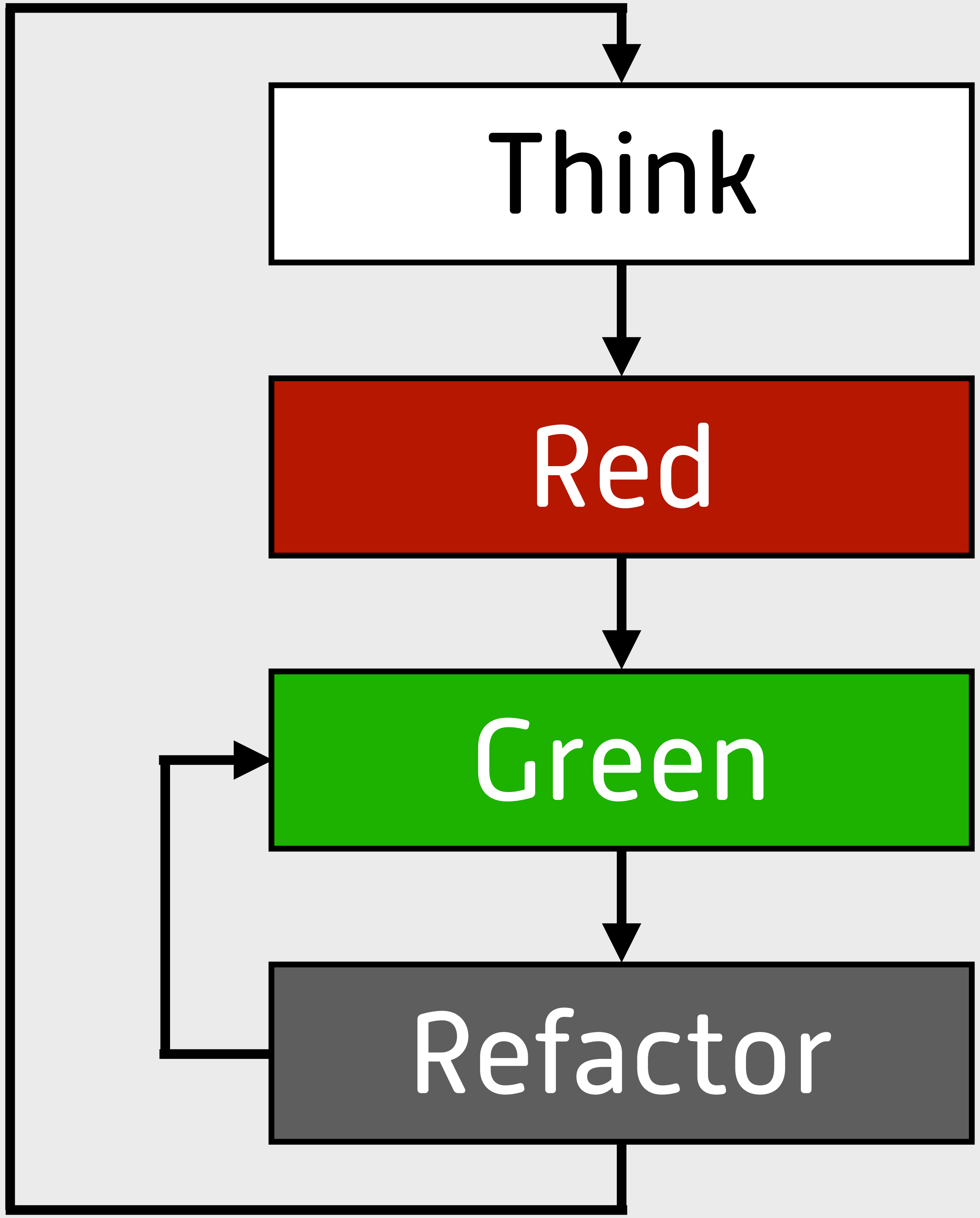
Programmer Errors

Guessing Game v1

- Person 1: Think of a whole number between 1 and 100.
- Person 2: Make **four different guesses** of the number, each at least 5 digits apart.
  🚫 **51, 52, 53, 54** ✅ **51, 56, 61, 66**
- Person 1: Say how many guesses were high, low, or right on, **but don’t say which guess is which.**
- **Repeat,** four guesses at a time, until you’ve guessed the number, then switch.

Guessing Game v2

- Person 1: Think of a whole number between 1 and 100.
- Person 2: Make **one guess** of the number.
- Person 1: Say if the guess was high, low, or right on.
- **Repeat,** one guess at a time, until you’ve guessed the number, then switch.

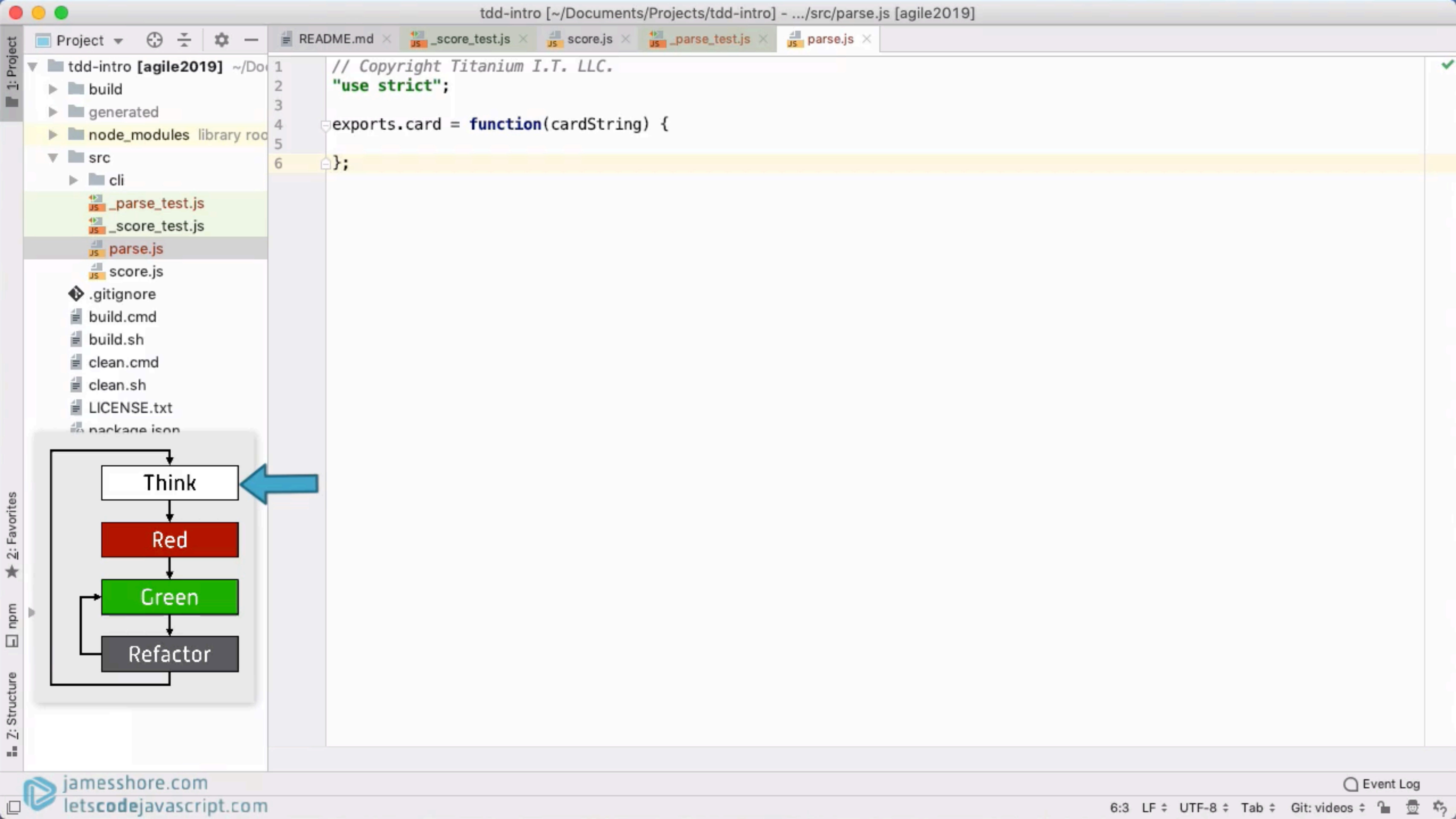


Think

Red

Green

Refactor



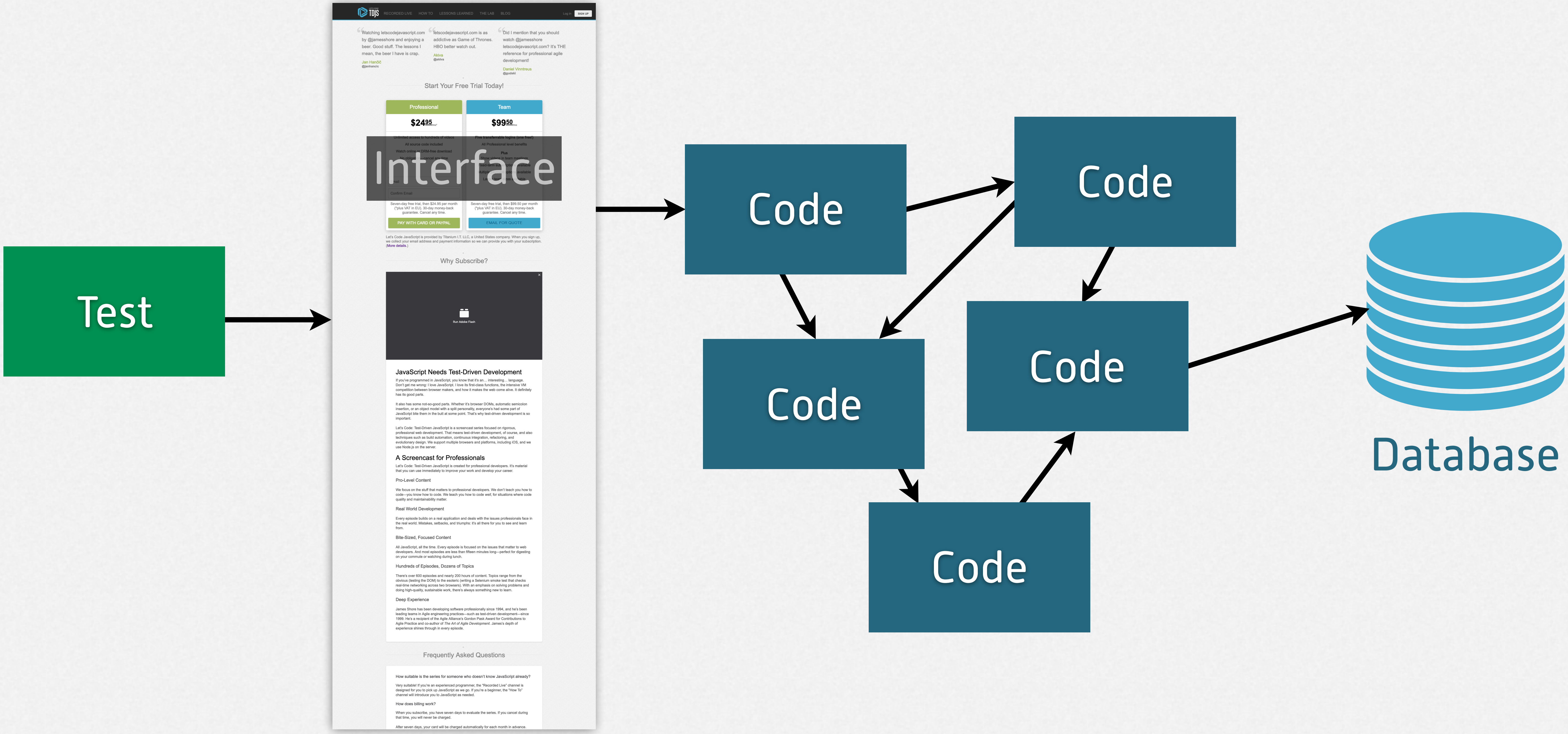
```
// Copyright Titanium I.T. LLC.
"use strict";

exports.card = function(cardString) {

};
```

jamesshore.com
letscodejavascript.com

Why TDD Works Better

- **Better Tests:** Work is fine-grained, covering more edge cases.
- **Improved Self-Discipline:** It’s easier to write tests as you go, and there’s less temptation to move on to the next thing.
- **Fast Feedback:** TDD is a series of small, validated hypotheses.

End-to-End Tests



Test

Interface

Code

Code

Code

Code

Code

Database

LET'S CODE TDJS RECORDED LIVE HOW TO LESSONS LEARNED THE LAB BLOG Log in SIGN UP

"Watching letscodejavascript.com by @jamesshore and enjoying a beer. Good stuff. The lessons I mean, the beer I have is crap.

Jan Hančič
@janhancic

"letscodejavascript.com is as addictive as Game of Thrones. HBO better watch out.

Akiva
@akiva

"Did I mention that you should watch @jamesshore letscodejavascript.com? It's THE reference for professional agile development!

Daniel Vinntreus
@gudski

Start Your Free Trial Today!

Professional	Team
$24.95/mo*	$99.50/mo
Unlimited access to hundreds of videos All source code included Watch online or DRM-free download No obligation—cancel any time	**Five transferrable logins (one free!)** All Professional level benefits Plus Show videos in team meetings Fixed-term subscriptions available Multiple payment options available Larger team sizes available
Email Confirm Email	
Seven-day free trial, then $24.95 per month (*plus VAT in EU). 30-day money-back guarantee. Cancel any time.	Seven-day free trial, then $99.50 per month (*plus VAT in EU). 30-day money-back guarantee. Cancel any time.
PAY WITH CARD OR PAYPAL	EMAIL FOR QUOTE

Let's Code JavaScript is provided by Titanium I.T. LLC, a United States company. When you sign up, we collect your email address and payment information so we can provide you with your subscription. (More details.)

Why Subscribe?

Run Adobe Flash

JavaScript Needs Test-Driven Development

If you've programmed in JavaScript, you know that it's an… interesting… language. Don't get me wrong: I love JavaScript. I love its first-class functions, the intensive VM competition between browser makers, and how it makes the web come alive. It definitely has its good parts.

It also has some not-so-good parts. Whether it's browser DOMs, automatic semicolon insertion, or an object model with a split personality, everyone's had some part of JavaScript bite them in the butt at some point. That's why test-driven development is so important.

Let's Code: Test-Driven JavaScript is a screencast series focused on rigorous, professional web development. That means test-driven development, of course, and also techniques such as build automation, continuous integration, refactoring, and evolutionary design. We support multiple browsers and platforms, including iOS, and we use Node.js on the server.

A Screencast for Professionals

Let's Code: Test-Driven JavaScript is created for professional developers. It's material that you can use immediately to improve your work and develop your career.

Pro-Level Content

We focus on the stuff that matters to professional developers. We don't teach you how to code—you know how to code. We teach you how to code *well*, for situations where code quality and maintainability matter.

Real World Development

Every episode builds on a real application and deals with the issues professionals face in the real world. Mistakes, setbacks, and triumphs: it's all there for you to see and learn from.

Bite-Sized, Focused Content

All JavaScript, all the time. Every episode is focused on the issues that matter to web developers. And most episodes are less than fifteen minutes long—perfect for digesting on your commute or watching during lunch.

Hundreds of Episodes, Dozens of Topics

There's over 600 episodes and nearly 200 hours of content. Topics range from the obvious (testing the DOM) to the esoteric (writing a Selenium smoke test that checks real-time networking across two browsers). With an emphasis on solving problems and doing high-quality, sustainable work, there's always something new to learn.

Deep Experience

James Shore has been developing software professionally since 1994, and he's been leading teams in Agile engineering practices—such as test-driven development—since 1999. He's a recipient of the Agile Alliance's Gordon Pask Award for Contributions to Agile Practice and co-author of *The Art of Agile Development*. James's depth of experience shines through in every episode.

Frequently Asked Questions

How suitable is the series for someone who doesn't know JavaScript already?

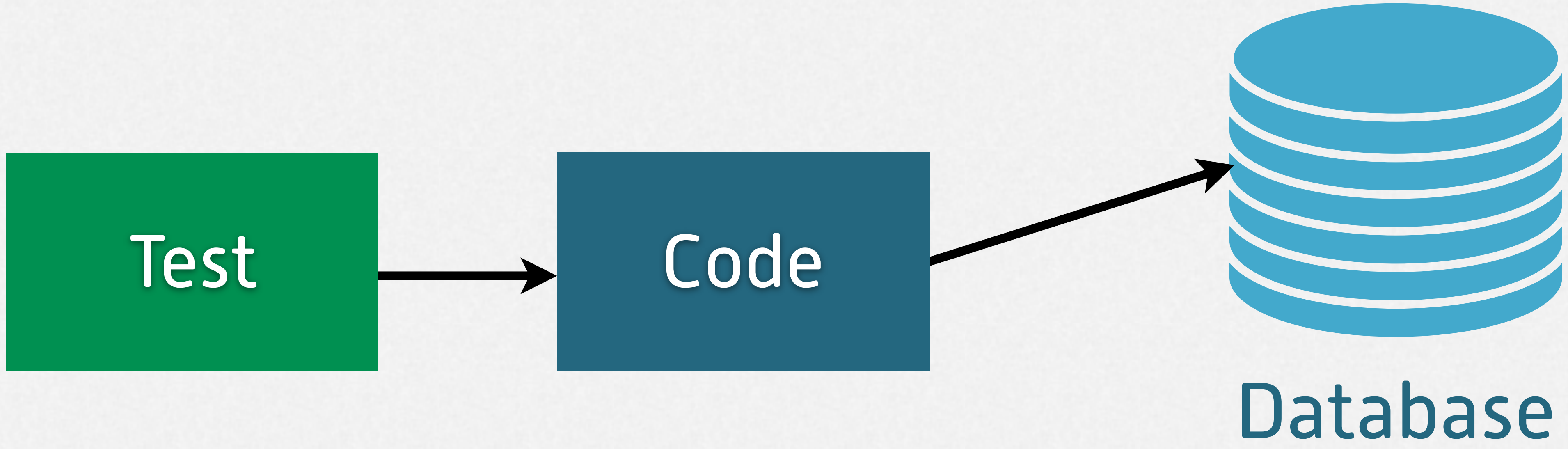
Very suitable! If you're an experienced programmer, the "Recorded Live" channel is designed for you to pick up JavaScript as we go. If you're a beginner, the "How To" channel will introduce you to JavaScript as needed.

How does billing work?

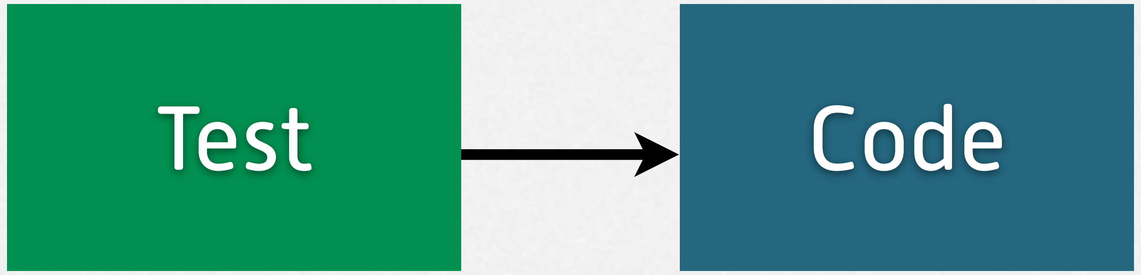
When you subscribe, you have seven days to evaluate the series. If you cancel during that time, you will never be charged.

After seven days, your card will be charged automatically for each month in advance.

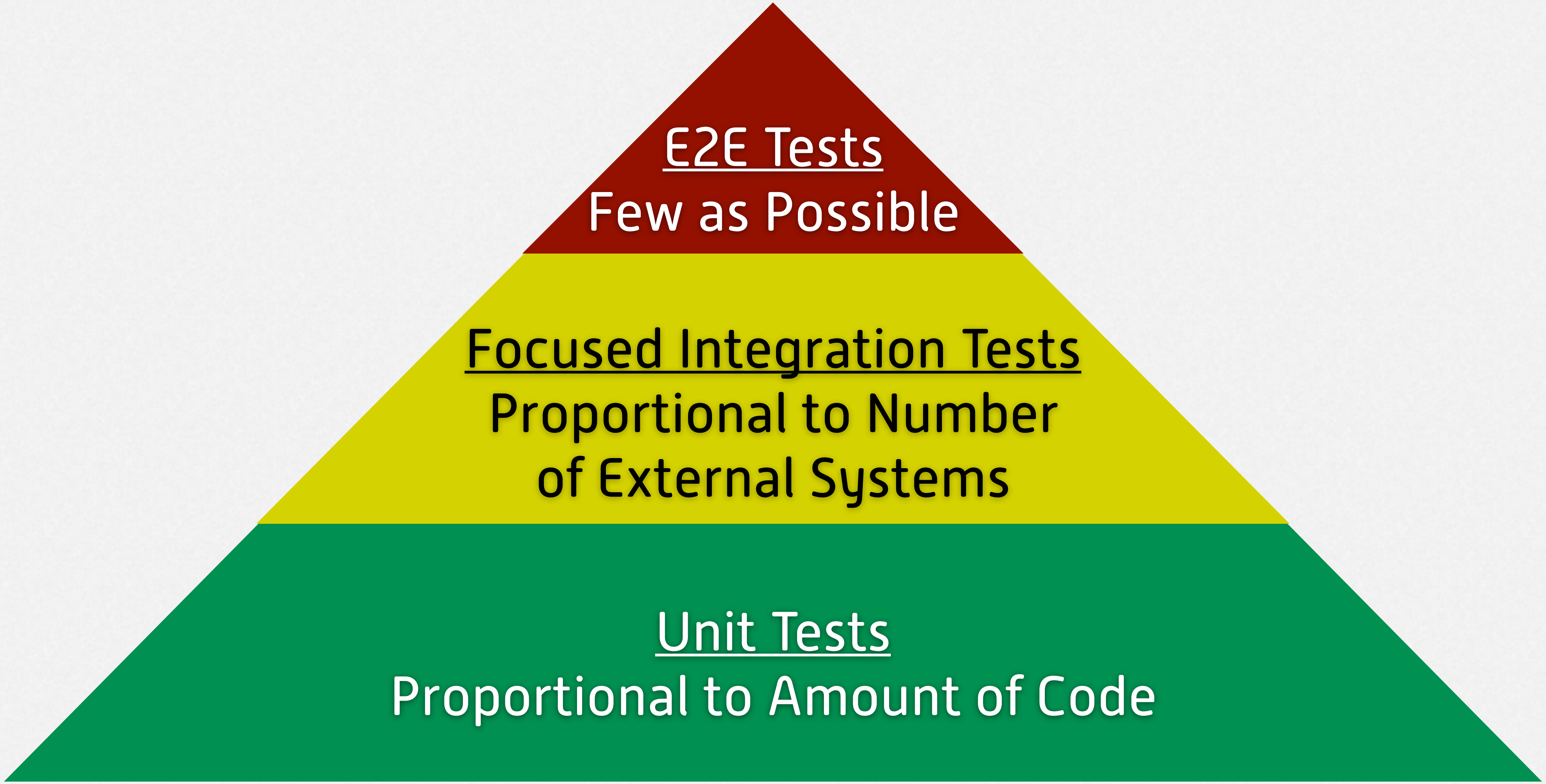
Focused Integration Tests



Unit Tests



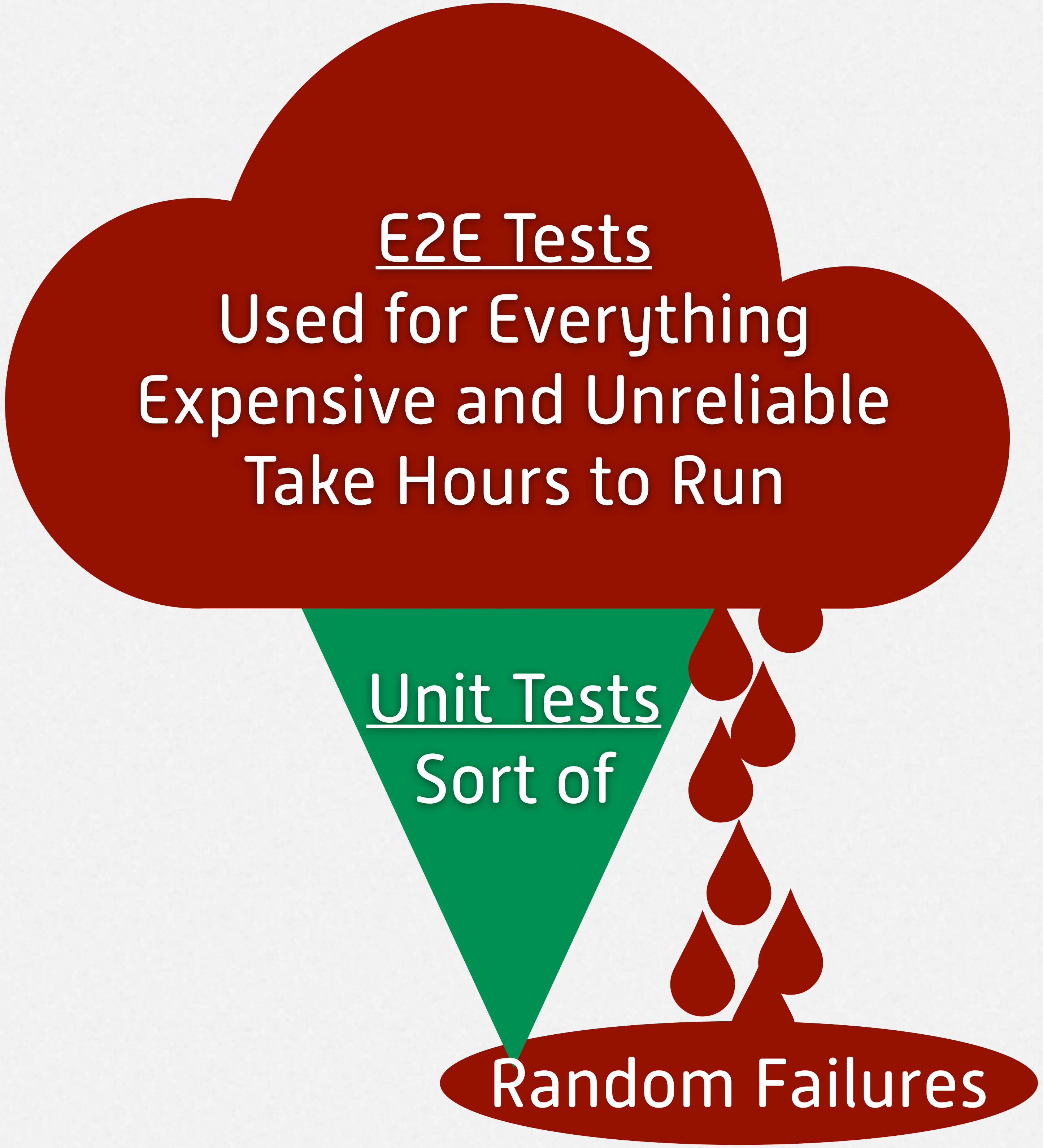
Follow the Test Pyramid


E2E Tests
Few as Possible
Focused Integration Tests
Proportional to Number of External Systems
Unit Tests
Proportional to Amount of Code

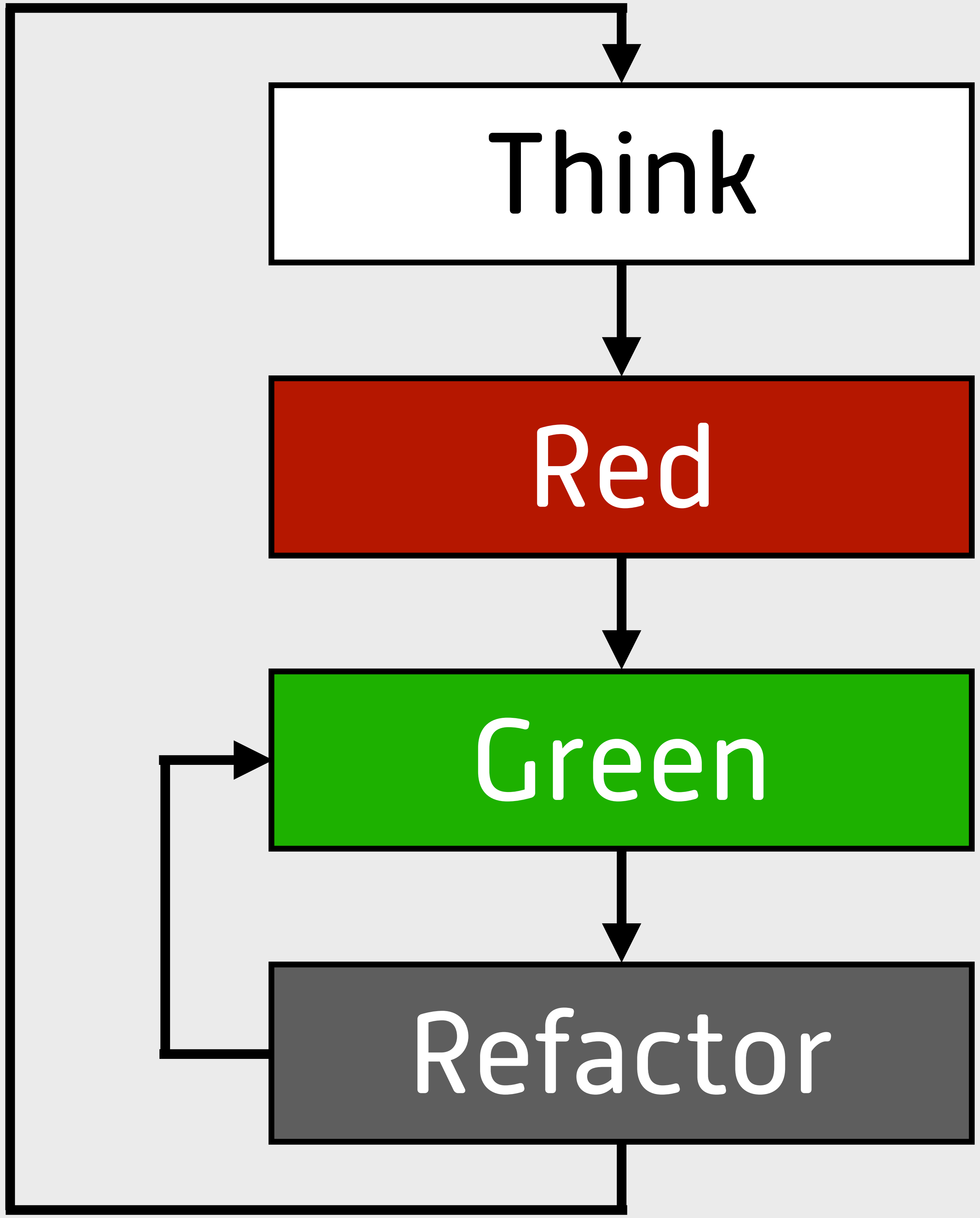
Test Pyramid originally conceived by Mike Cohn with Lisa Crispin, 2004.
Adapted by James Shore, 2019.

jamesshore.com

Beware the Test Ice Cream Cone



Test Ice Cream Cone originally conceived by Alister Scott, 2012.
Adapted by James Shore, 2019.



Think
Red
Green
Refactor

Prevent Programmer Errors

Test-Driven Development

Pairing or Mobbing

Energized Work

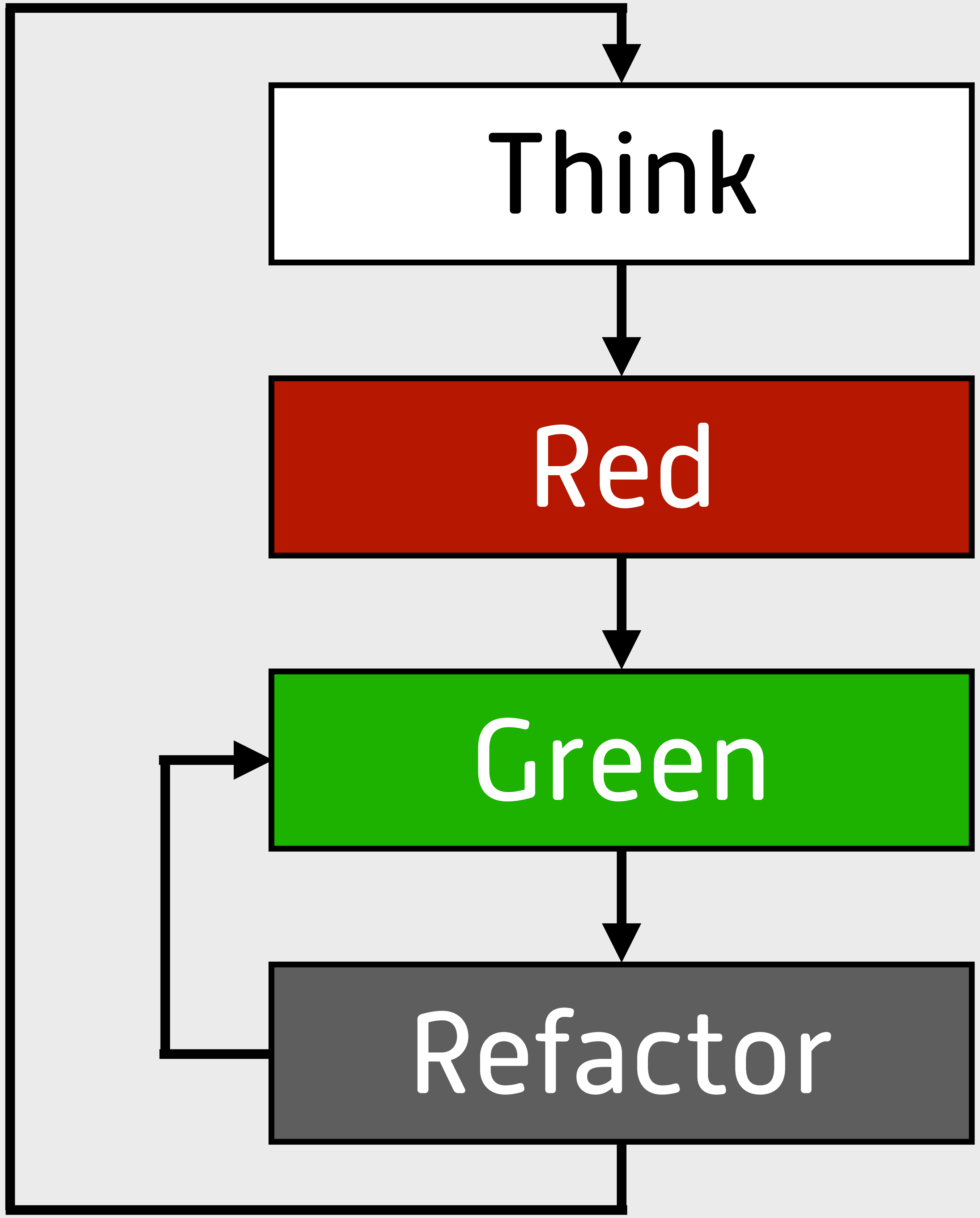
Programmer Errors

Defect-Prone Designs

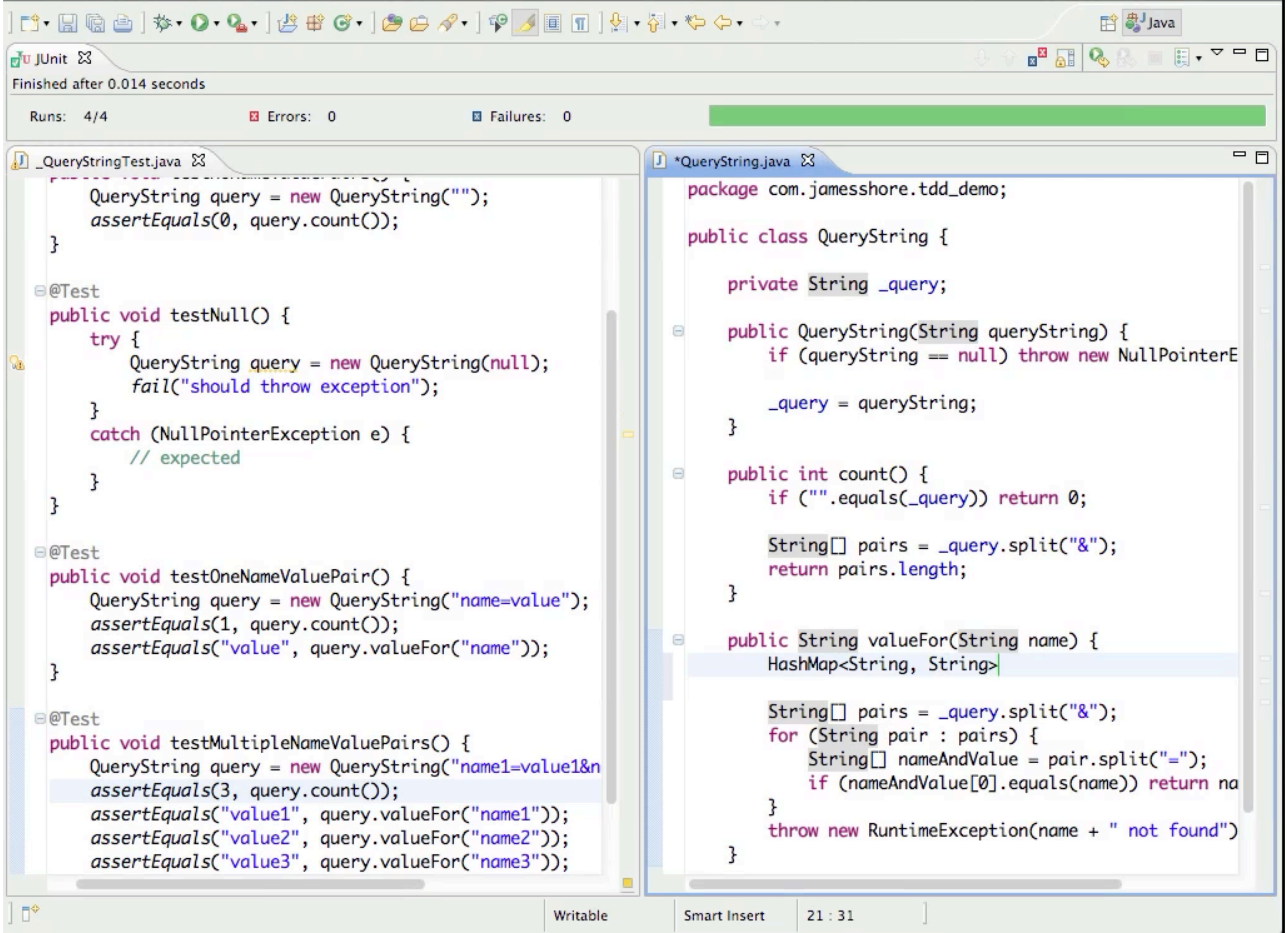
Requirements Misunderstandings

Systemic Blind Spots

Defect-Prone Designs



Think
Red
Green
Refactor

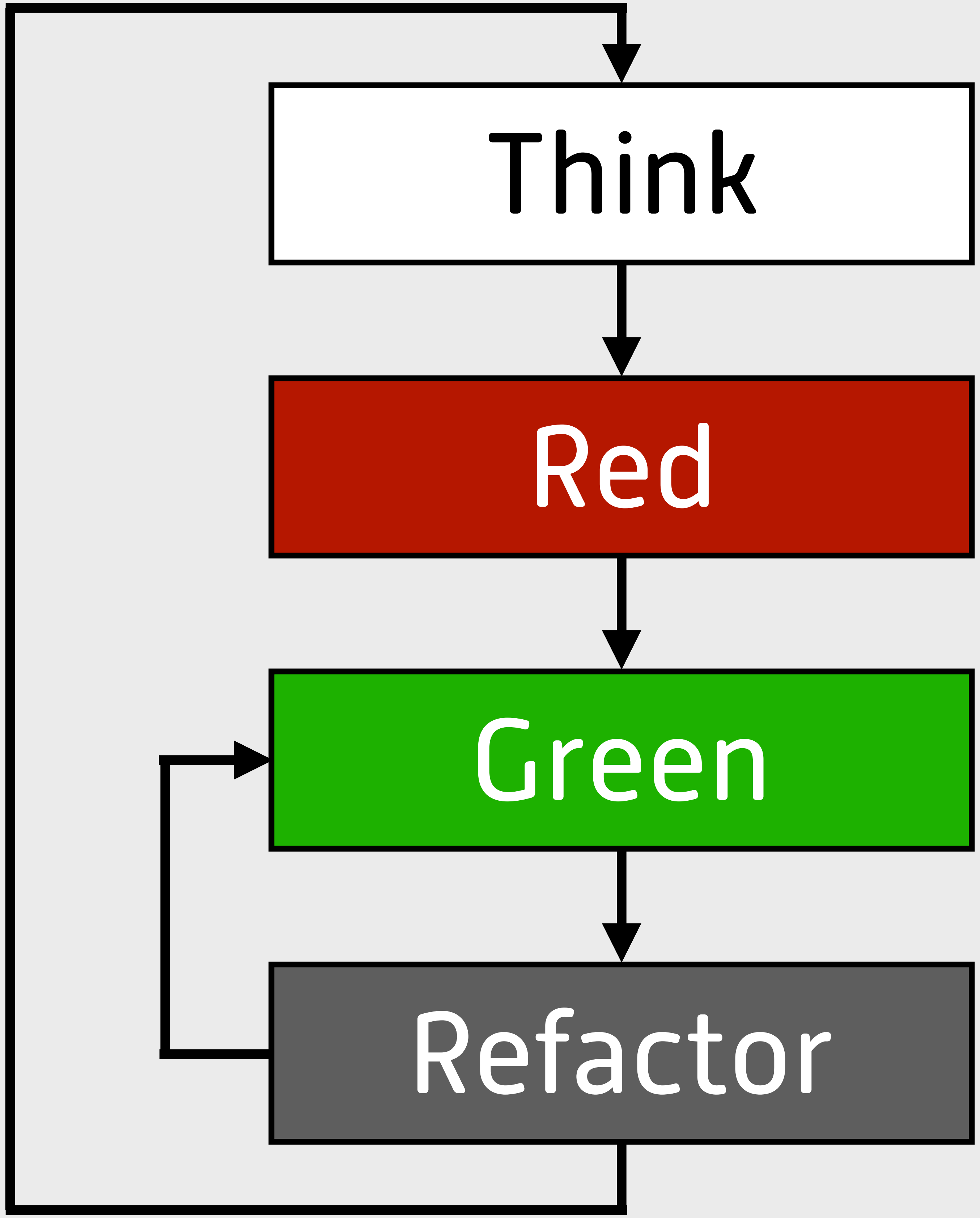


Finished after 0.014 seconds

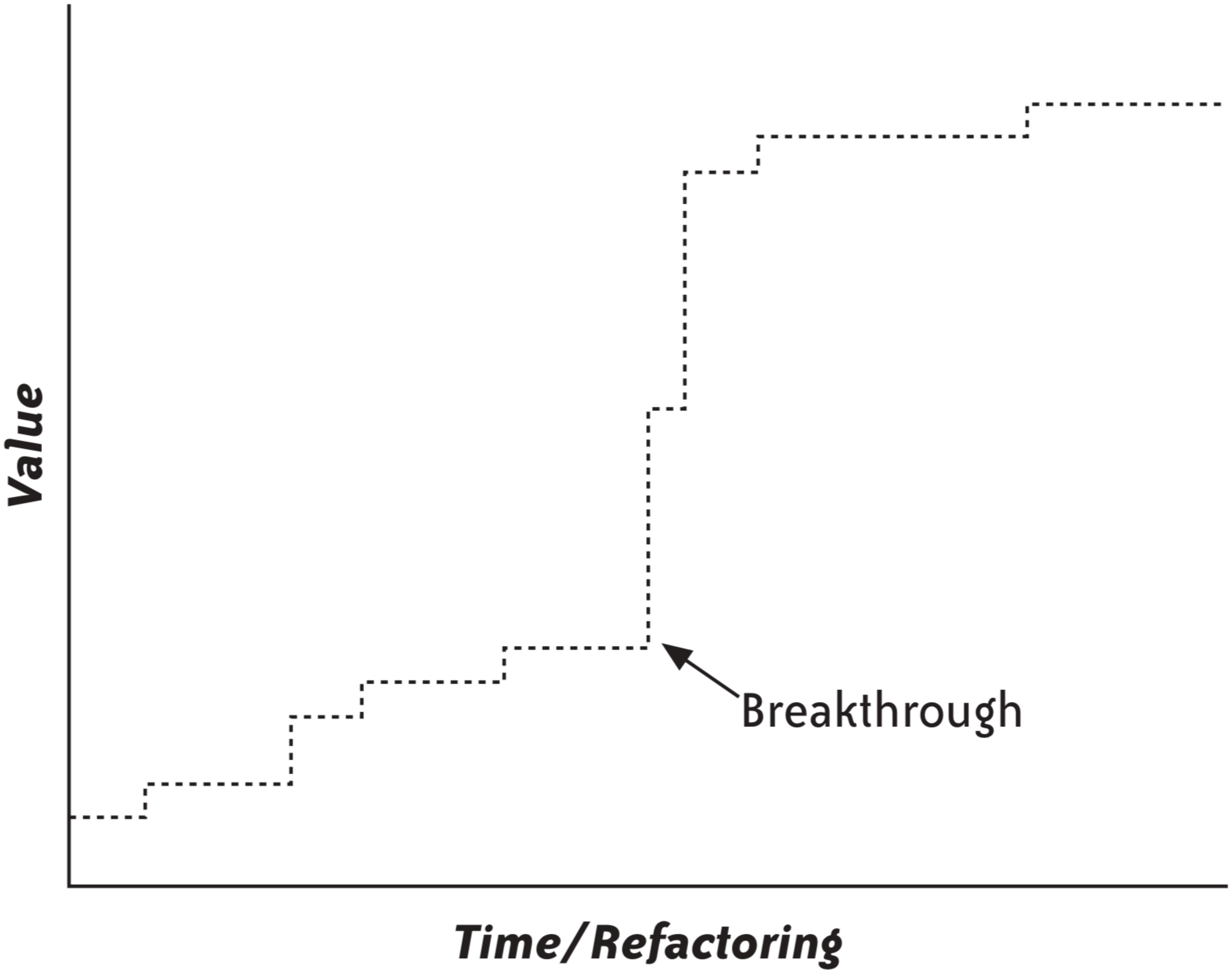
Runs: 4/4 Errors: 0 Failures: 0

```
        QueryString query = new QueryString("");
        assertEquals(0, query.count());
    }

    @Test
    public void testNull() {
        try {
            QueryString query = new QueryString(null);
            fail("should throw exception");
        }
        catch (NullPointerException e) {
            // expected
        }
    }

    @Test
    public void testOneNameValuePair() {
        QueryString query = new QueryString("name=value");
        assertEquals(1, query.count());
        assertEquals("value", query.valueFor("name"));
    }

    @Test
    public void testMultipleNameValuePairs() {
        QueryString query = new QueryString("name1=value1&n
        assertEquals(3, query.count());
        assertEquals("value1", query.valueFor("name1"));
        assertEquals("value2", query.valueFor("name2"));
        assertEquals("value3", query.valueFor("name3"));
```

```
package com.jamesshore.tdd_demo;

public class QueryString {

    private String _query;

    public QueryString(String queryString) {
        if (queryString == null) throw new NullPointerE

        _query = queryString;
    }

    public int count() {
        if ("".equals(_query)) return 0;

        String[] pairs = _query.split("&");
        return pairs.length;
    }

    public String valueFor(String name) {
        HashMap<String, String>

        String[] pairs = _query.split("&");
        for (String pair : pairs) {
            String[] nameAndValue = pair.split("=");
            if (nameAndValue[0].equals(name)) return na
        }
        throw new RuntimeException(name + " not found")
    }
```



Think
Red
Green
Refactor



Based on *Domain-Driven Design* by Eric Evans

SavingsAccount

Prevent Defect-Prone Designs

Merciless Refactoring

Evolutionary Design

Programmer Errors

Defect-Prone Designs

Requirements Misunderstandings

Systemic Blind Spots

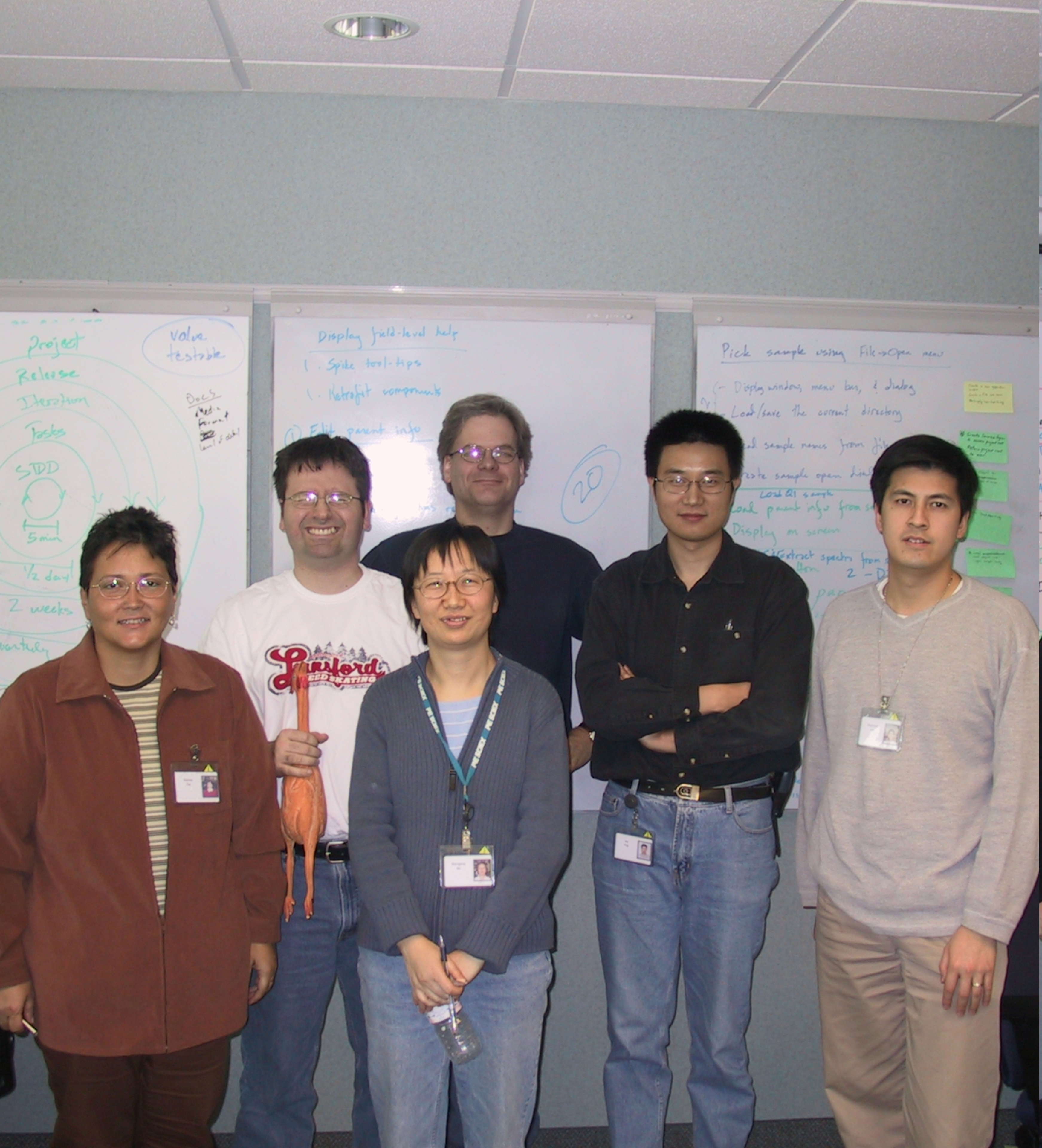
Requirements Misunderstandings

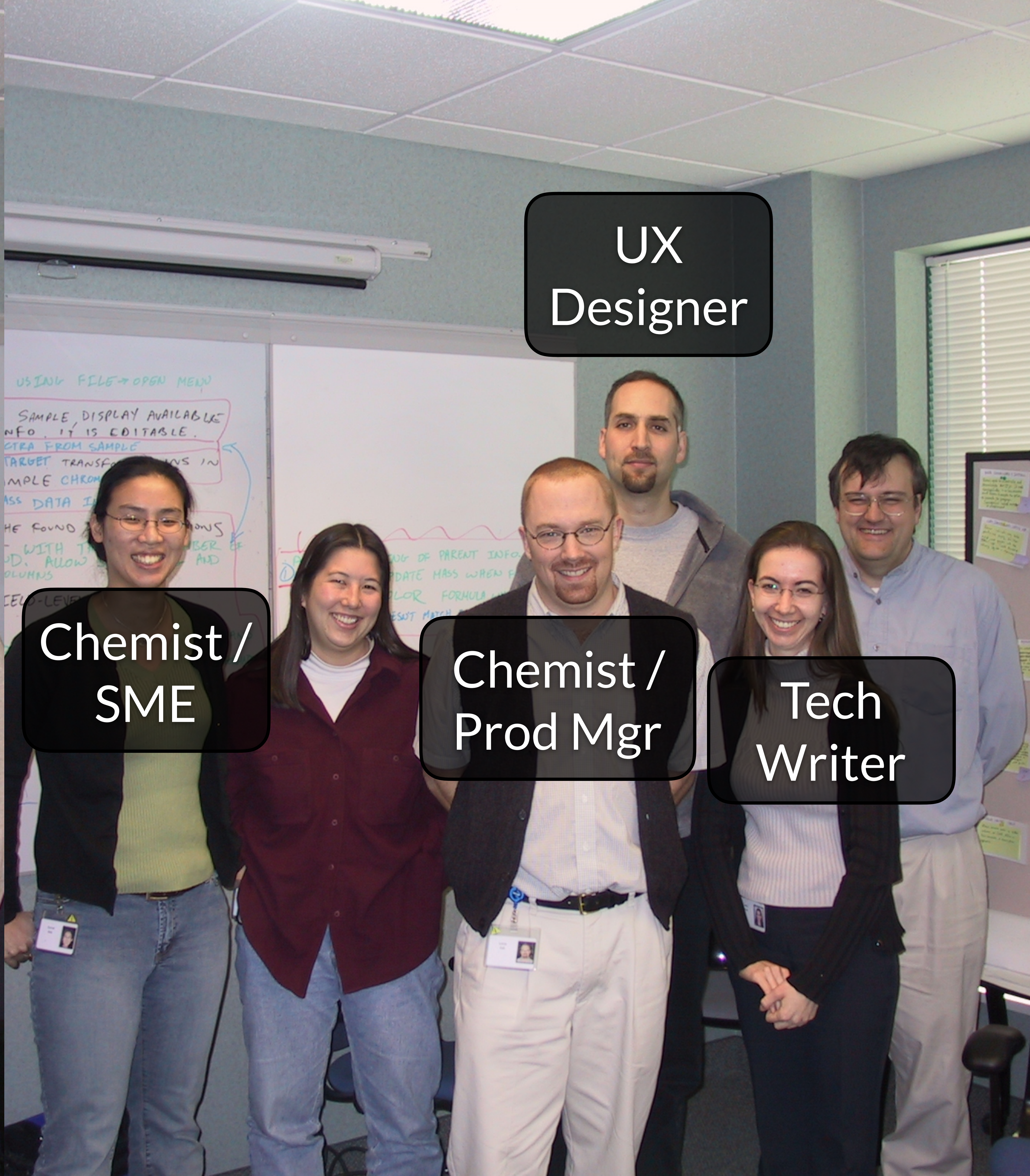
Requirements Game v1

1. On the bottom half of the handout, write instructions, **words only**, to reproduce the picture on your handout. If you have two pictures, choose the easiest one. (5 min)
2. Tear off instructions and exchange with someone in a **row above or below yours**. Keep your pictures hidden.
3. Use the other person's instructions to reproduce their picture. **Don't communicate** in any other way. (5 min)
4. Compare results.

Requirements Game v2

- Work with the same person as before. One of you will have another picture, the other will have a blank square. You'll have 5 minutes total.
- Person with picture: Tell the other person how to reproduce your picture. Keep it hidden. **Words only,** no gestures or props.
- Person with blank square: As the talker describes their picture, reproduce it in your blank square. You can **ask questions** and **show your progress**.





UX Designer

Chemist / SME

Chemist / Prod Mgr

Tech Writer

Prevent Misunderstandings

On-Site Customers

Customer Examples

Customer Review

Programmer Errors

Defect-Prone Designs

Requirements Misunderstandings

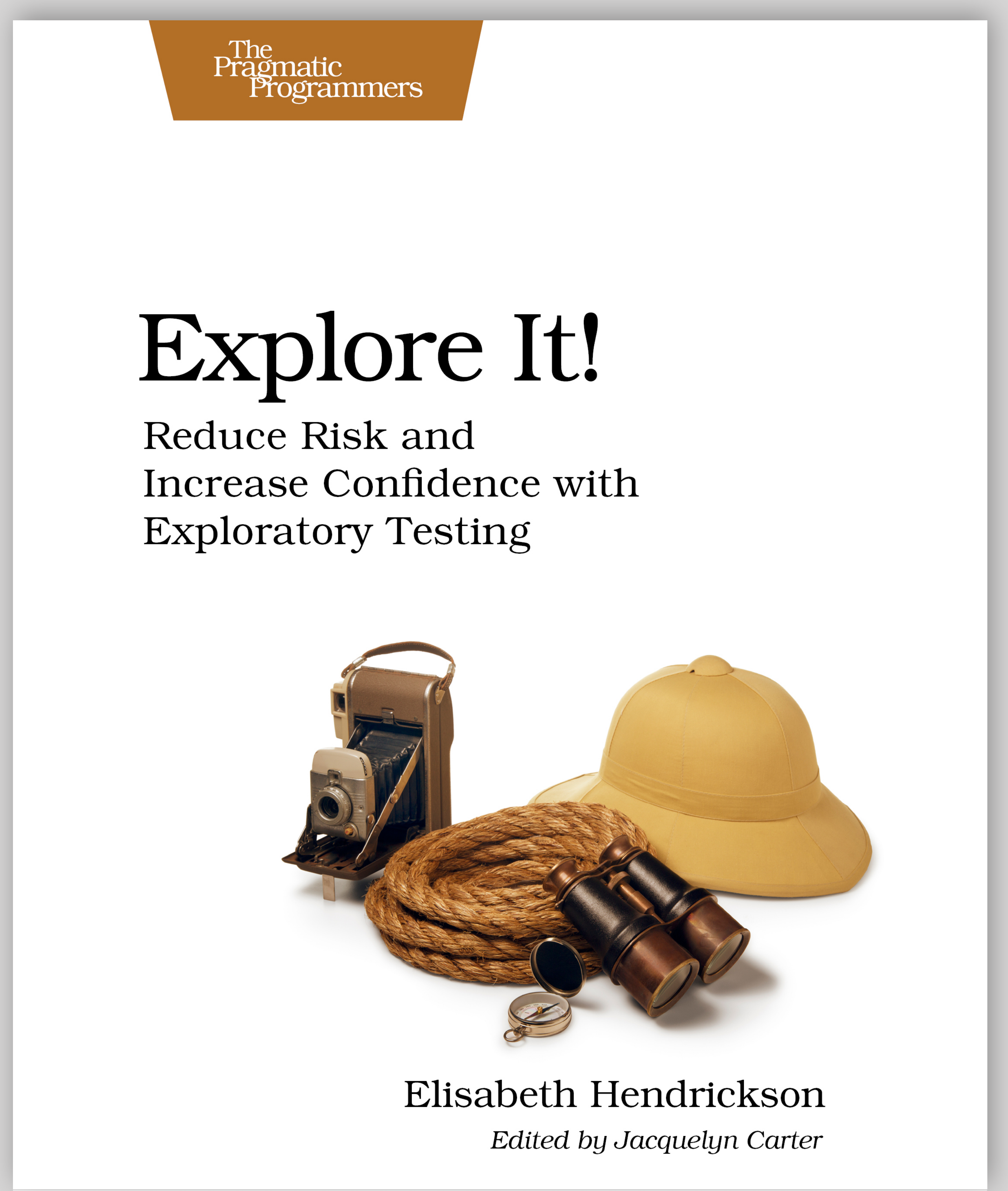
Systemic Blind Spots

Systemic Blind Spots

Excerpted from “I’ll Let Myself In: Tactics of Physical Pen Testers” by Deviant Ollam
Presented at Wild West Hackin’ Fest, 2017   youtube.com/watch?v=rnmcRTnTNC8

Dead Latches Rely on Proper Door Fitment

http://enterthecore.net



The Pragmatic Programmers

Explore It!

Reduce Risk and
Increase Confidence with
Exploratory Testing

Elisabeth Hendrickson

Edited by Jacquelyn Carter

How to Fix a Bug

- **Fix the bug.** Write a unit test, fix the code.
- **Fix the design.** What about the software design allowed this bug to hide from view? Refactor to make this category of bugs impossible or obvious.
- **Fix the process.** What enabled this type of bug to exist in the first place? Look at systems, processes, and habits, not people. Can they be improved?
- **Explore further.** Based on what we've learned, what similar bugs are likely to exist? Find and fix them, too.

BUGS ARE FOR OTHER PEOPLE

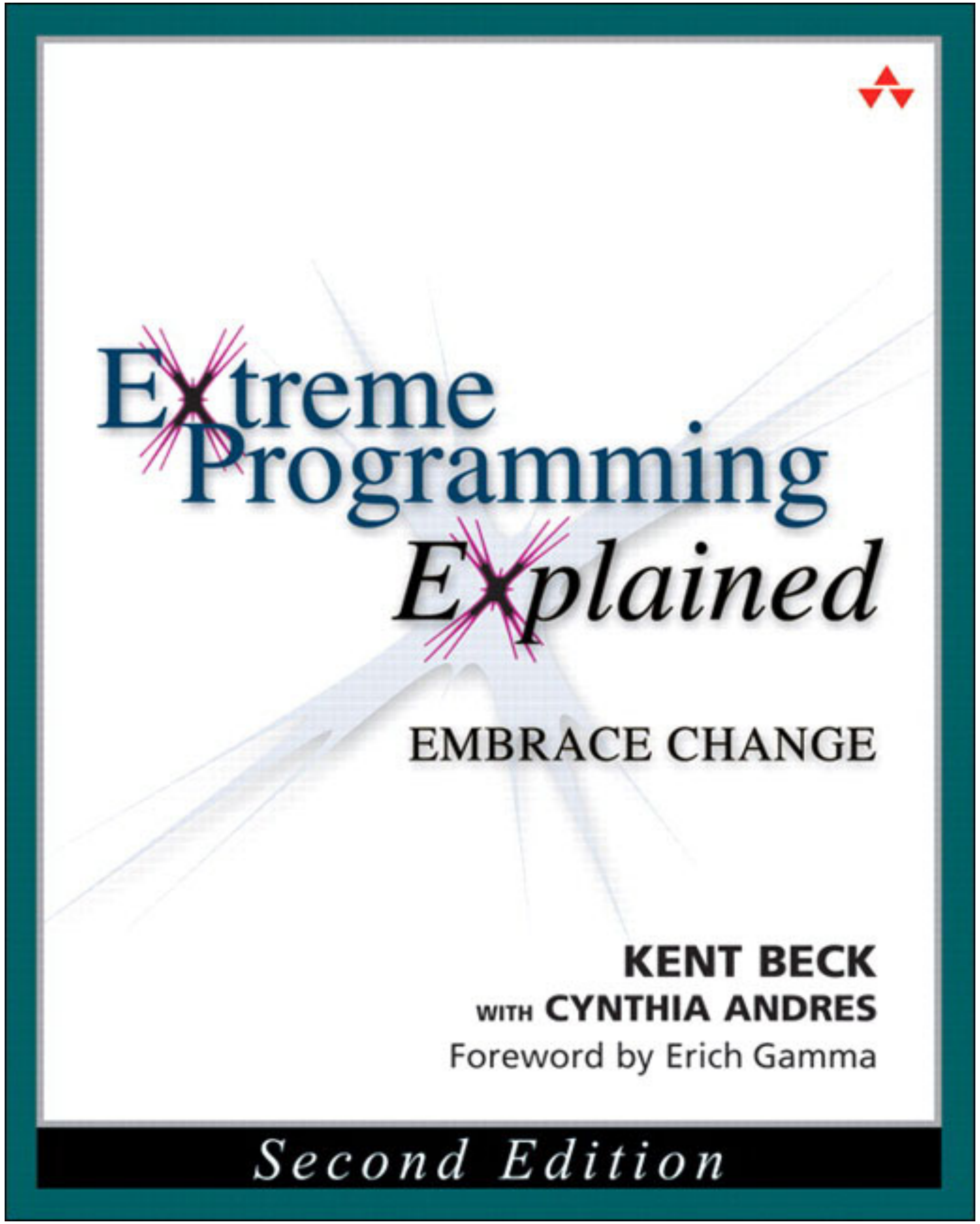
DIVINA
ANTIPASTI
BETTER THAN BOUILLON
KOO-KOO-R
GOOD FOR
LIFETIME
SNICKERS
CLAW MEAT
BYRD
BETTER THAN BOUILLON

Prevent Systemic Blind Spots

Exploratory Testing

Root-Cause Analysis

No Bug Database (‘Tude)



Extreme Programming Explained

EMBRACE CHANGE

KENT BECK

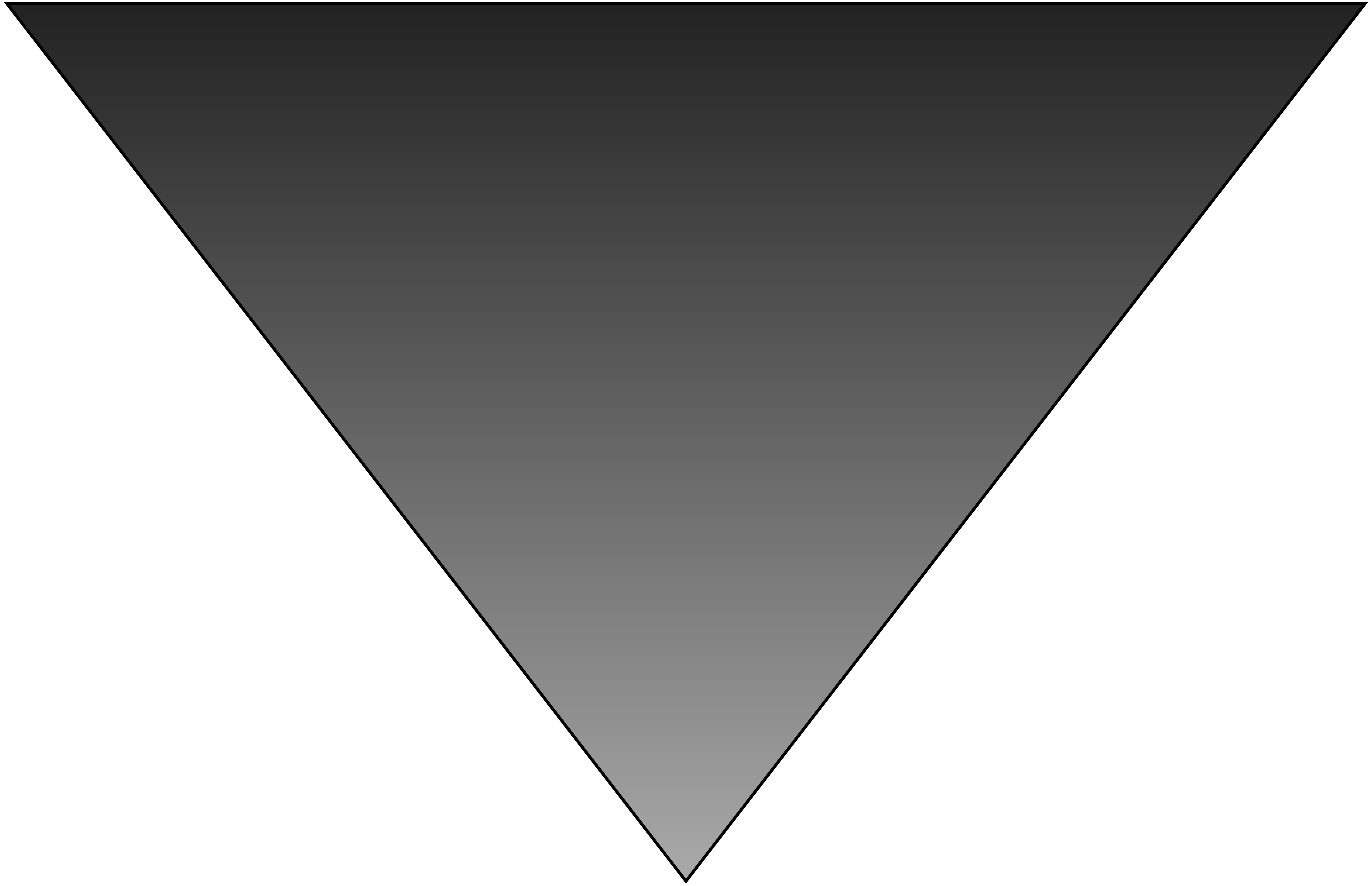
WITH CYNTHIA ANDRES

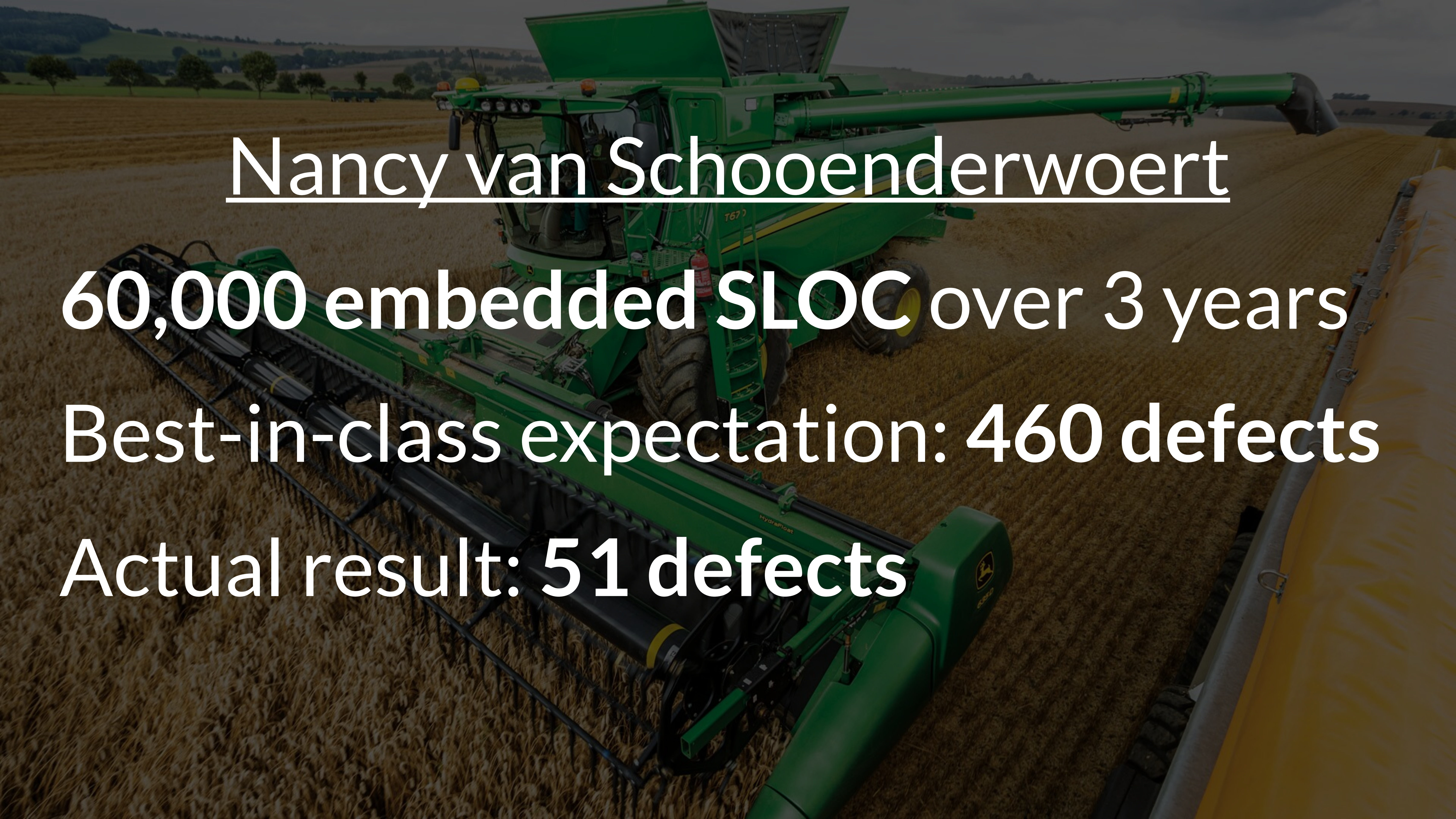
Foreword by Erich Gamma

Second Edition

Explained

EMBRACE CHANGE





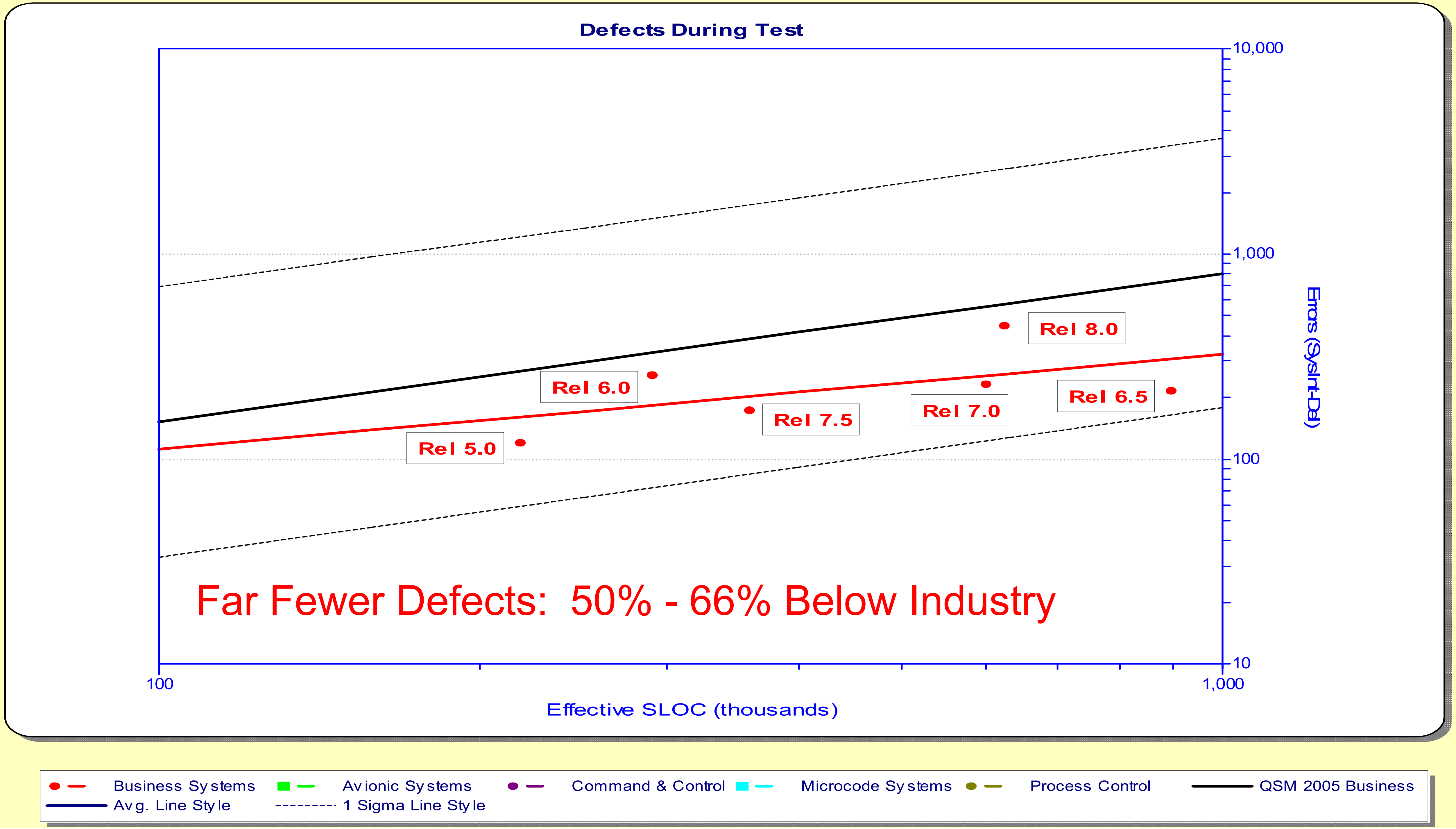
Nancy van Schooenderwoert

60,000 embedded SLOC over 3 years

Best-in-class expectation: **460 defects**

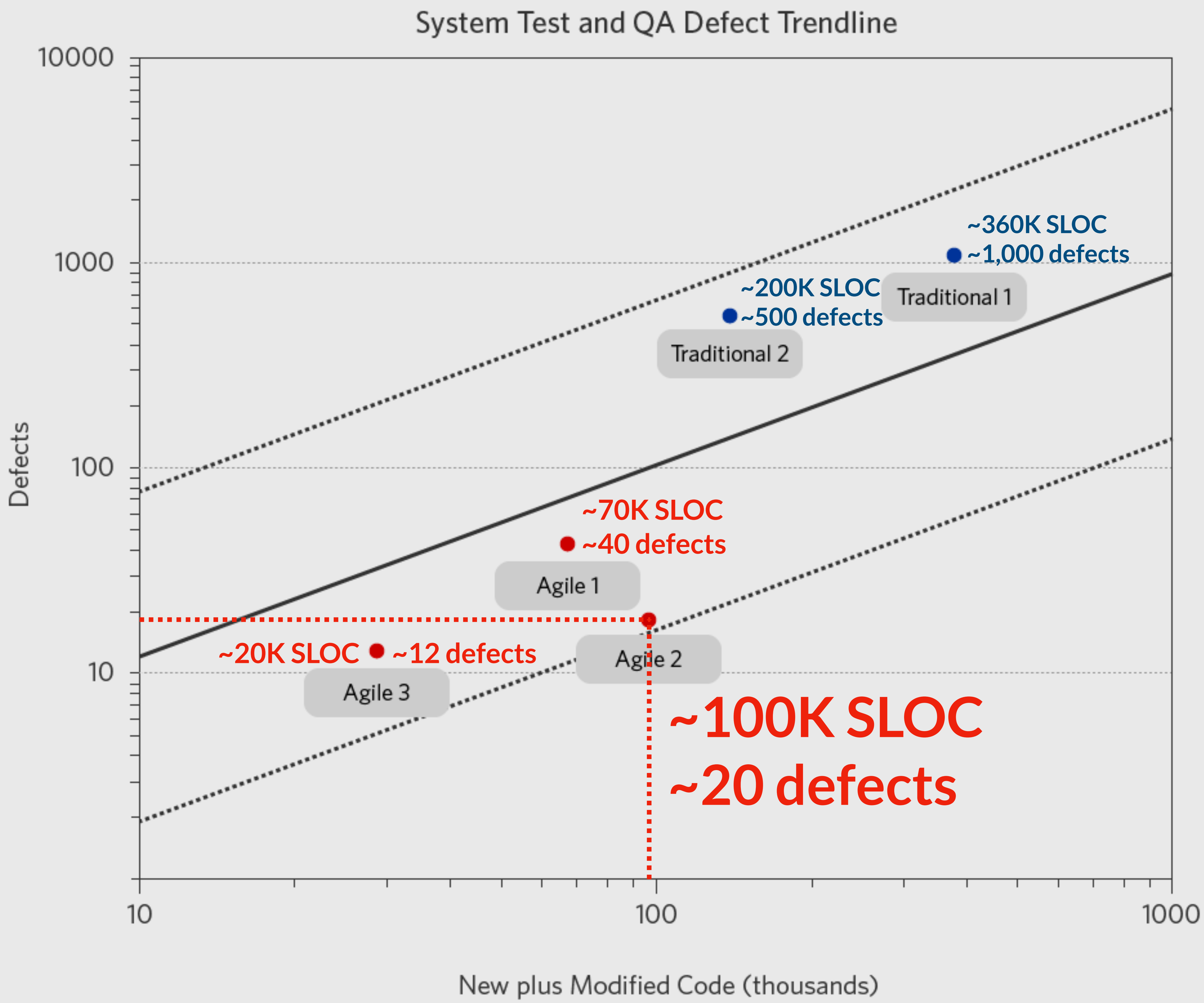
Actual result: **51 defects**

Trendline Assessment – Defects/Quality



Far Fewer Defects: 50% - 66% Below Industry

Excerpted from Michael Mah PNSQC 2010 presentation:
"The Good, the Bad, and the Puzzling: The Agile Experience at 5 Companies"



https://www.industriallogic.com/an-agile-xp-transition-continues-to-thrive/

Prevent Programmer Errors

Test-Driven Development
Pairing or Mobbing
Energized Work

Prevent Defect-Prone Designs

Merciless Refactoring
Evolutionary Design

Prevent Misunderstandings

On-Site Customers
Customer Examples
Customer Review

Prevent Blind Spots

Exploratory Testing
Root-Cause Analysis
No Bug Database (‘Tude)

Agile Without Dedicated QA

James Shore

TWITTER: @jamesshore
EMAIL: jshore@jamesshore.com
WEB: jamesshore.com
GITHUB: github.com/jamesshore

Pacific Northwest Software Quality Conference
October 15, 2019