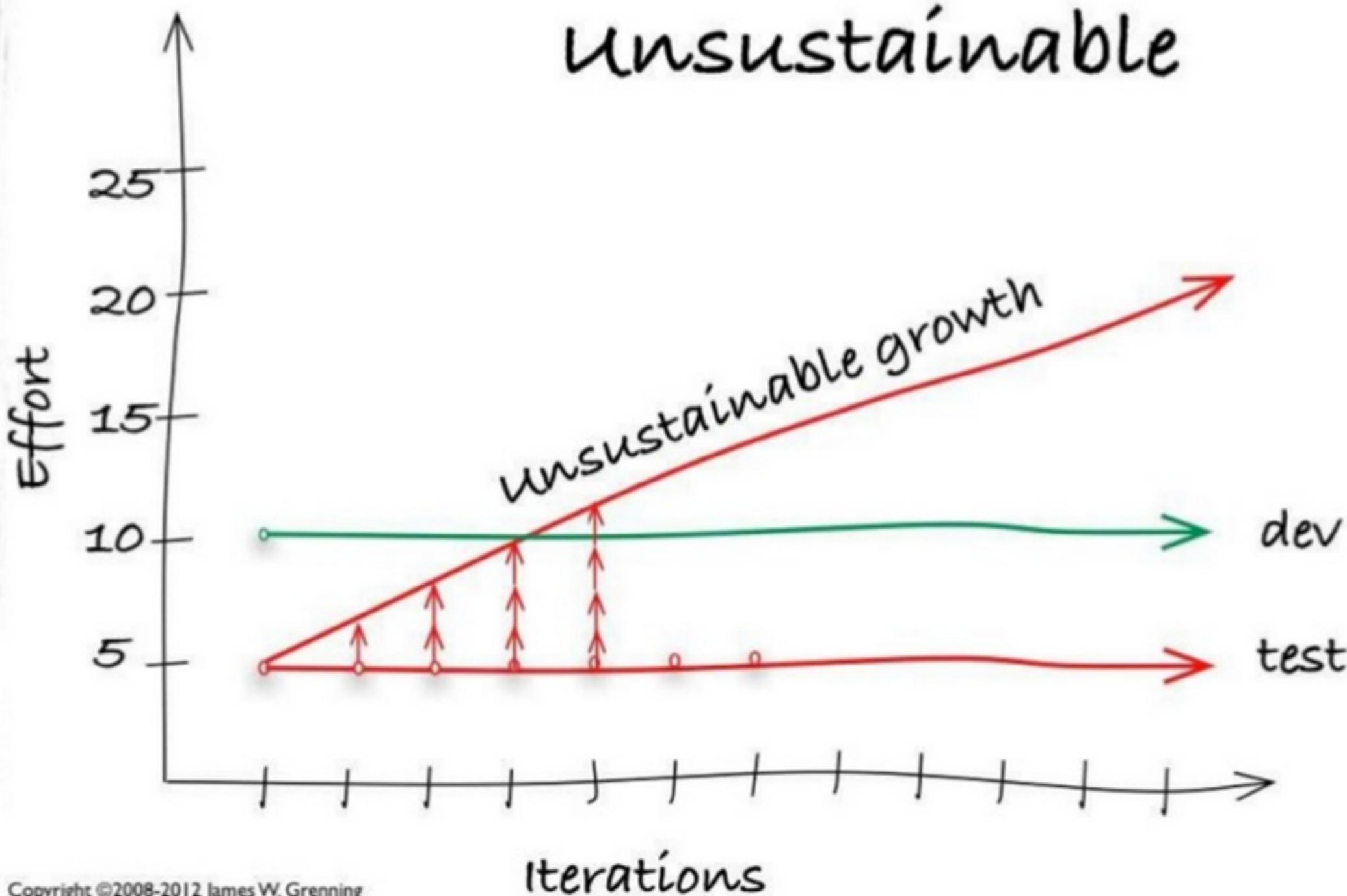


Developer with testing

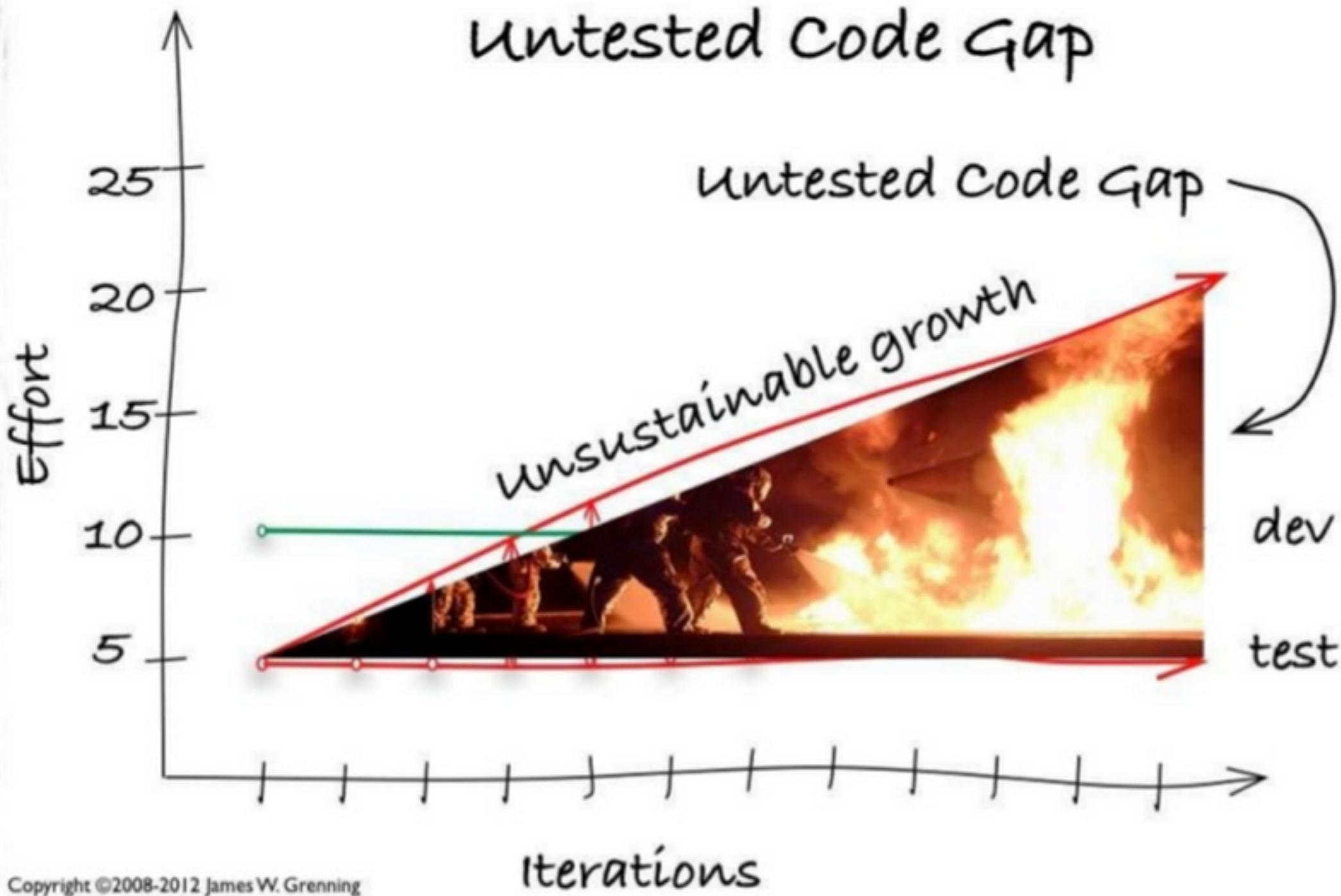


Manual Test is Unsustainable



Copyright ©2008-2012 James W. Grenning
All Rights Reserved.

Risk Accumulates in the Untested Code Gap



Copyright ©2008-2012 James W. Grenning
All Rights Reserved.

Iterative and Incremental Development: A Brief History



Although many view iterative and incremental development as a modern practice, its application dates as far back as the mid-1950s. Prominent software-engineering thought leaders from each succeeding decade supported IID practices, and many large projects used them successfully.

Craig Larman
Valtech

Victor R. Basili
University of Maryland

As agile methods become more popular, some view iterative, evolutionary, and incremental software development—a cornerstone of these methods—as the “modern” replacement of the waterfall model, but its practiced and published roots go back decades. Of course, many software-engineering students are aware of this, yet surprisingly, some commercial and government organizations still are not.

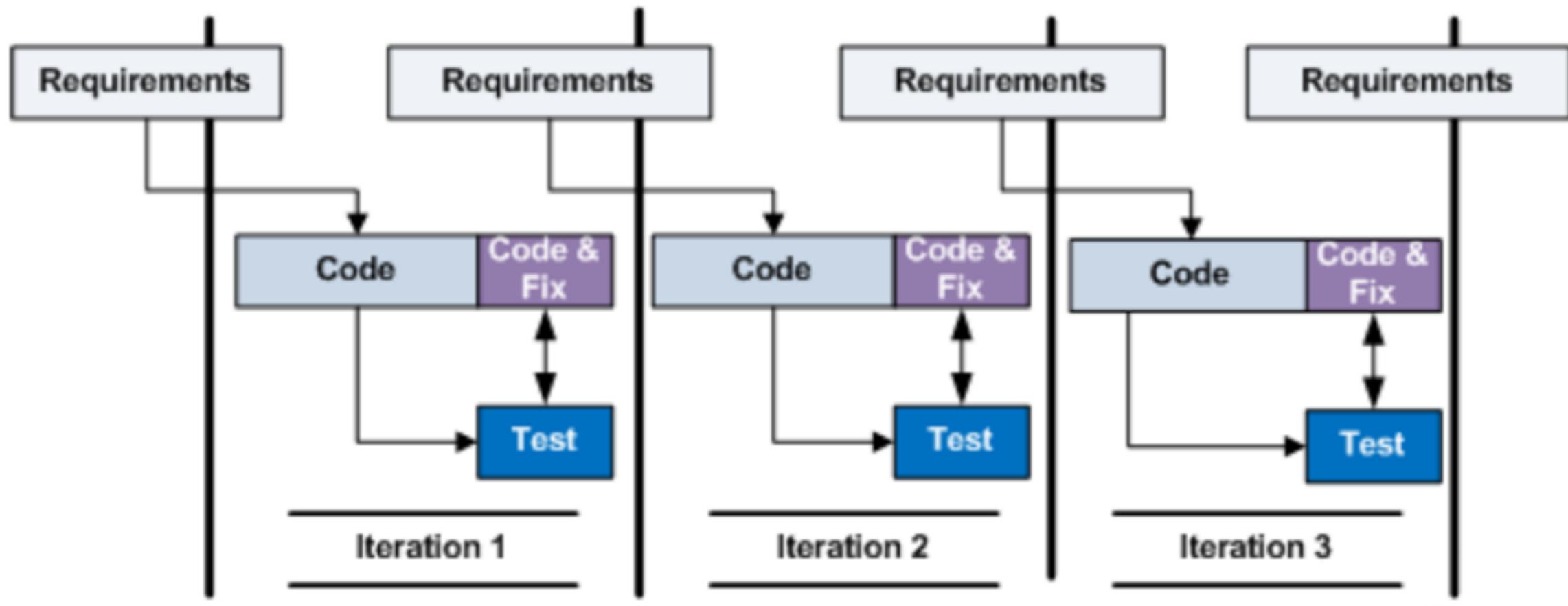
This description of projects and individual contributions provides compelling evidence of iterative and incremental development (IID) practices in the 1950s and 1960s.

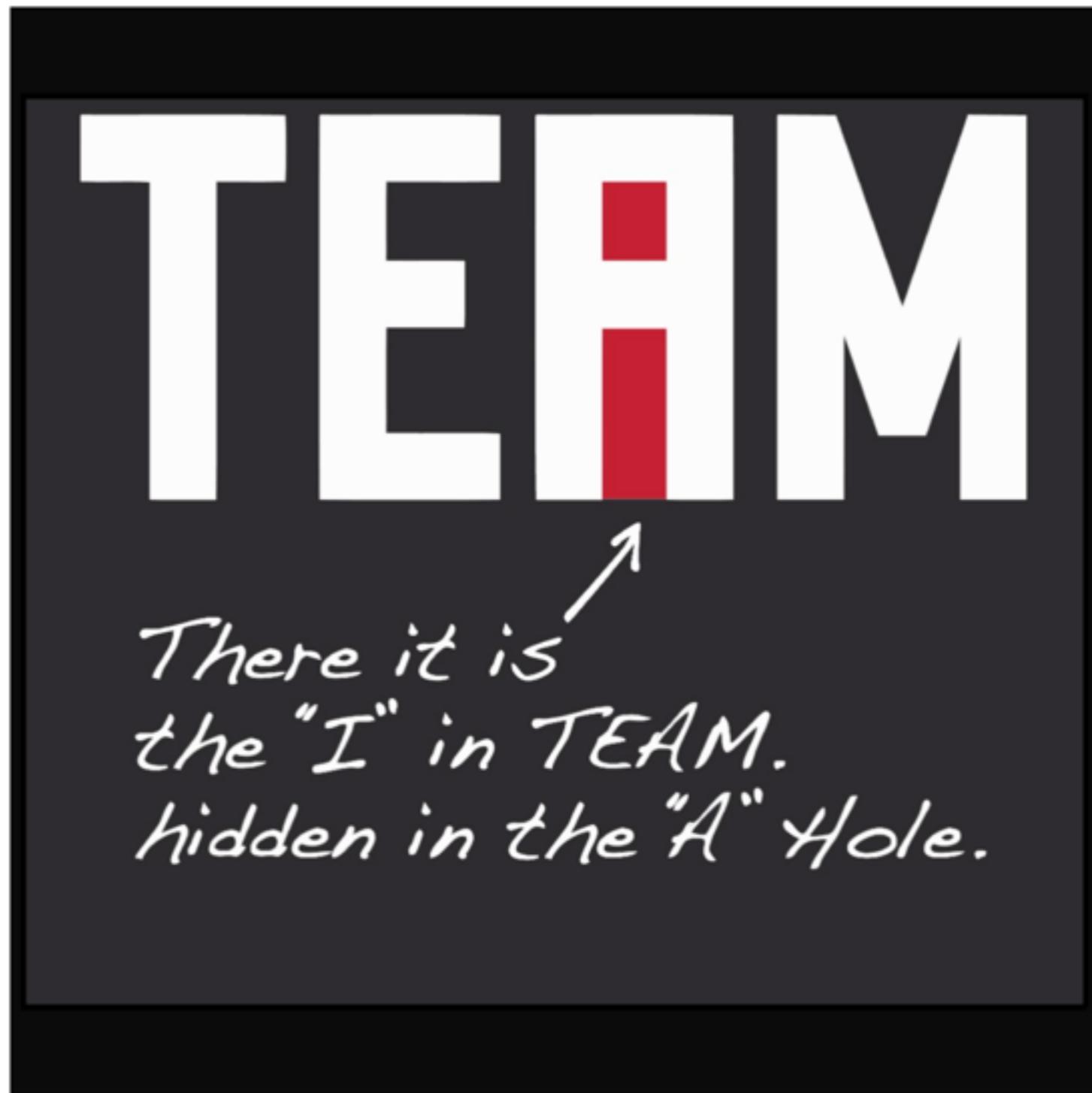
opment” merely for rework, in modern agile methods the term implies not just revisiting work, but also evolutionary advancement—a usage that dates from at least 1968.

PRE-1970

IID grew from the 1930s work of Walter Shewhart,¹ a quality expert at Bell Labs who proposed a series of short “plan-do-study-act” (PDSA) cycles for quality improvement. Starting in the 1940s, quality guru W. Edwards Deming began

Real problem ?



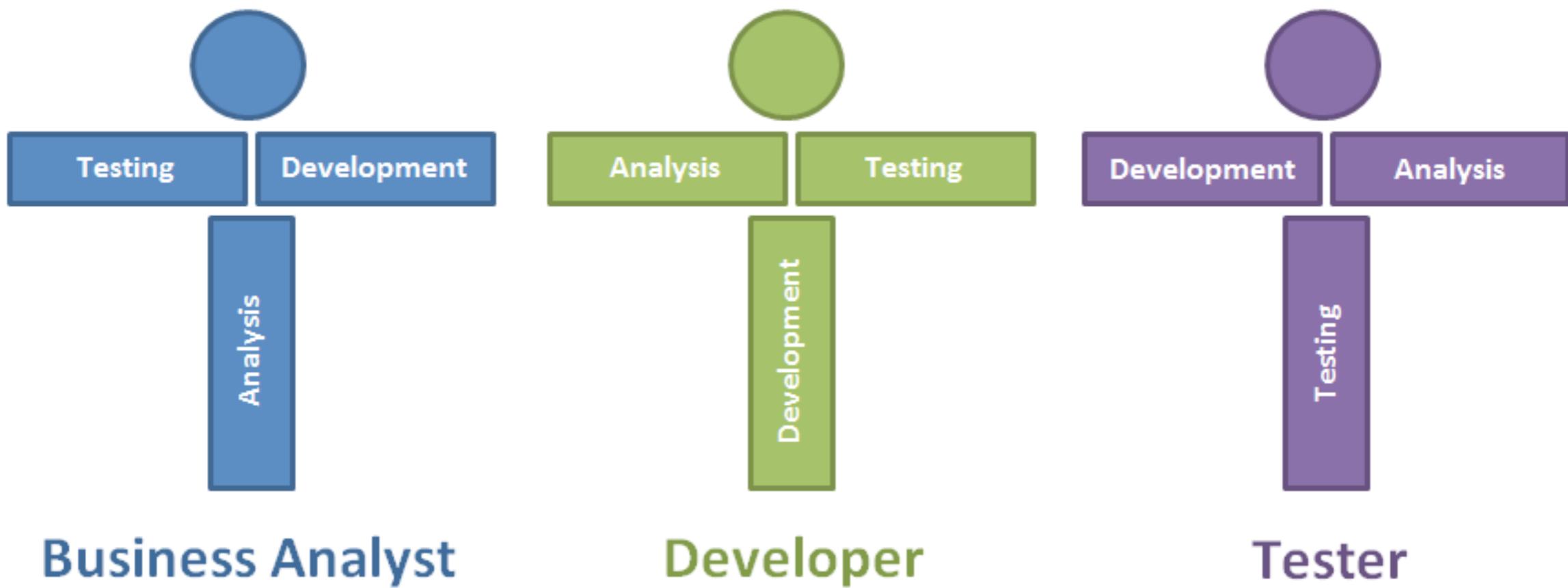


I in Team - There it is hidden in the... by ginjavv

Zazzle

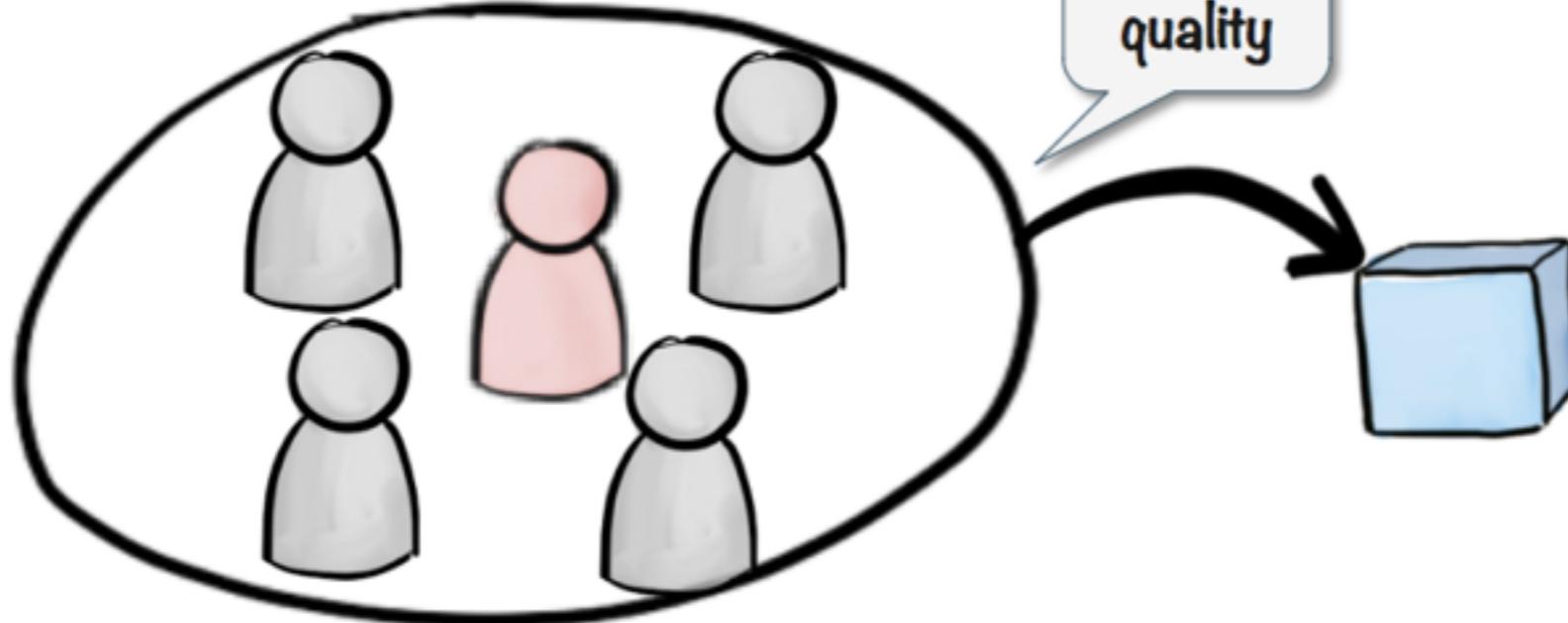
T-Shaped Agile Team

@katrina_tester



<http://blog.scottlogic.com/2014/07/07/School-of-Testing.html>

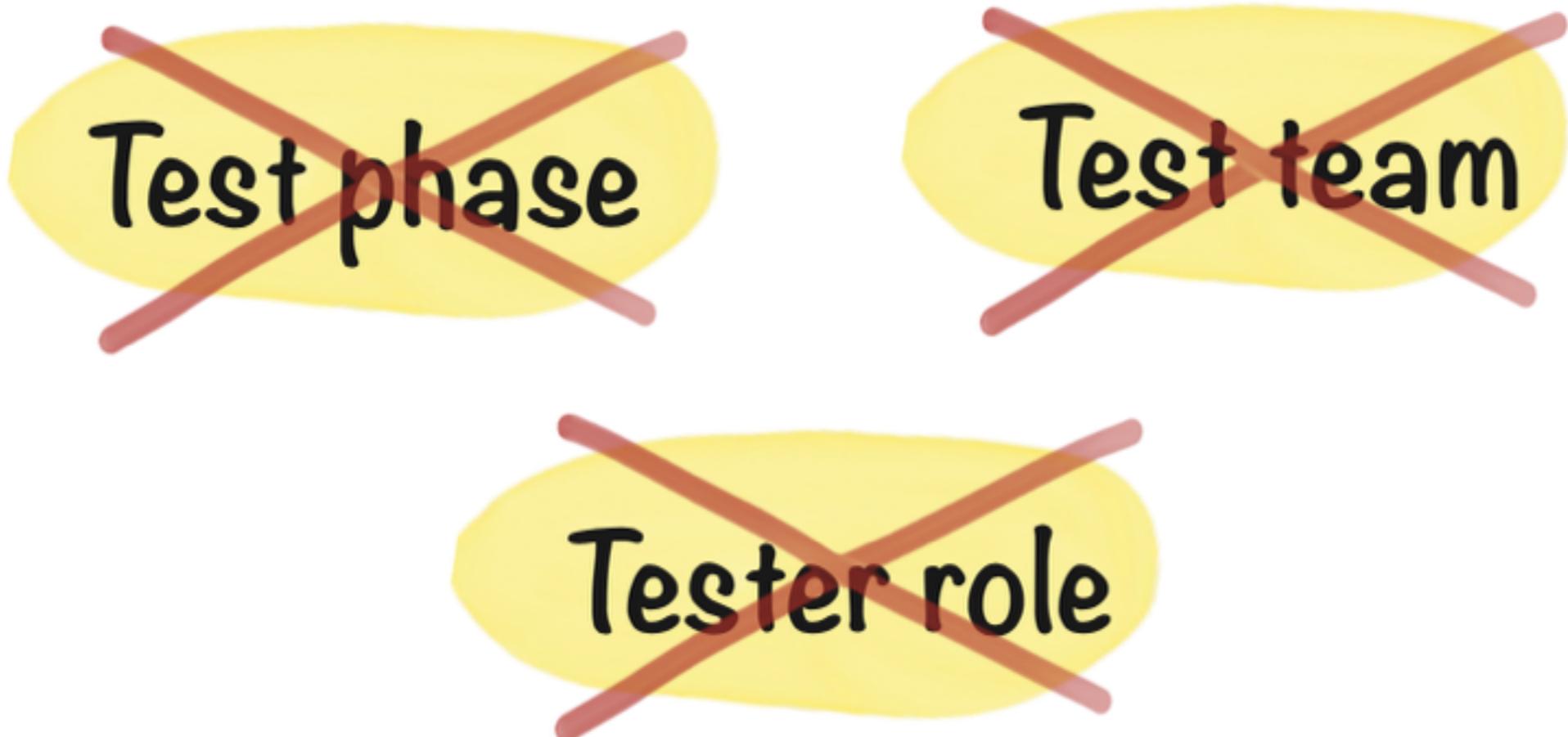
Role → **Competency**



Cross-functional development team

~~QA = Quality Assurance~~
= Quality Assistance

<http://blog.crisp.se/wp-content/uploads/2013/08/20130820-What-is-Agile.pdf>

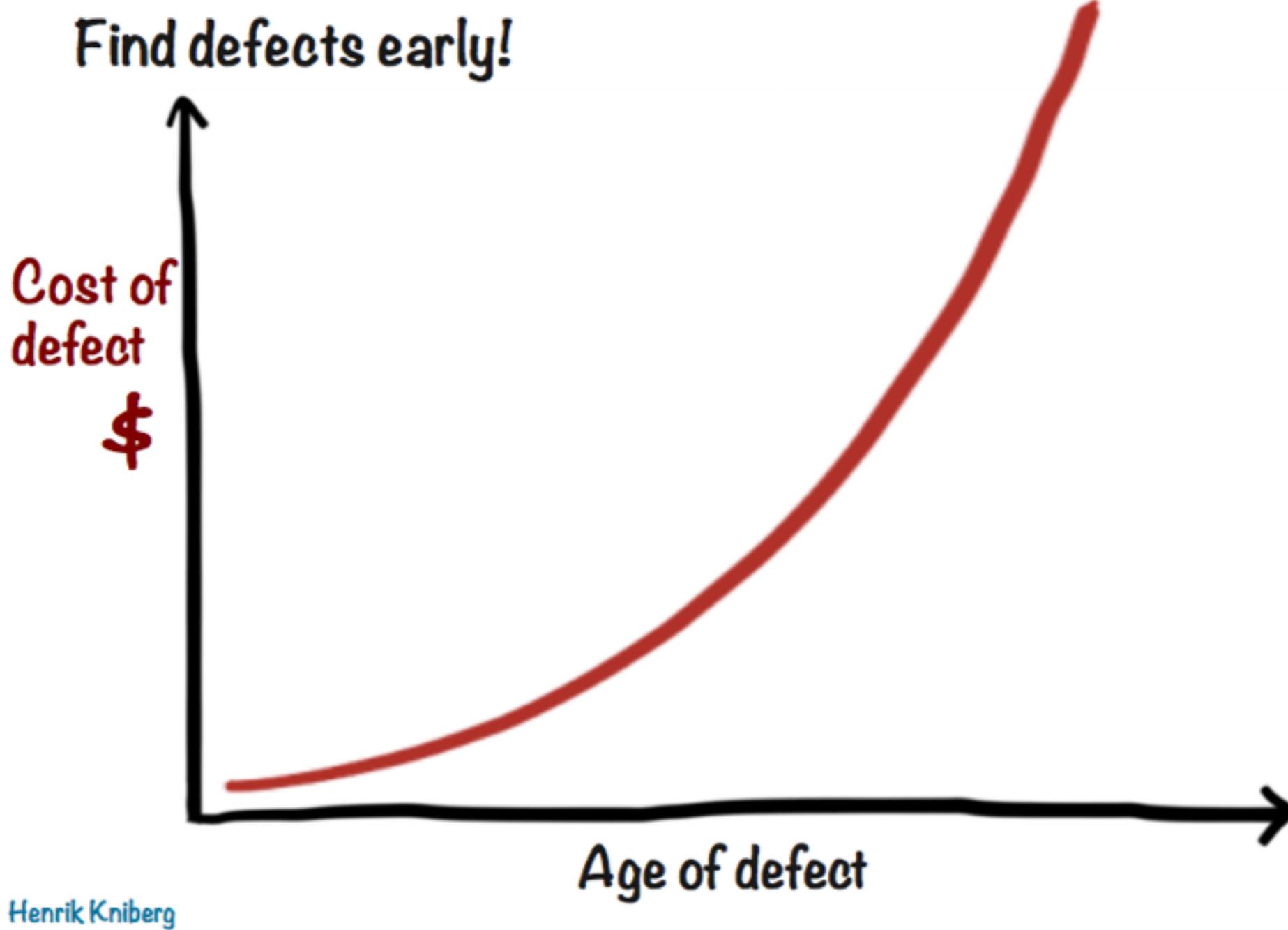


<http://blog.crisp.se/wp-content/uploads/2013/08/20130820-What-is-Agile.pdf>

Testing is Activity ?

Make sure all stories are
Testable & Valuable

Show your testing activity



<http://blog.crisp.se/wp-content/uploads/2013/08/20130820-What-is-Agile.pdf>

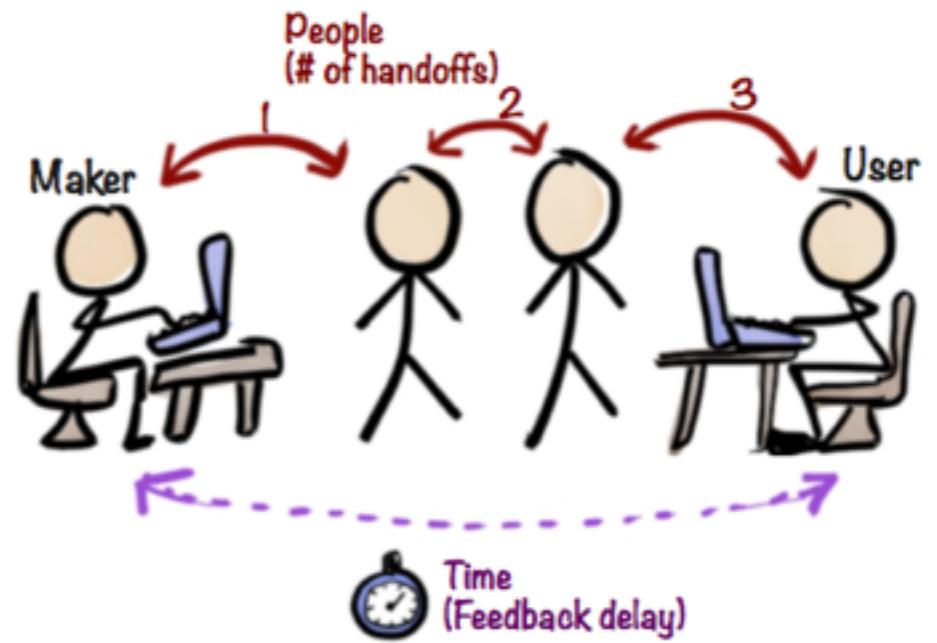
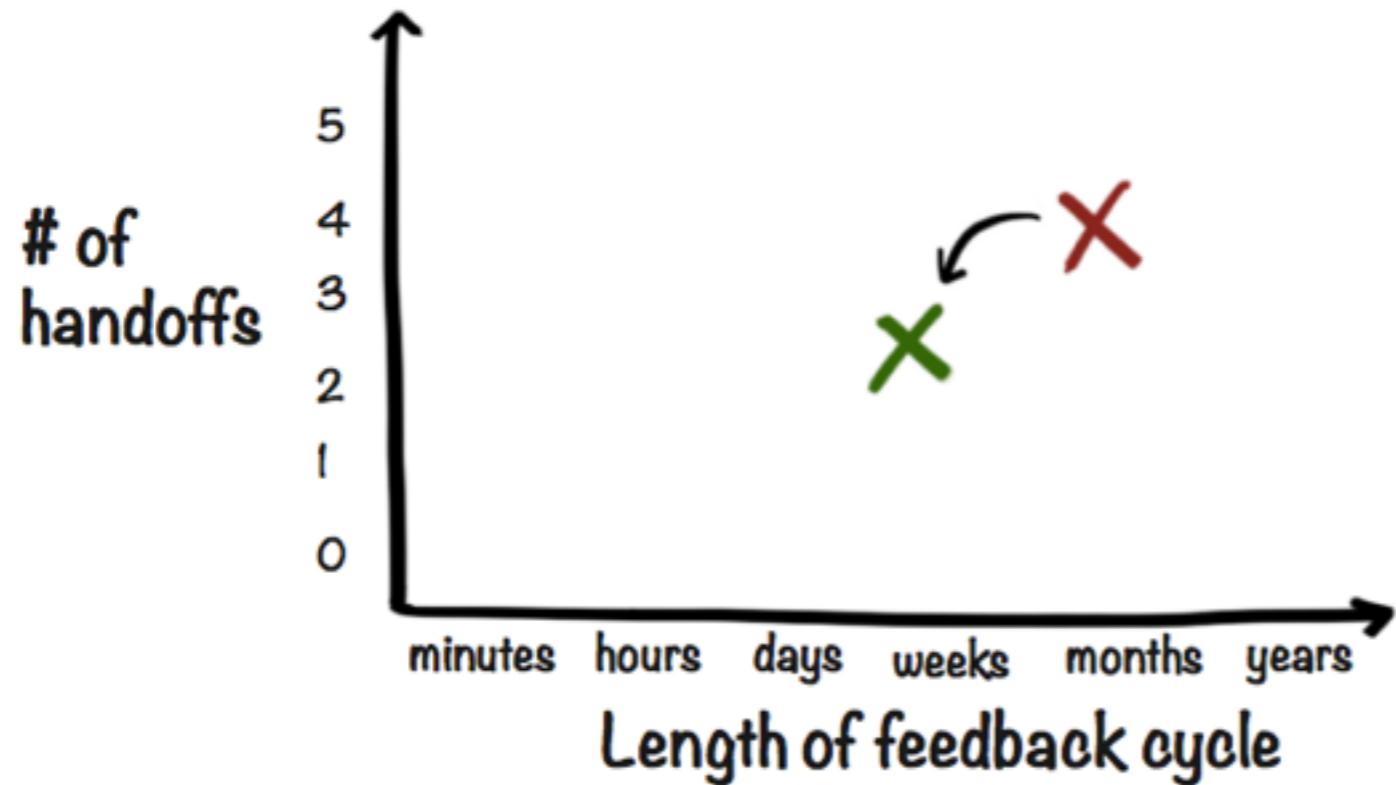
Testing early & often

Sit with developers



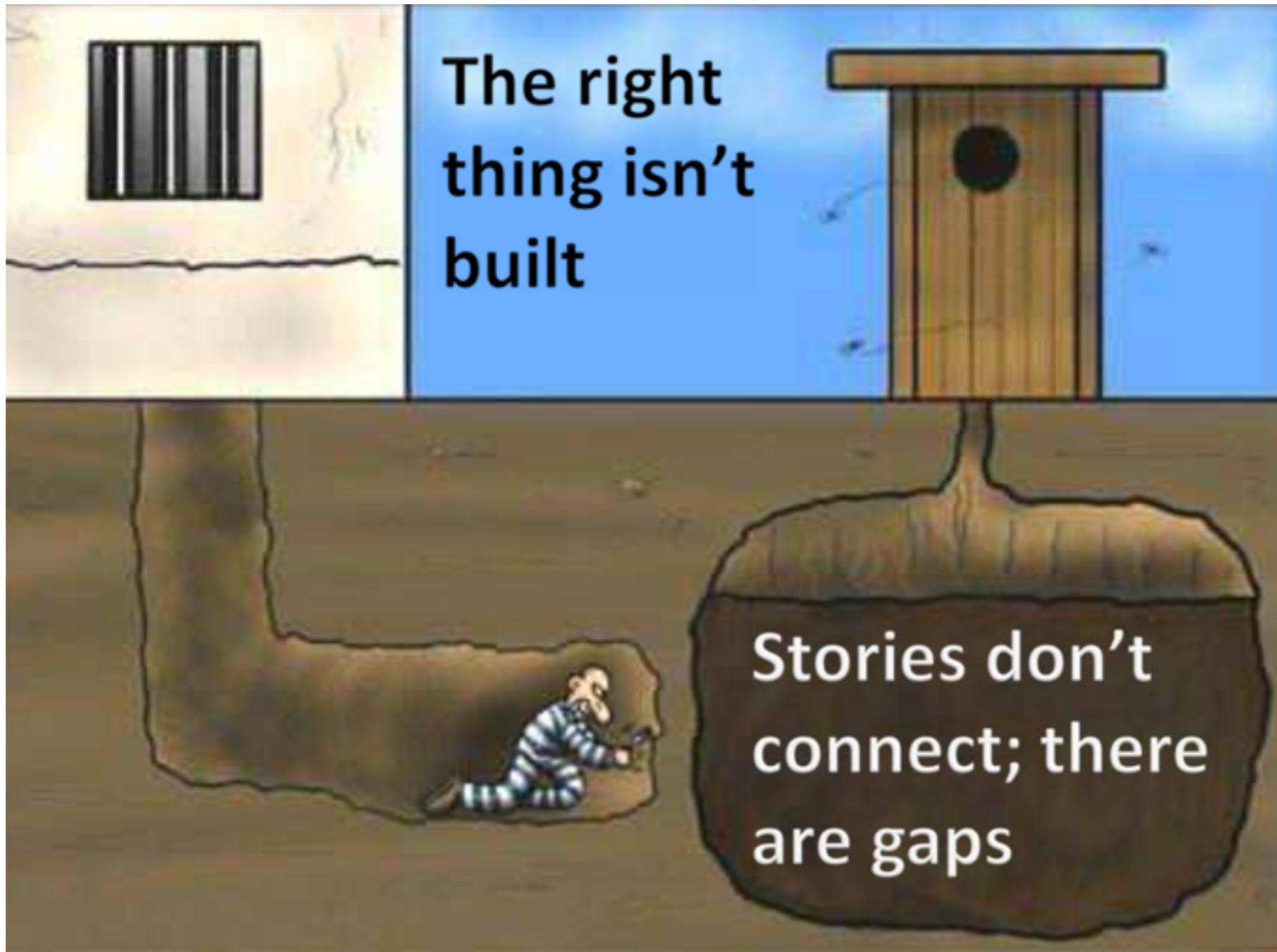
All the defects are Fixed and Retested by Tom and Jerry

Shorten the feedback loop

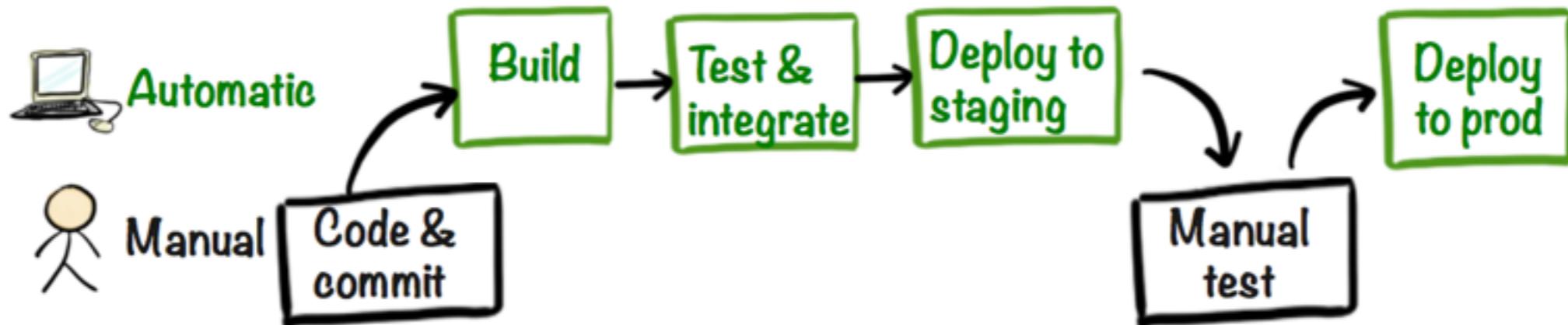


<http://blog.crisp.se/wp-content/uploads/2013/08/20130820-What-is-Agile.pdf>

Understand big picture



Push for Continuous Delivery



<http://blog.crisp.se/wp-content/uploads/2013/08/20130820-What-is-Agile.pdf>



**It worked and tested in
dev**

**Now
it's devops problem**

quickmeme.com

Create a shared vocabulary

Unit
Test?

Integration
test?

Quality?

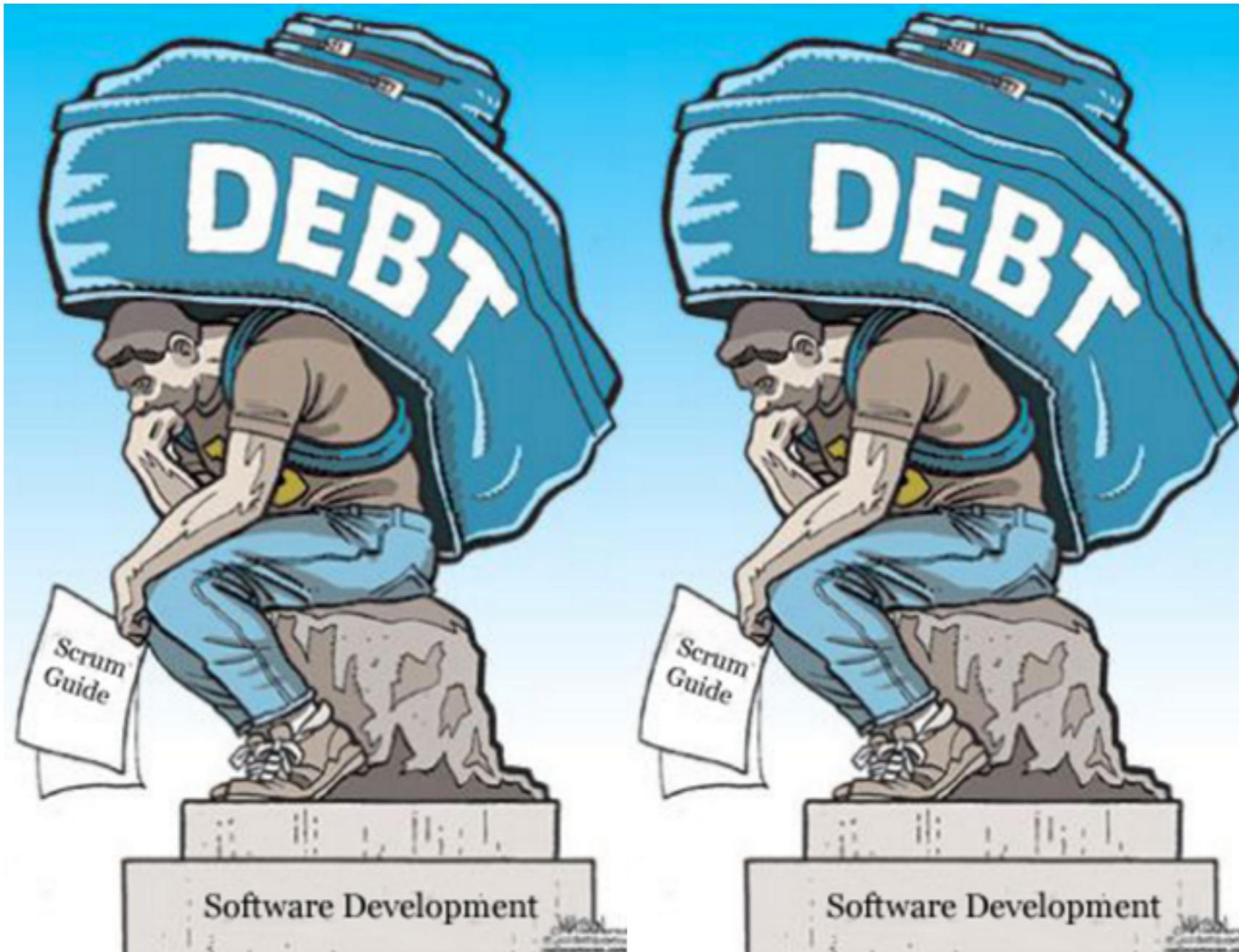
Acceptance
test?

Technical
debt?



<http://blog.crisp.se/wp-content/uploads/2013/08/20130820-What-is-Agile.pdf>

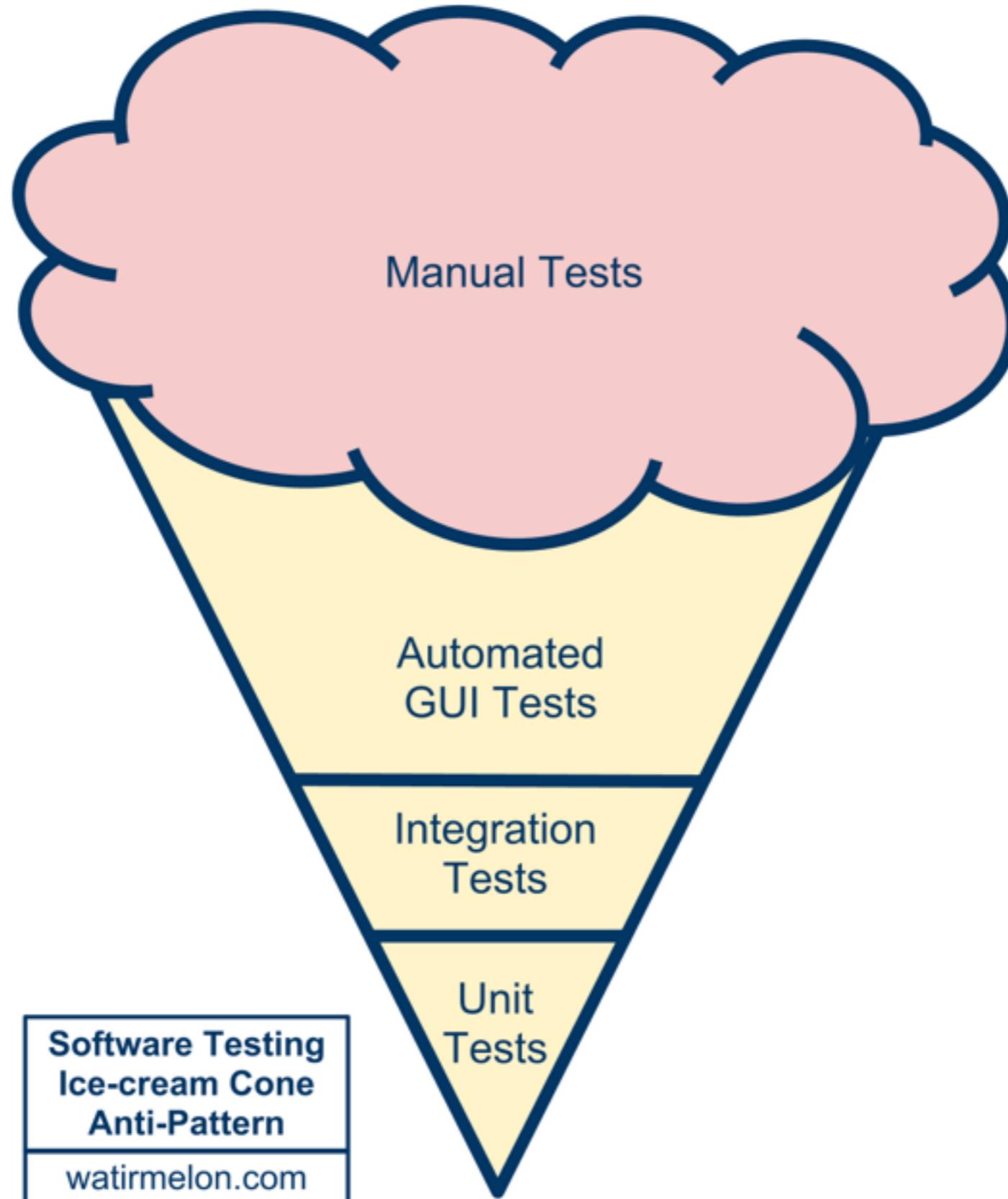
Don't add technical debt

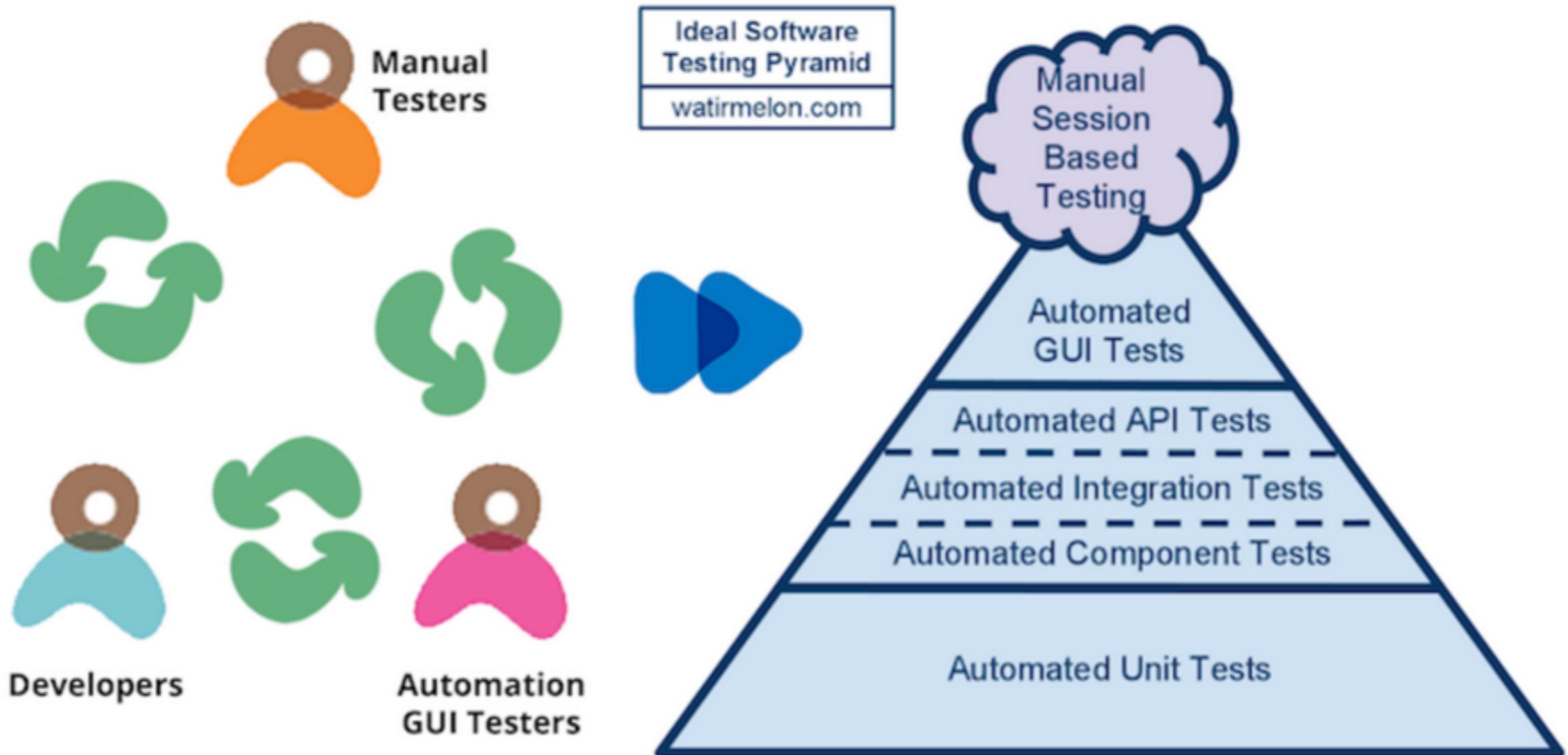


Quality Build-in

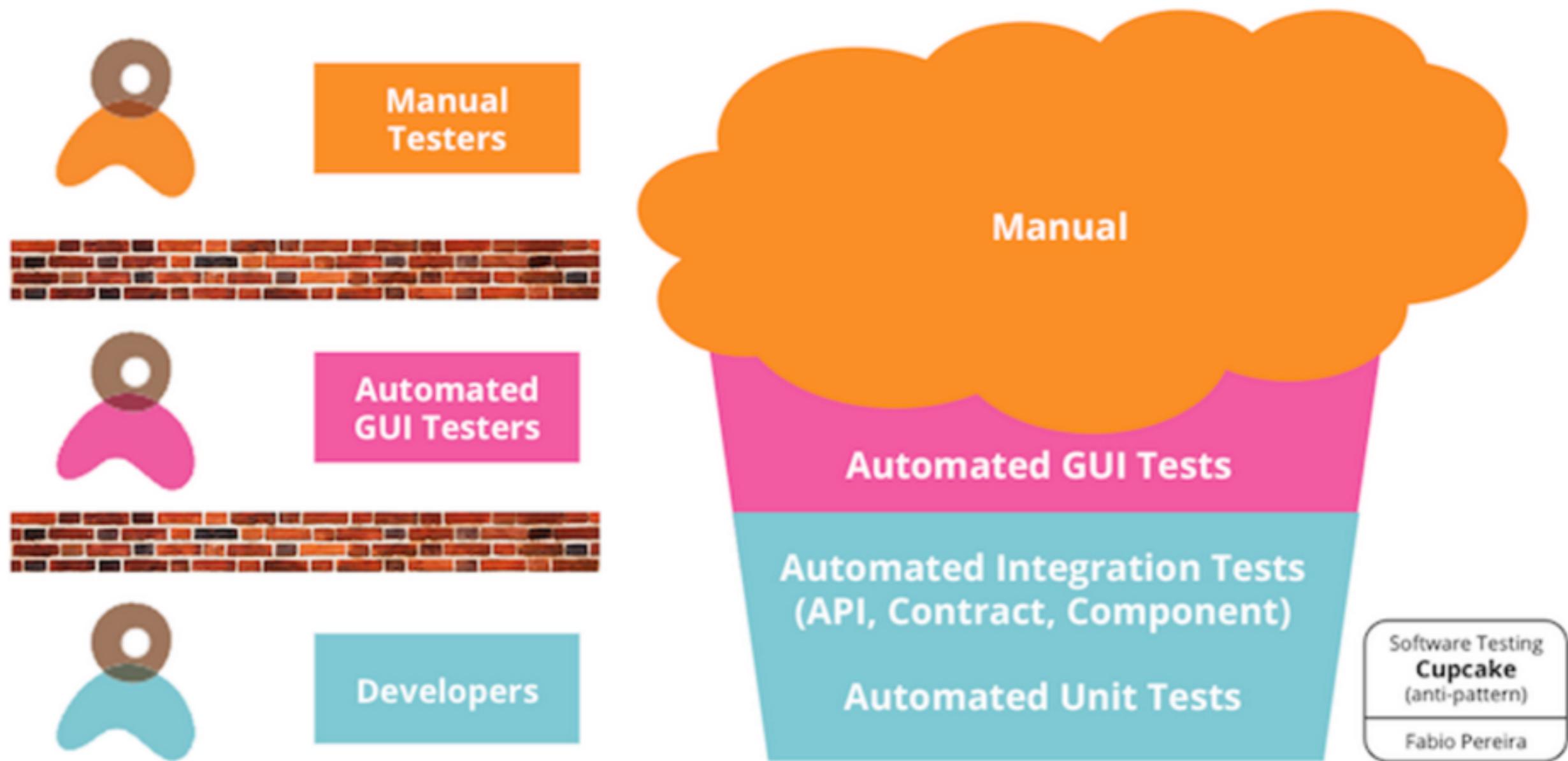
**Quality is
everyone's responsibility**

Software Testing pattern

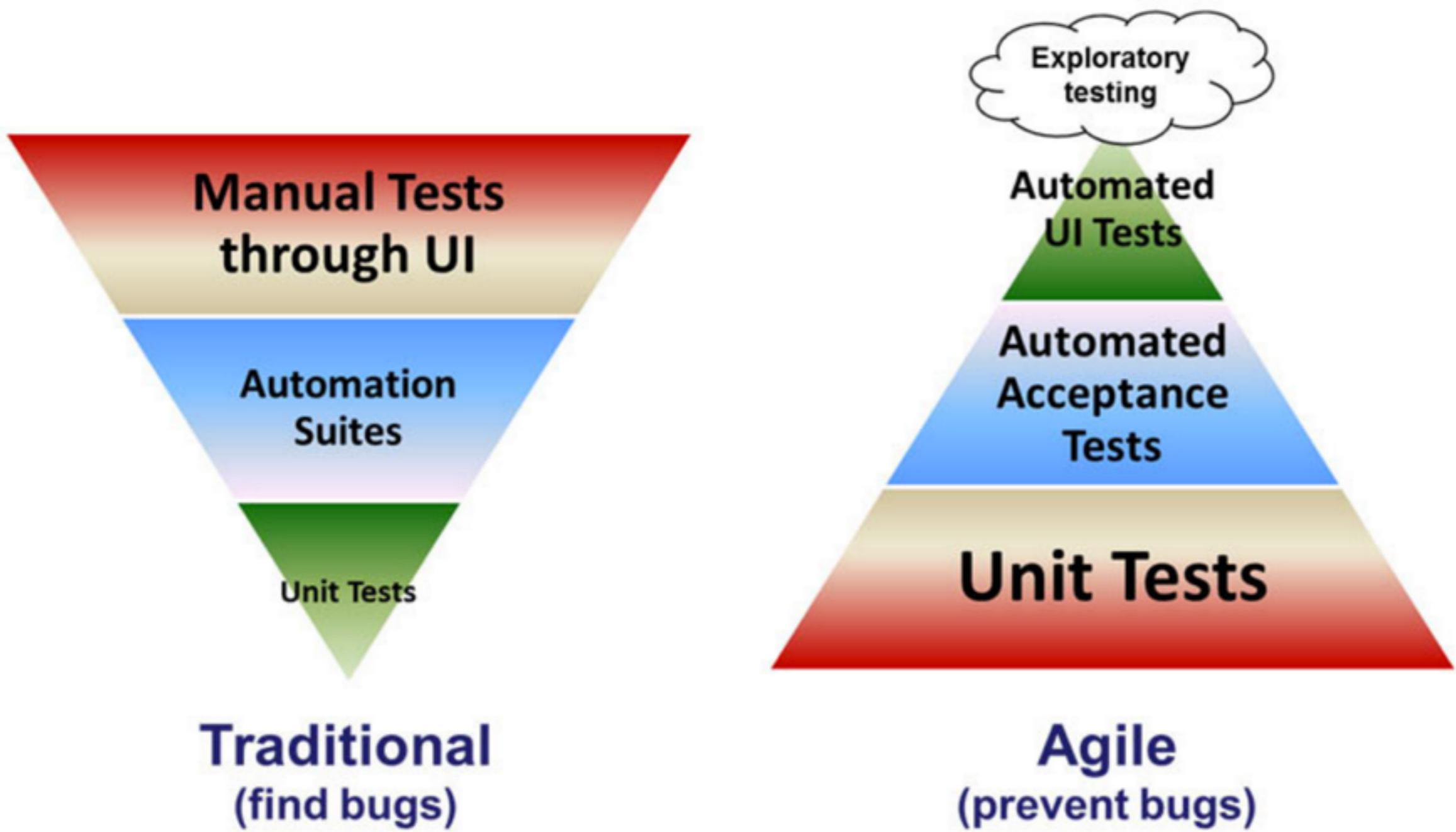




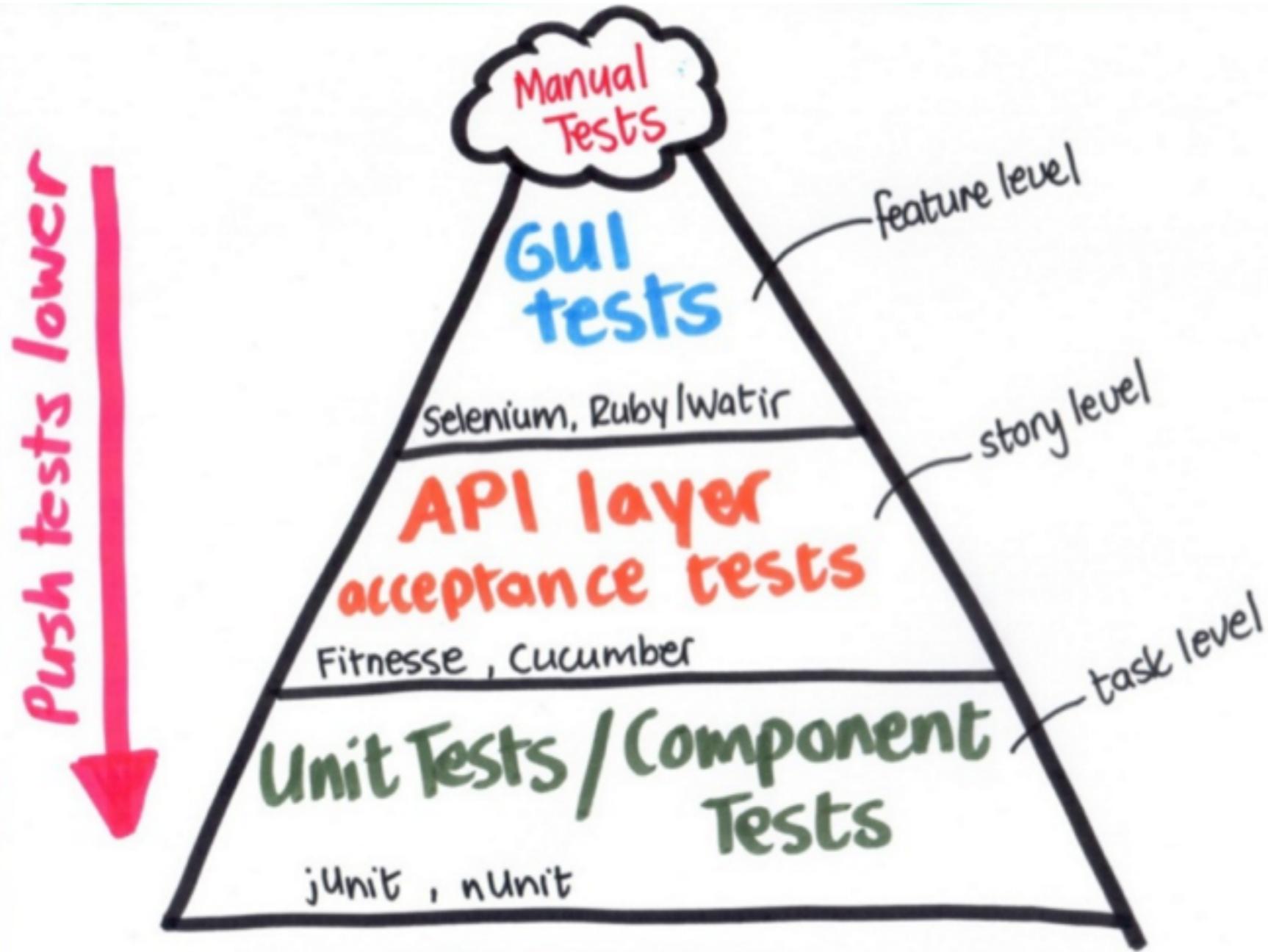
<https://www.thoughtworks.com/insights/blog/introducing-software-testing-cupcake-anti-pattern>



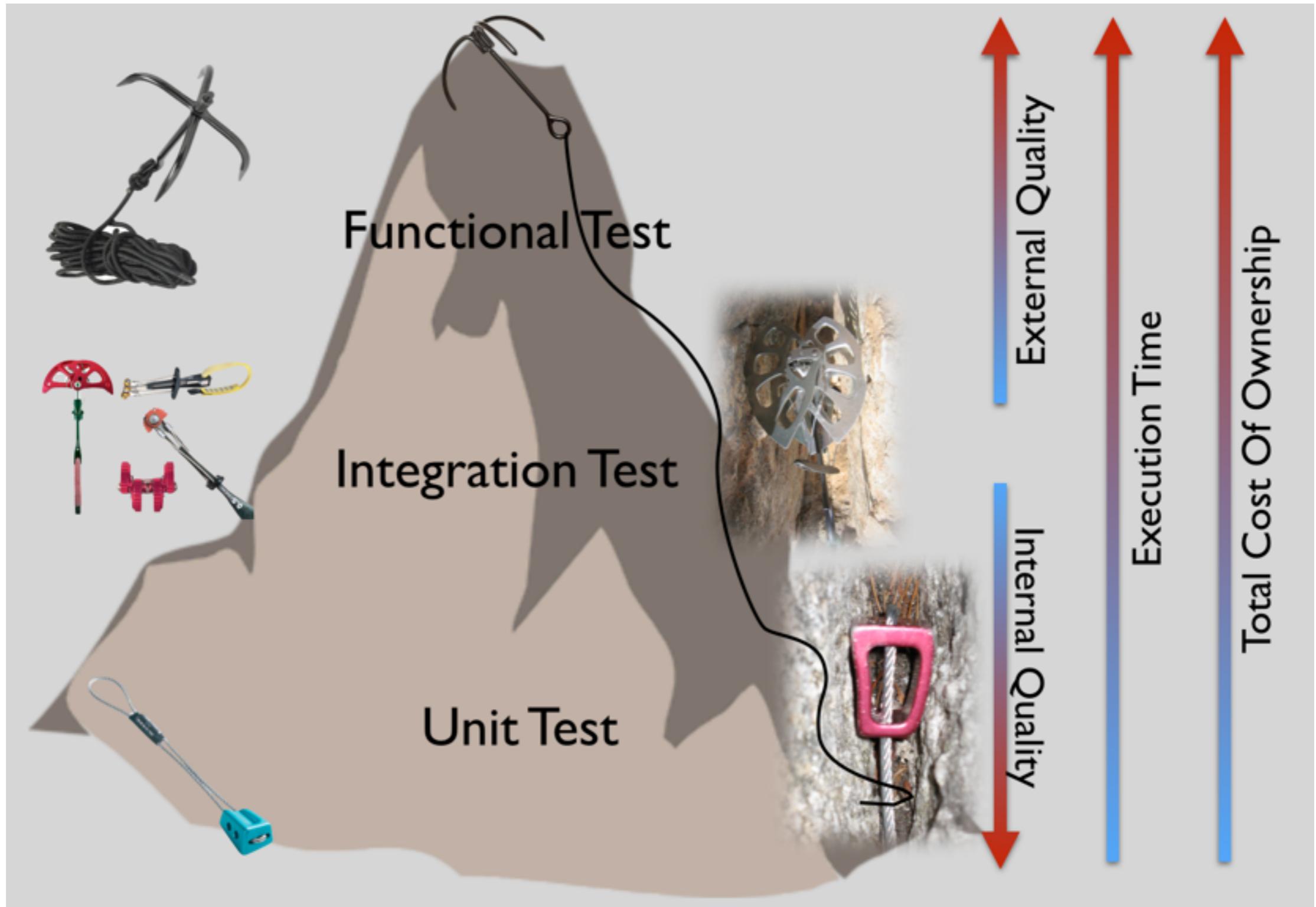
<https://www.thoughtworks.com/insights/blog/introducing-software-testing-cupcake-anti-pattern>



<http://testingtweaks.com/what-is-agile-testing/>

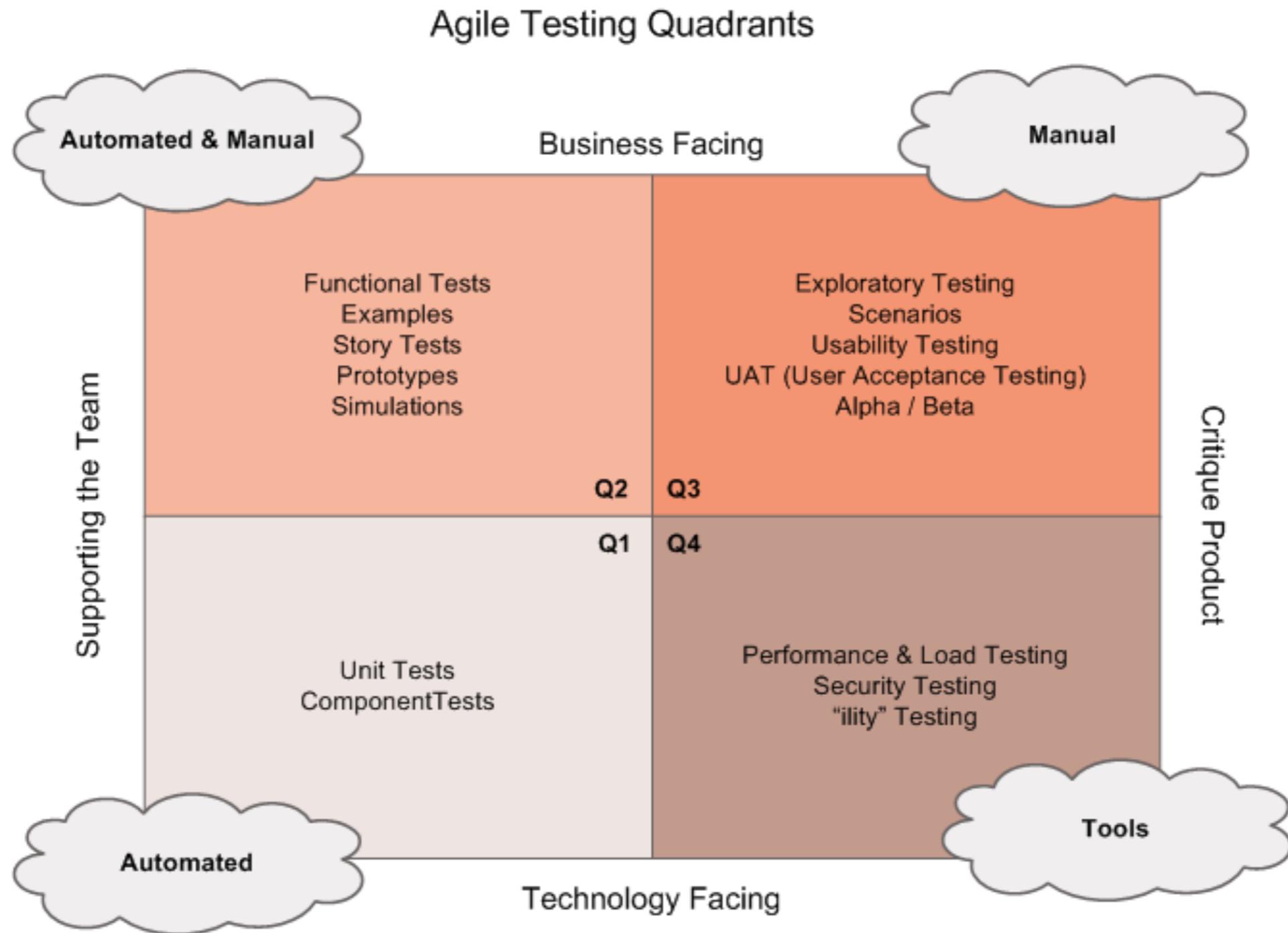


<http://www.slideshare.net/growingagile/expoqa-tutorial-agile-testing-techniques-for-the-whole-team>



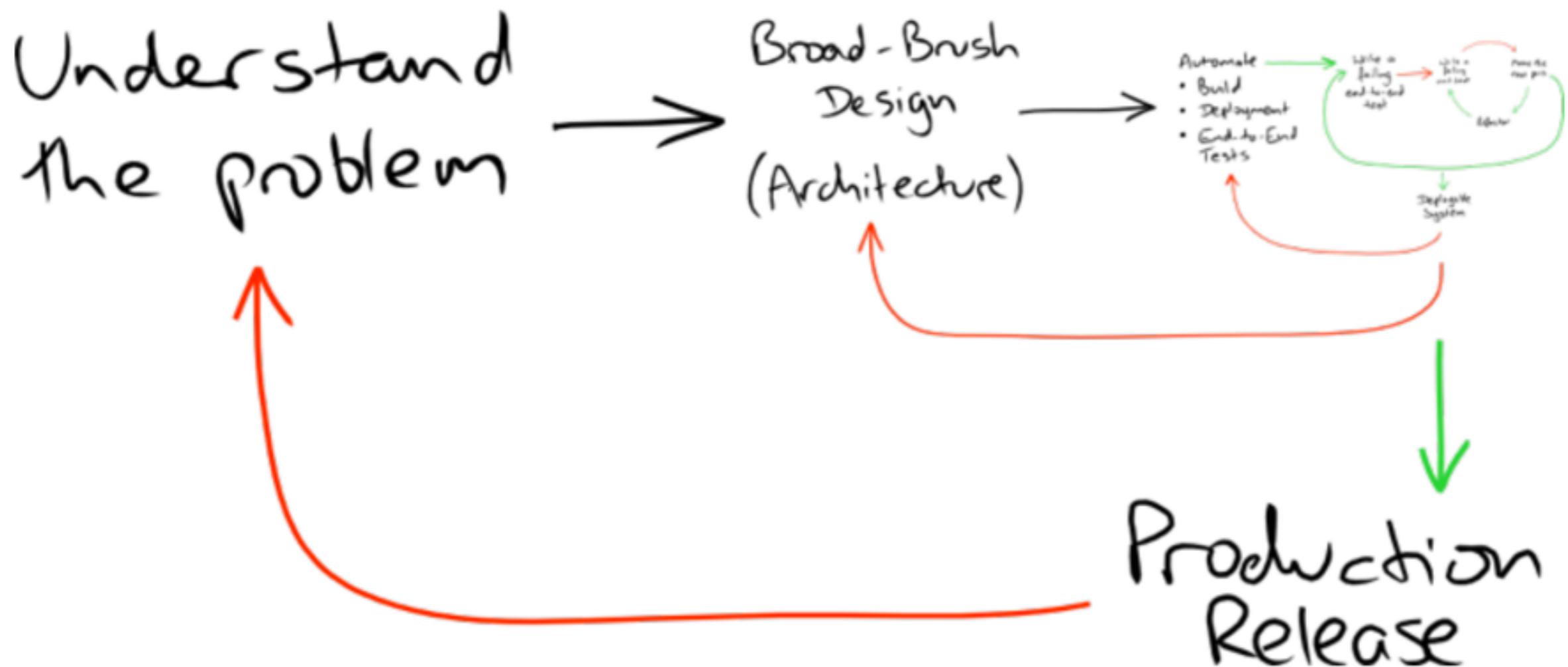
<http://less.works/less/technical-excellence/unit-testing.html>

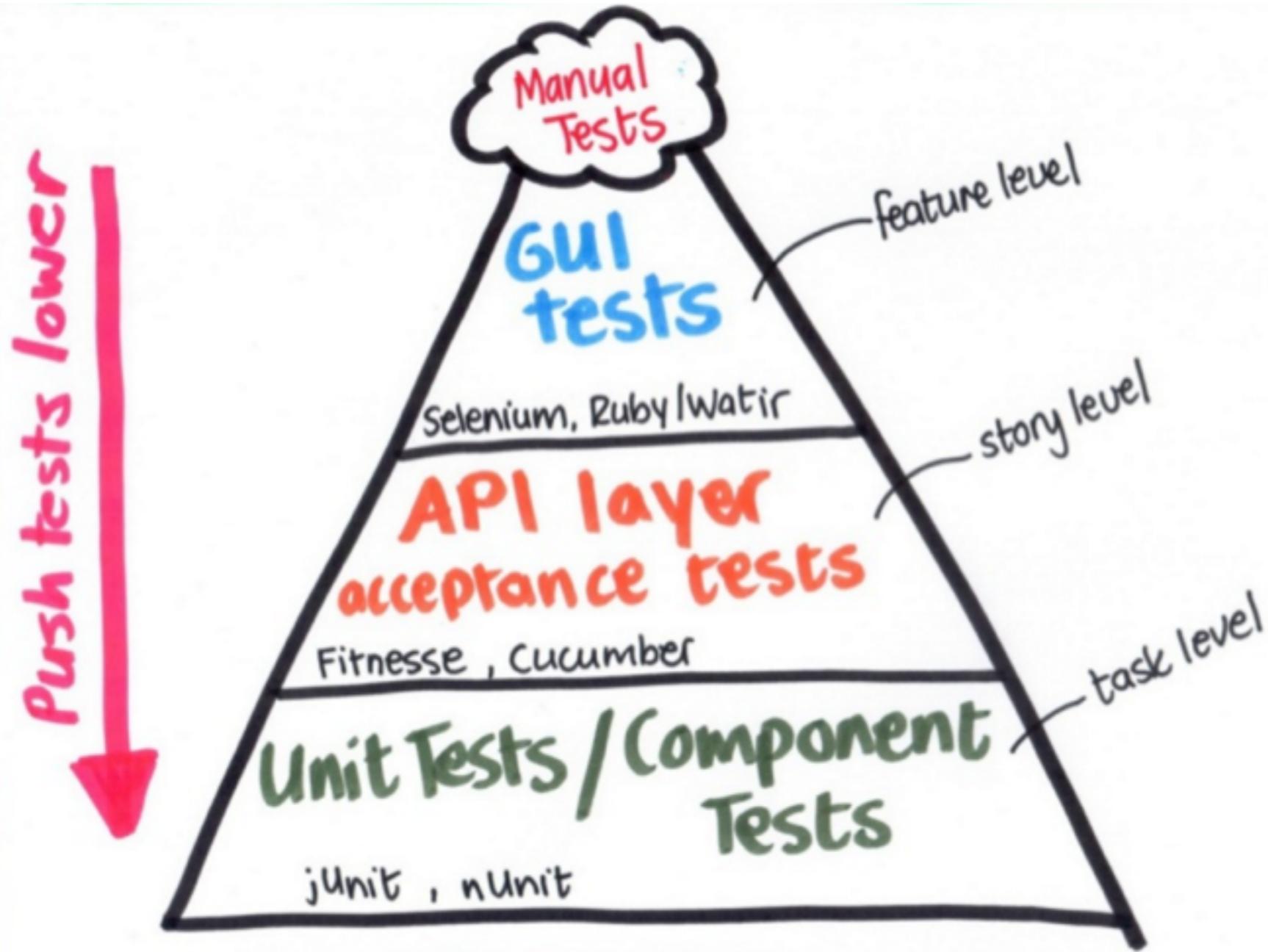
Agile Testing Quadrants



เริ่มอย่างไรดี ?

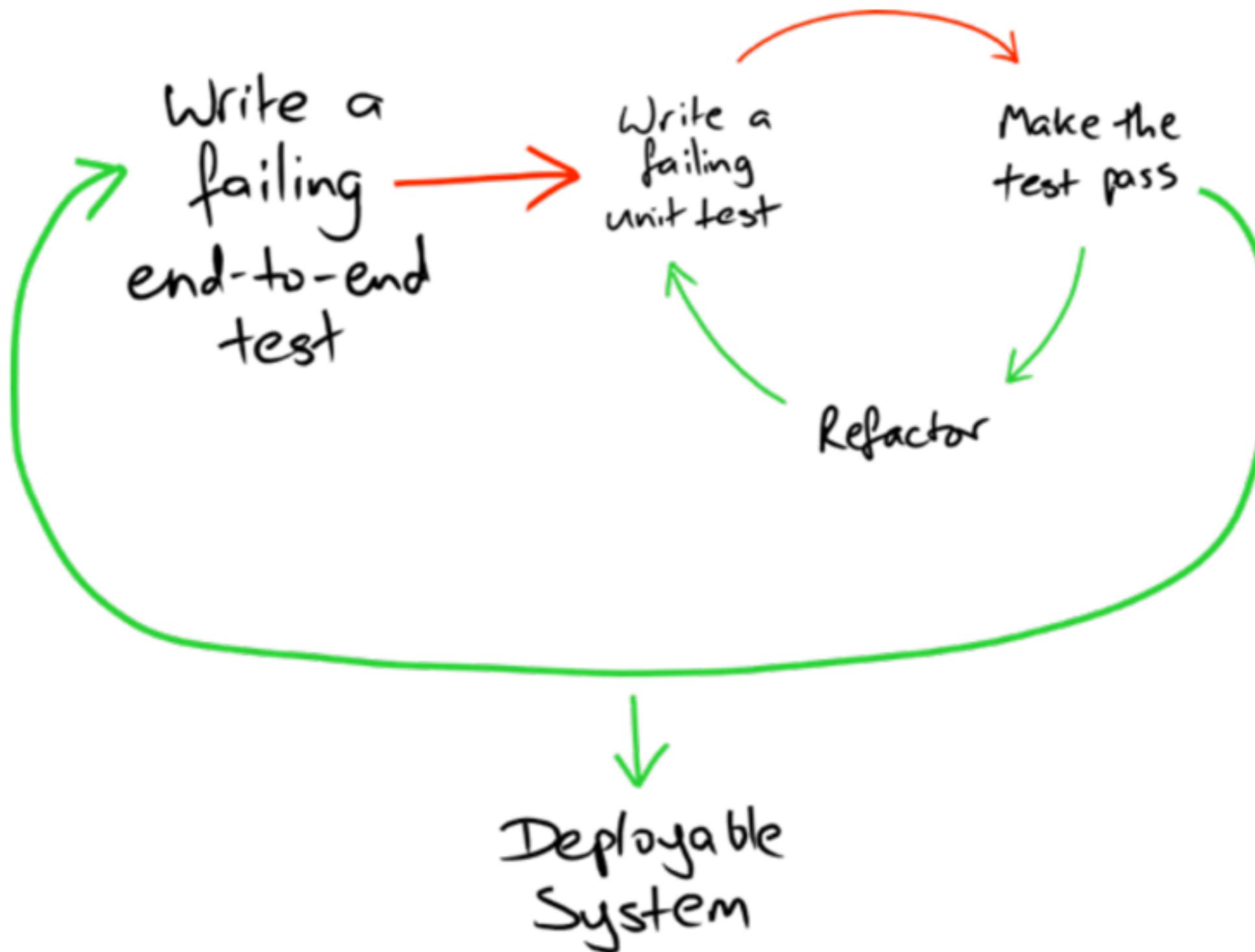
LARGER FEEDBACK LOOP

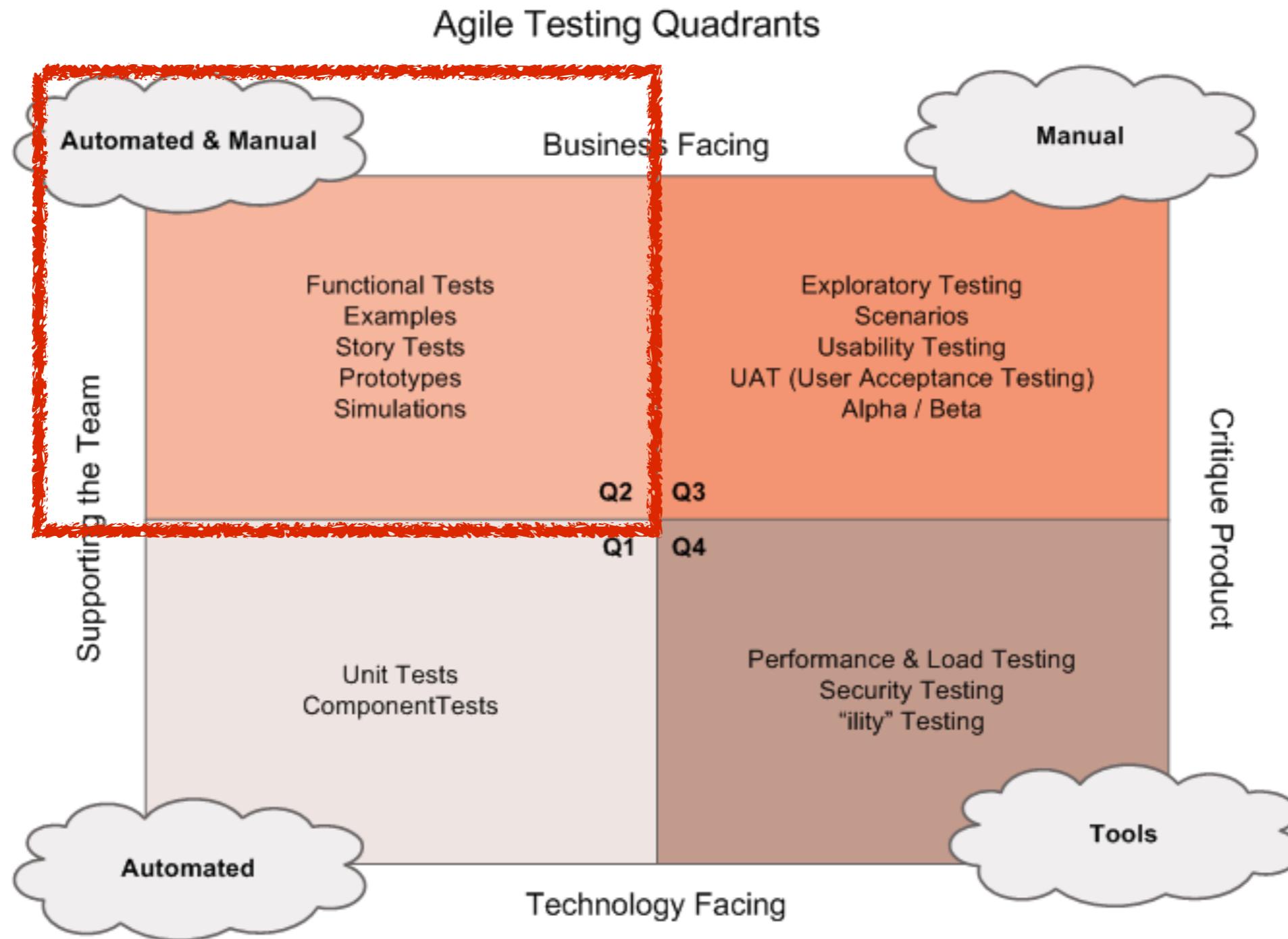




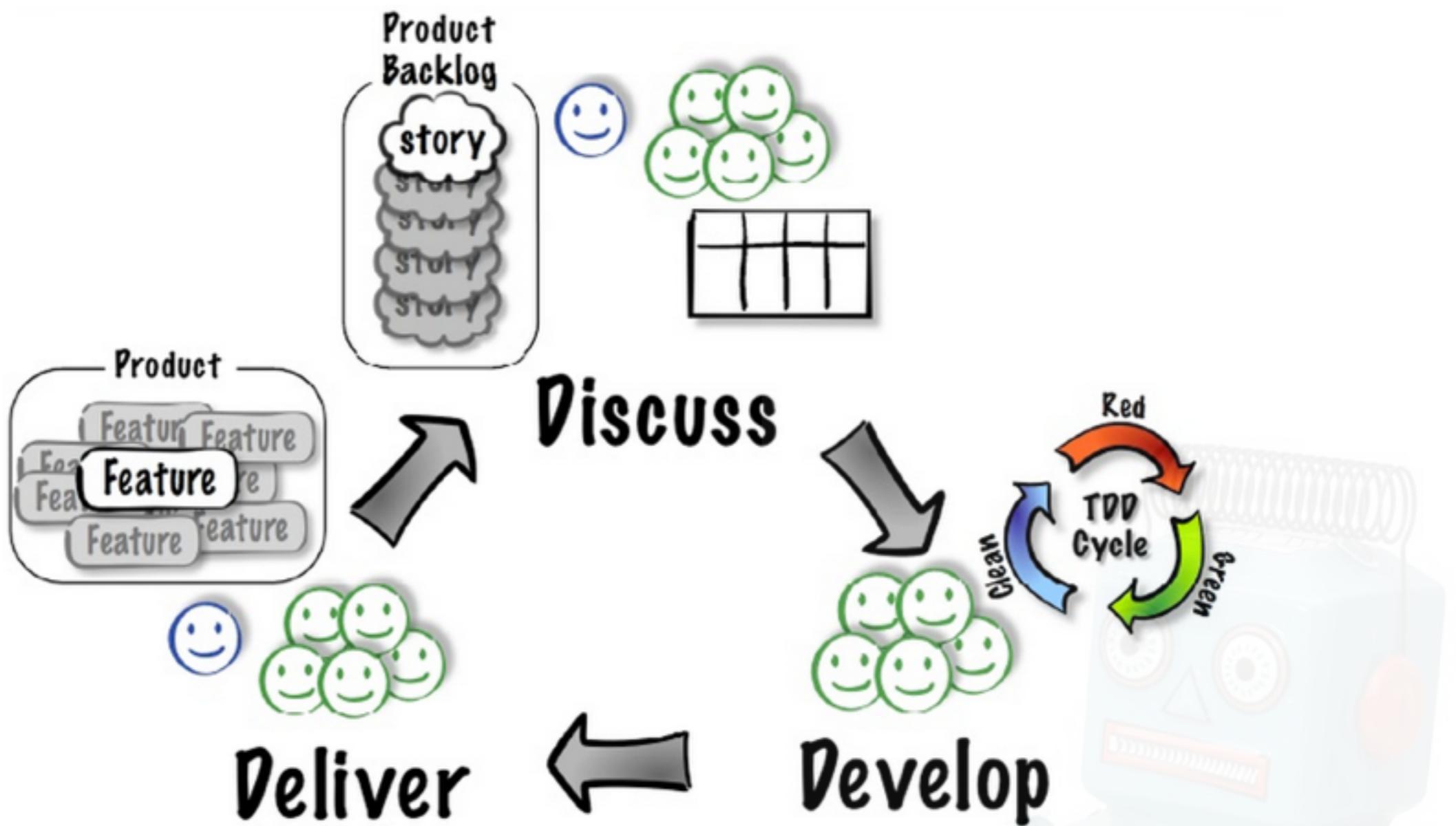
<http://www.slideshare.net/growingagile/expoqa-tutorial-agile-testing-techniques-for-the-whole-team>

ACCEPTANCE TEST DRIVEN DEVELOPMENT





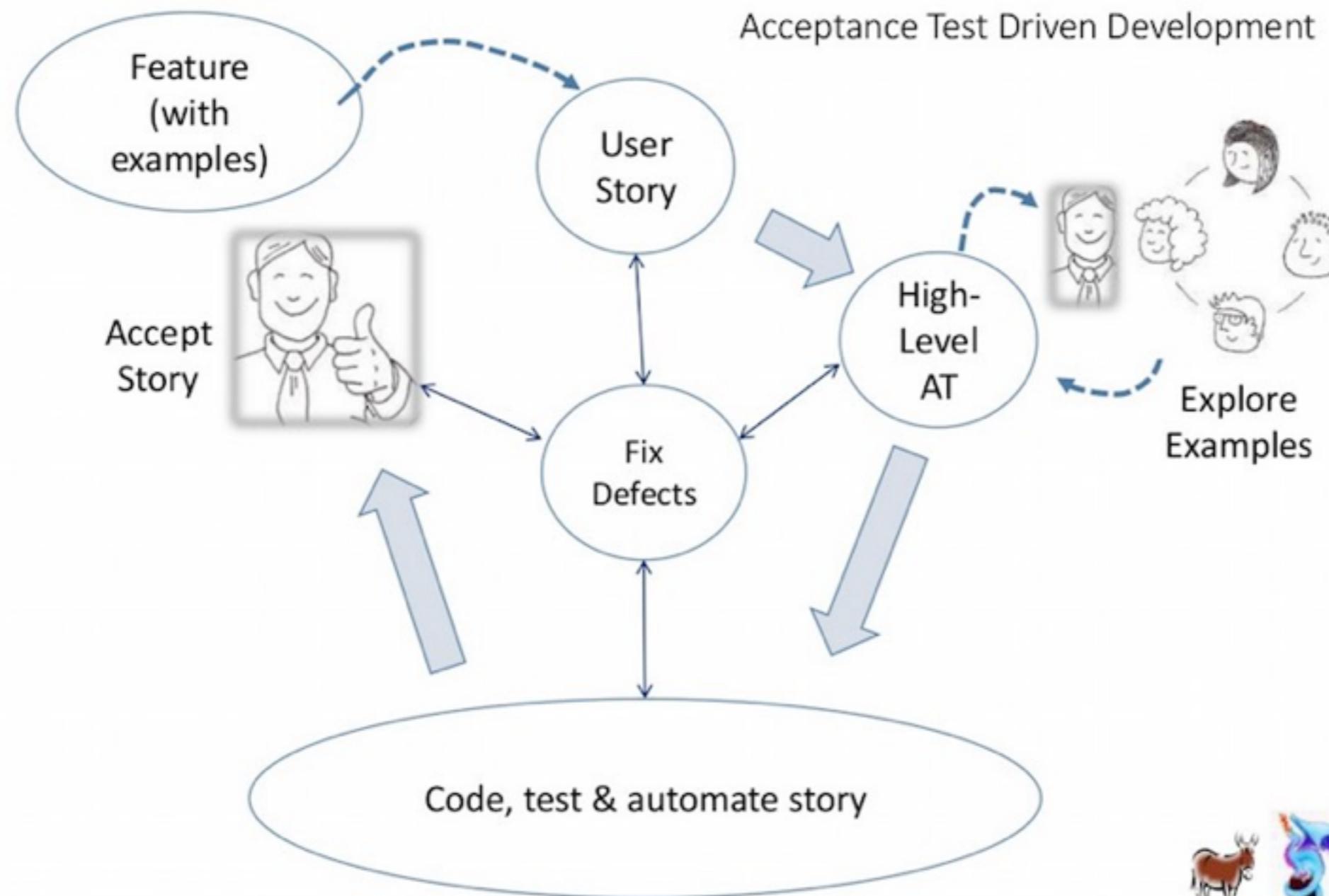
การทำงานของ ATDD



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

ATDD

Acceptance Test Driven Development



Workshop #1

Demo :: Grade Converter

Score :: 50

Convert to grade



Workshop #1

Demo :: Grade Converter

Score :: 50

Convert to grade

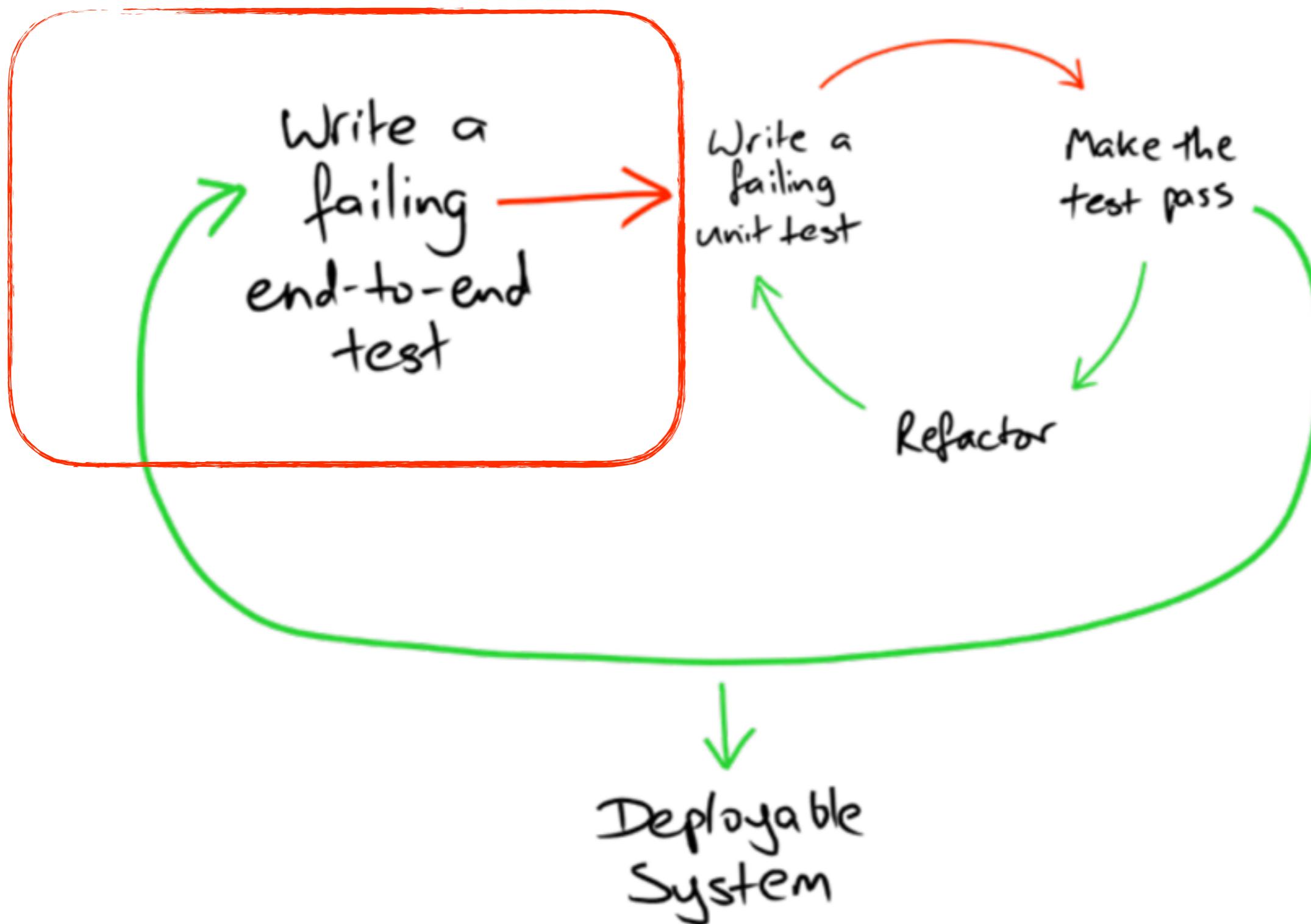
You got grade D

Workshop #1

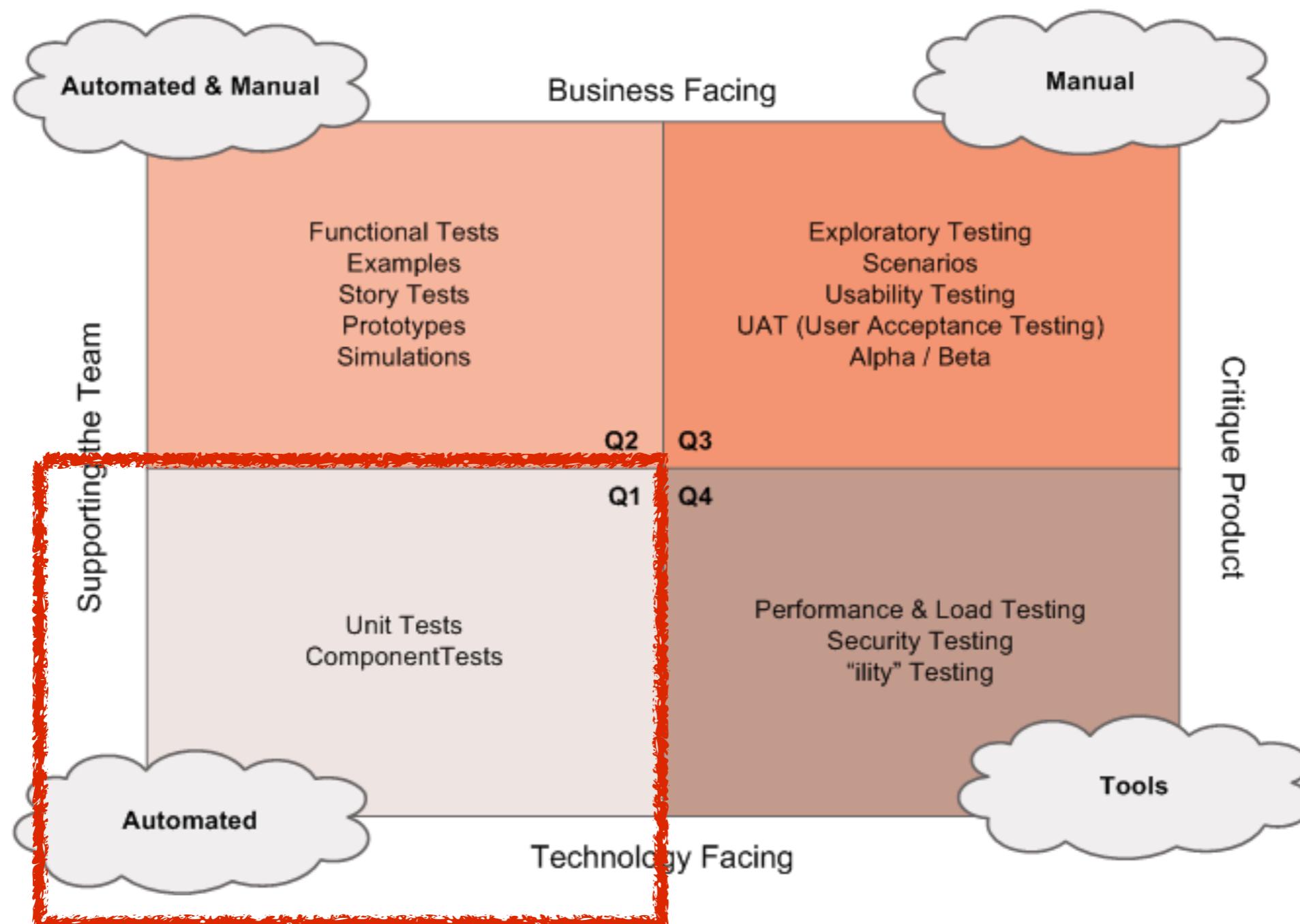
Test case = Biz rule + Test data



ACCEPTANCE TEST DRIVEN DEVELOPMENT



Agile Testing Quadrants

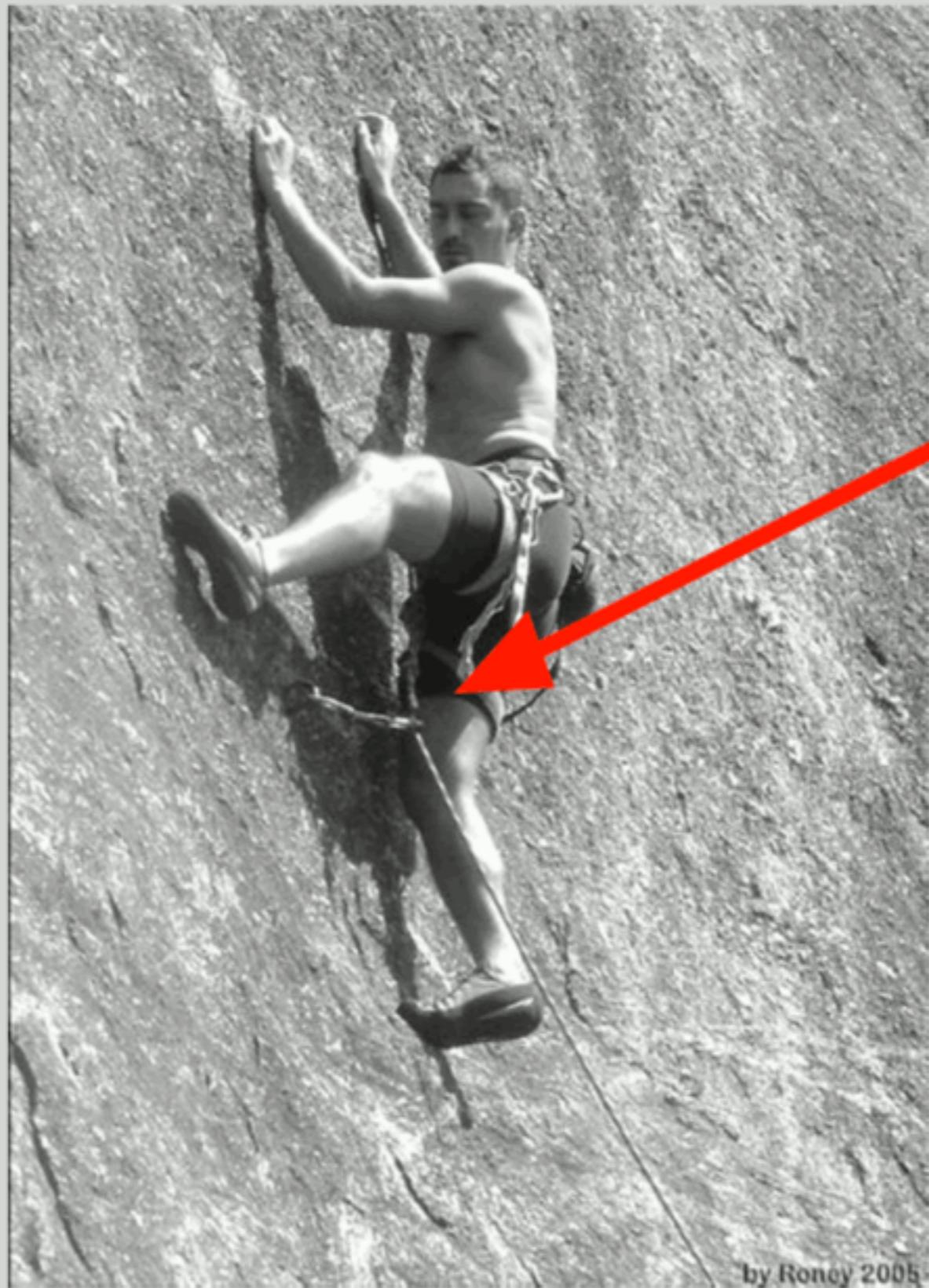


Unit Test คืออะไร ?

ส่วนหนึ่งของ code

ที่เรียกใช้งานส่วนอื่นๆ ของ code

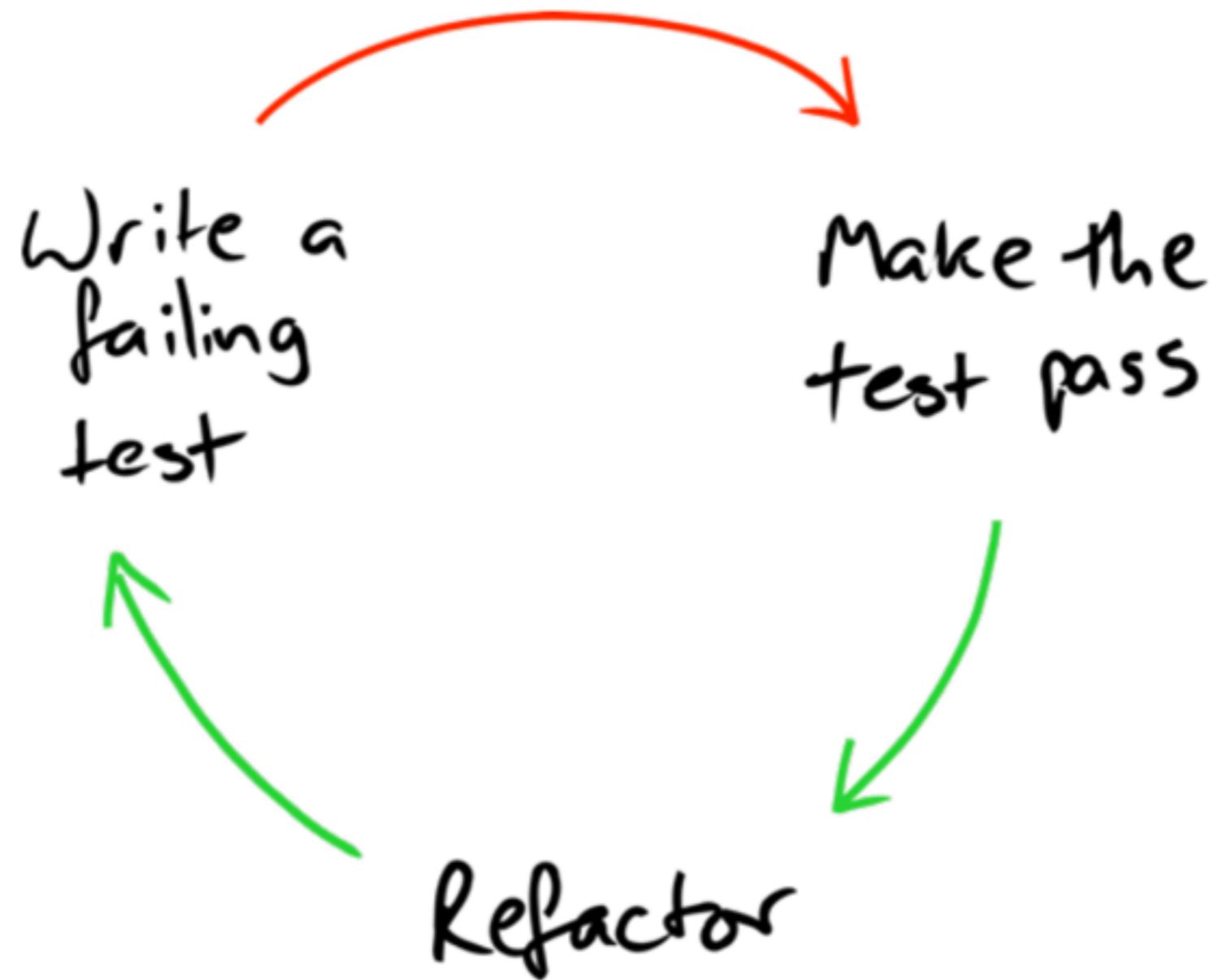
ใช้ตรวจสอบ สมมุติฐานว่าถูกต้องหรือไม่



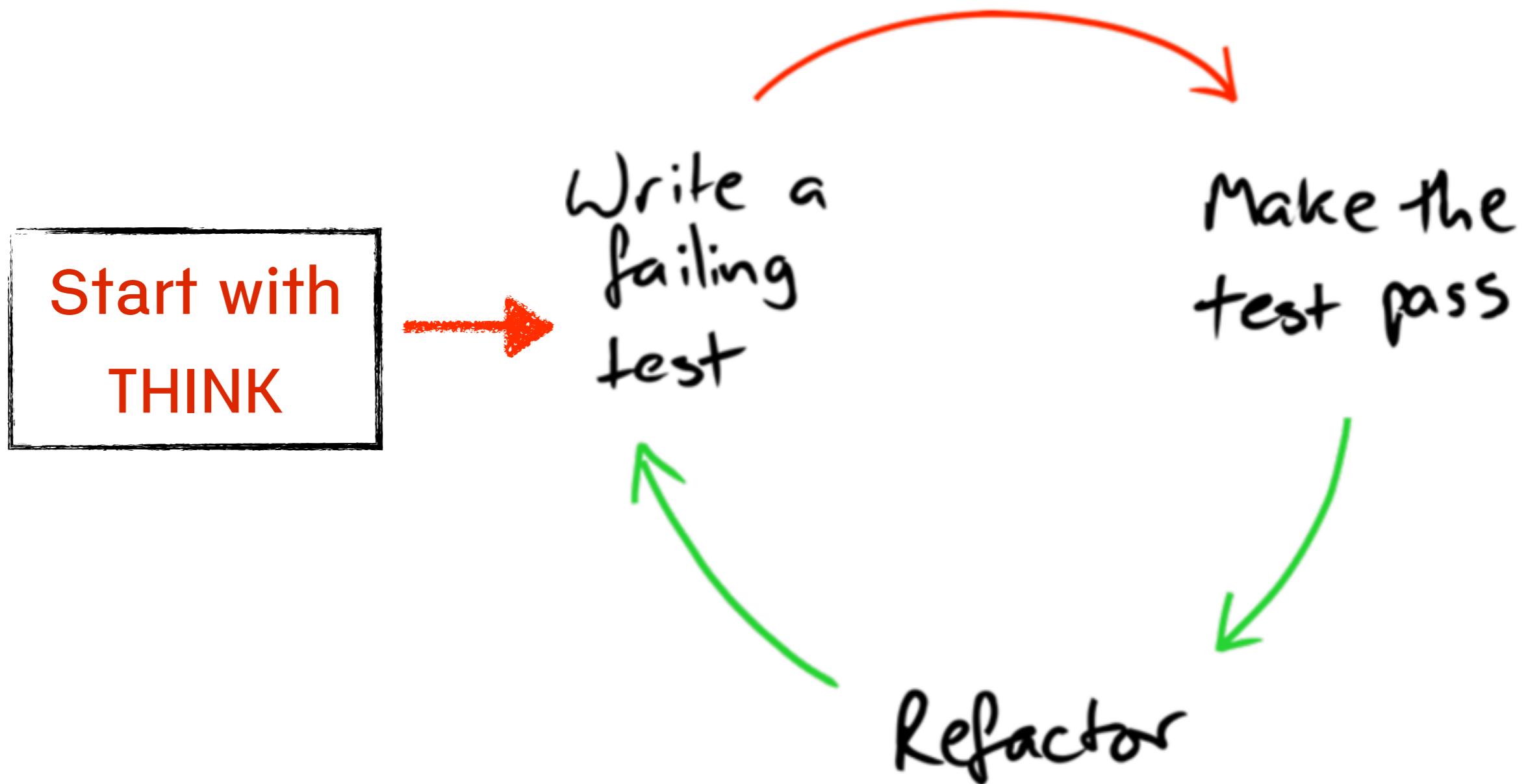
Unit Test

<http://less.works/less/technical-excellence/unit-testing.html>

TDD Cycle



Improve TDD Cycle



Simple TDD Rules

**Write a failing automated test before
you write any code**

Remove duplication

วิธีการทำให้ Test Pass

Fake or Hard code

Triangulation

Use obvious implementation

Why TDD ?

ทำไมต้อง TDD

เพื่อทำความเข้าใจกับปัญหาต่างๆ

คุณเข้าใจปัญหาหรือไม่ ?

ทำไมต้อง TDD

Software **เปลี่ยนแปลงอยู่เสมอ**

คนเรามัก**โน** และ **ทำผิดพลาดอยู่เสมอ**

ทำไมต้อง TDD

บอกคุณว่า ผิดตรงไหน

ไม่ต้องเสียเวลา **debug**

ทำไมต้อง TDD

ได้รับ feedback ที่รวดเร็ว

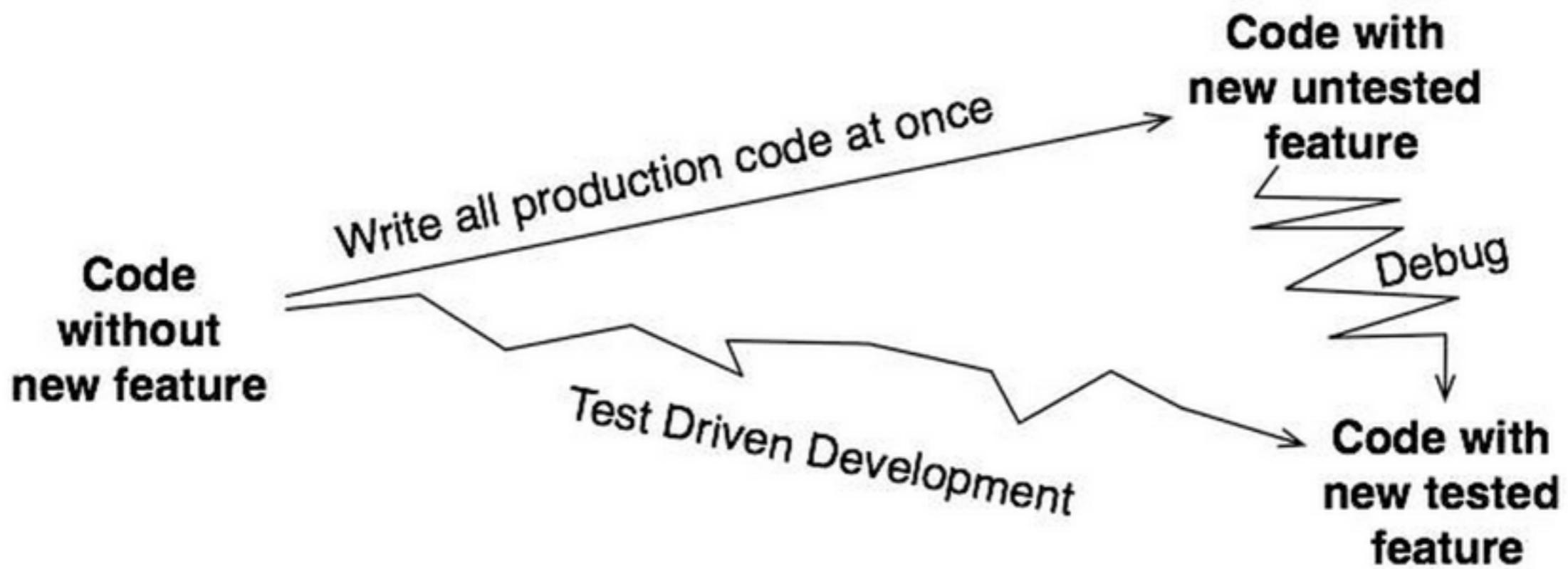
Many small changes vs One Big Change

ทำไมต้อง TDD

ให้ developer ทำงานเป็นทีมได้ดีขึ้น

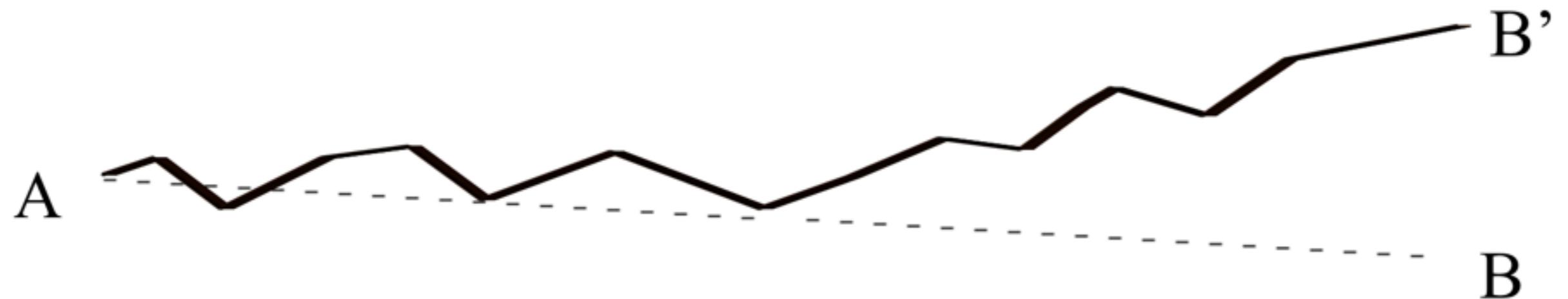
TDD vs DLP

DLP = Debug Later Programming



รูปจาก <http://www.renaissancesoftware.net/>

Remember :: Small step



Test ที่ดีต้อง FIRST

Fast
Independent
Repeatable
Self-validating
Timely

สิ่งที่ไม่ใช่ Unit test

External service

File system

Database

WebService

API calls

สิ่งที่ไม่ใช่ Unit test

Over specification

Compare screen image

Compare HTML

“What’s in a name?”

That which we call a rose

by any other name would smell

as sweet

Romeo and Juliet

Don't use the word “test”



Use the word “should”

```
transferShouldDeductSumFromSourceAccountBalance()
```

```
transferShouldAddSumLessFeesToDestinationAccountBalance()
```

```
depositShouldAddAmountToAccountBalance()
```

Guide to Test Writing

Test class names are important too

Structure your test well

Tests are **deliverable** too

Test Structure



The ratio of time spent (code)
versus writing is
over 10 to 1

Robert C. Martin, Clean Code

Good Unit Test

```
@Test  
public void sayFizzWhenNumberIsDevidedByThree() {  
Arrange   FizzBuzz fizzBuzz = new FizzBuzz();  
Act       String actualResult = fizzBuzz.say(3);  
Assert    assertEquals("Fizz", actualResult);  
}
```



Workshop #1

Grade Calculator

Test coverage

Packages

All
[demo.grade](#)
[launch](#)

All Packages

Classes

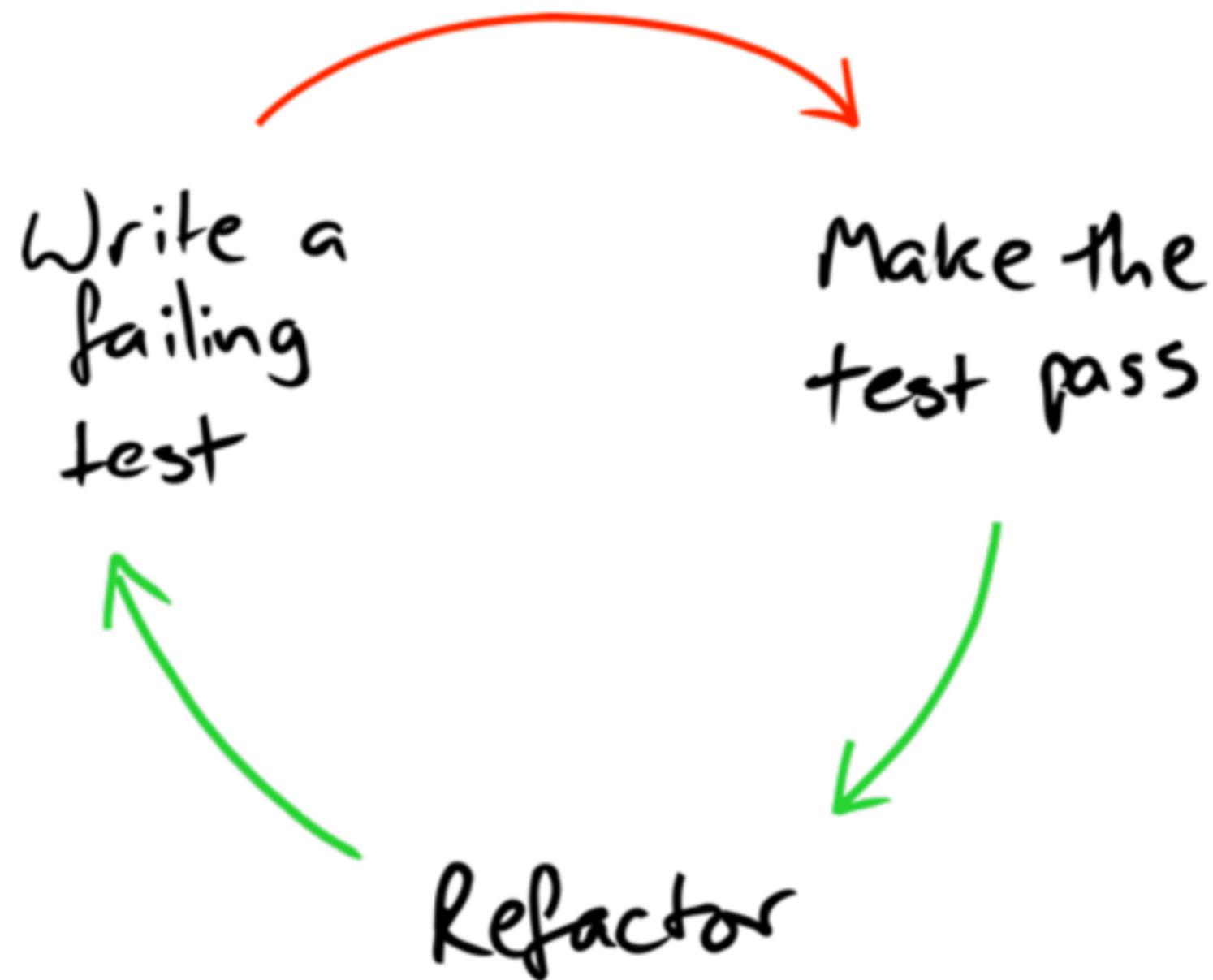
[GradeController](#) (0%)
[Main](#) (0%)

Coverage Report - All Packages

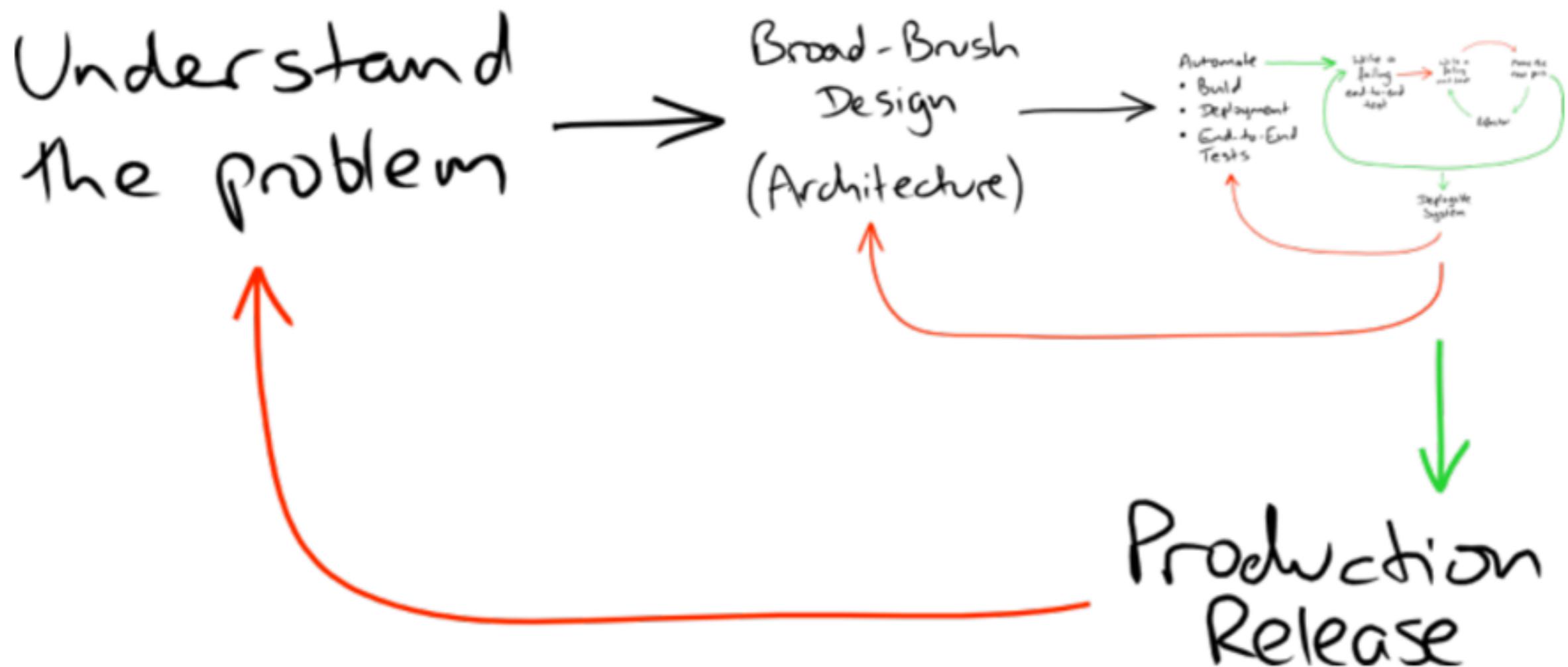
| Package | # Classes | Line Coverage | Branch Coverage | Complexity |
|----------------------------|-----------|---------------|-----------------|------------|
| All Packages | 2 | 0% 0/27 | 0% 0/8 | 2 |
| demo.grade | 1 | 0% 0/14 | 0% 0/4 | 1.667 |
| launch | 1 | 0% 0/13 | 0% 0/4 | 3 |

Report generated by [Cobertura](#) 2.0.3 on 3/23/16 12:10 PM.

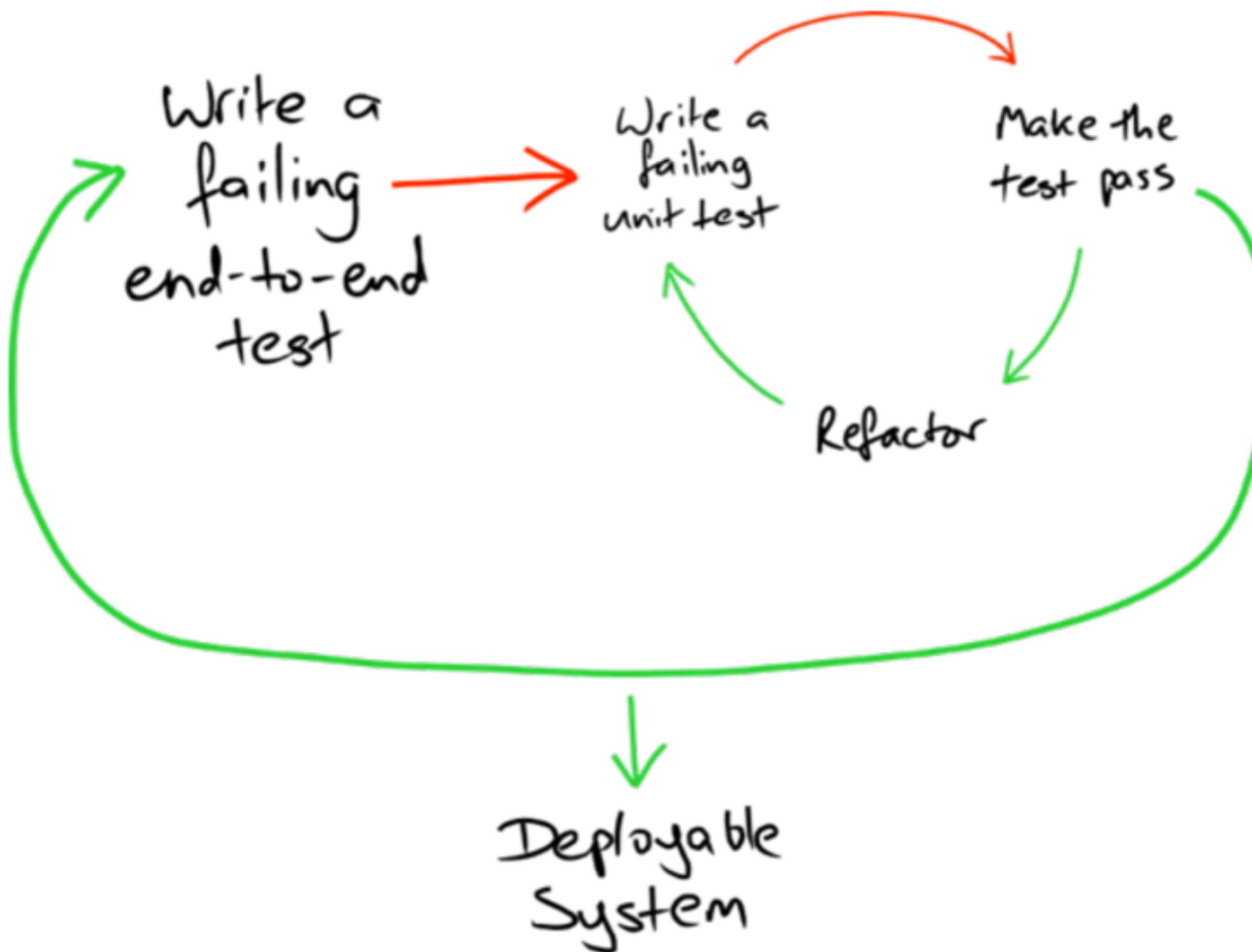
TEST DRIVEN DEVELOPMENT

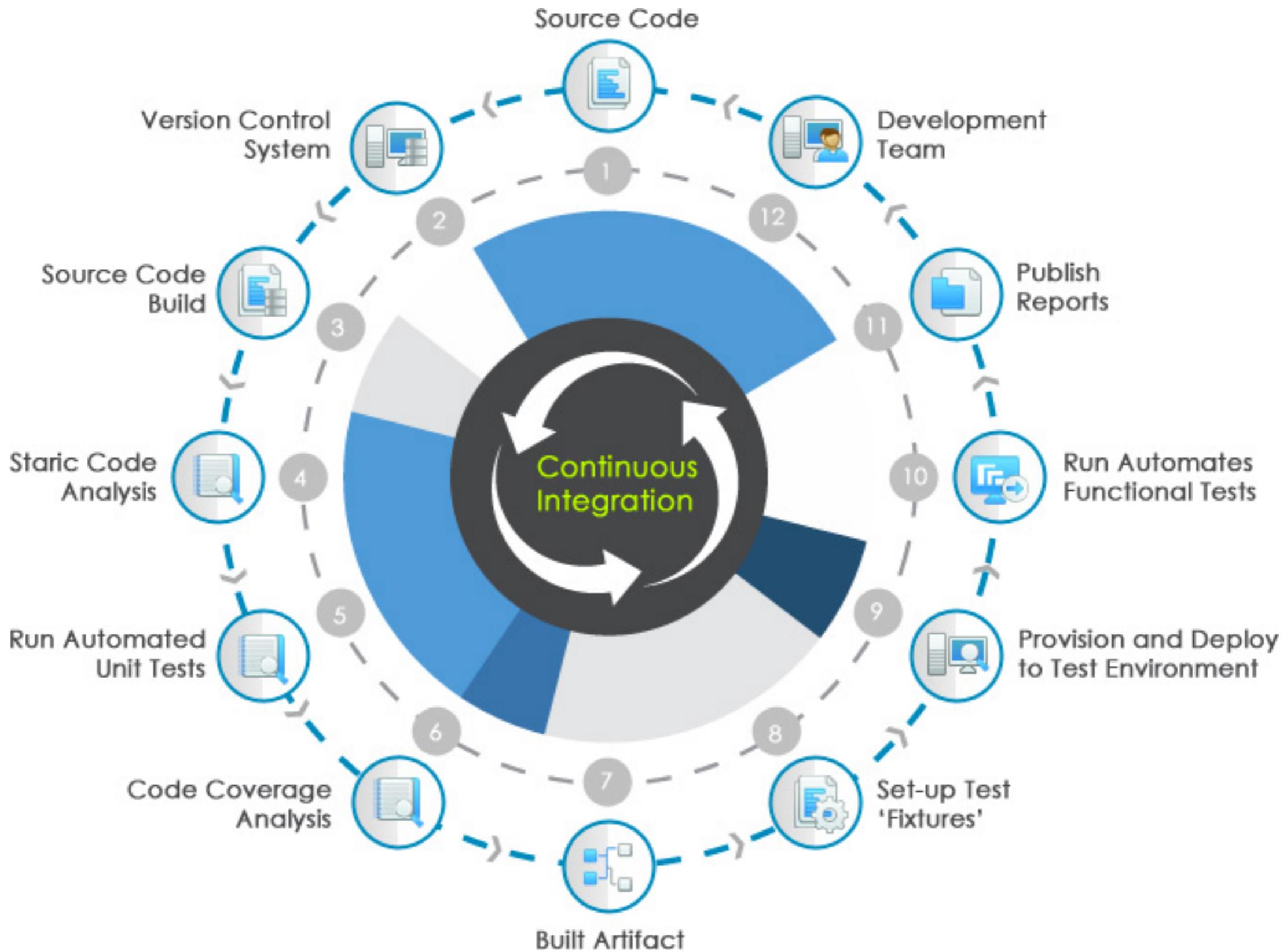


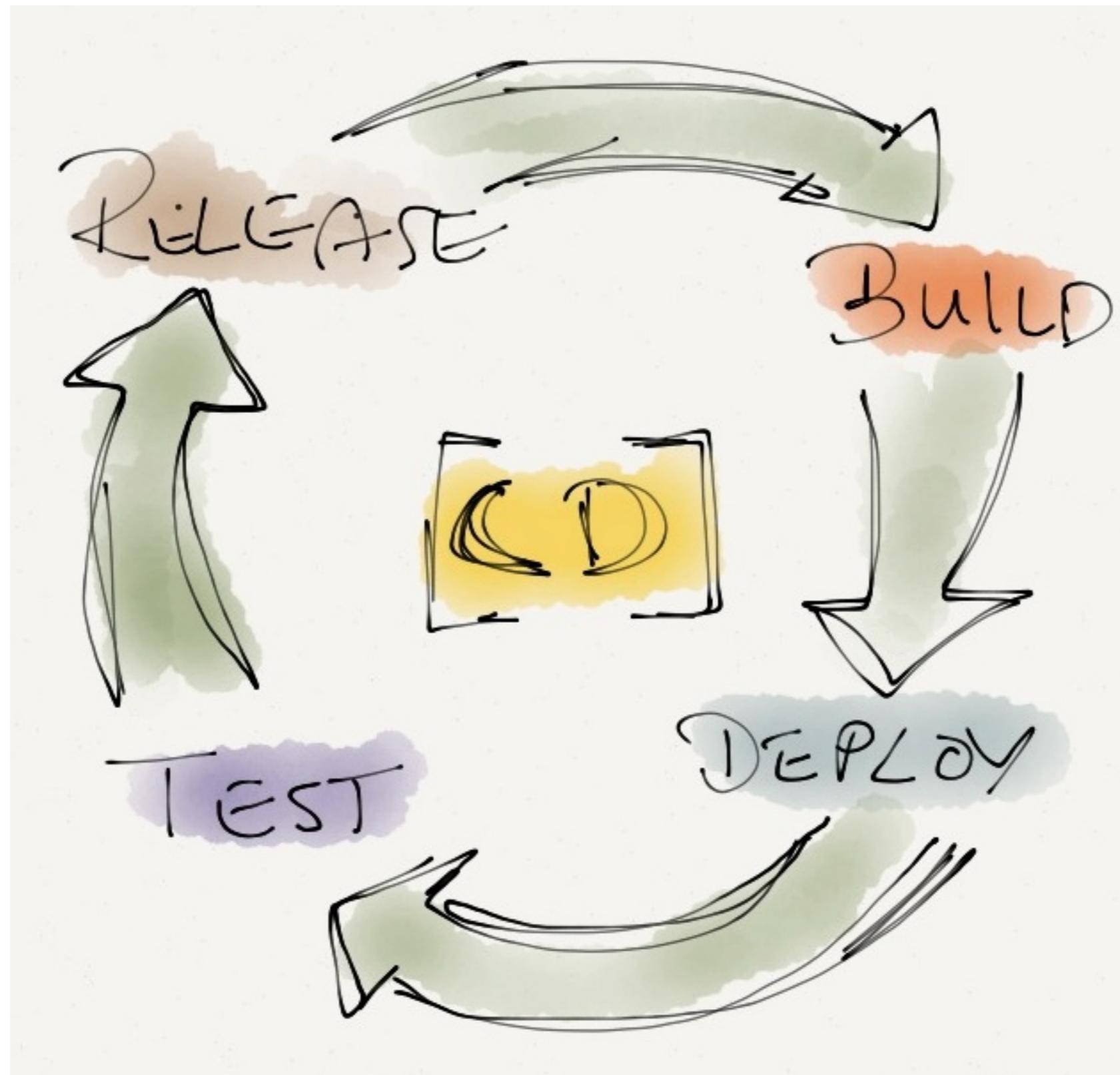
LARGER FEEDBACK LOOP



ACCEPTANCE TEST DRIVEN DEVELOPMENT







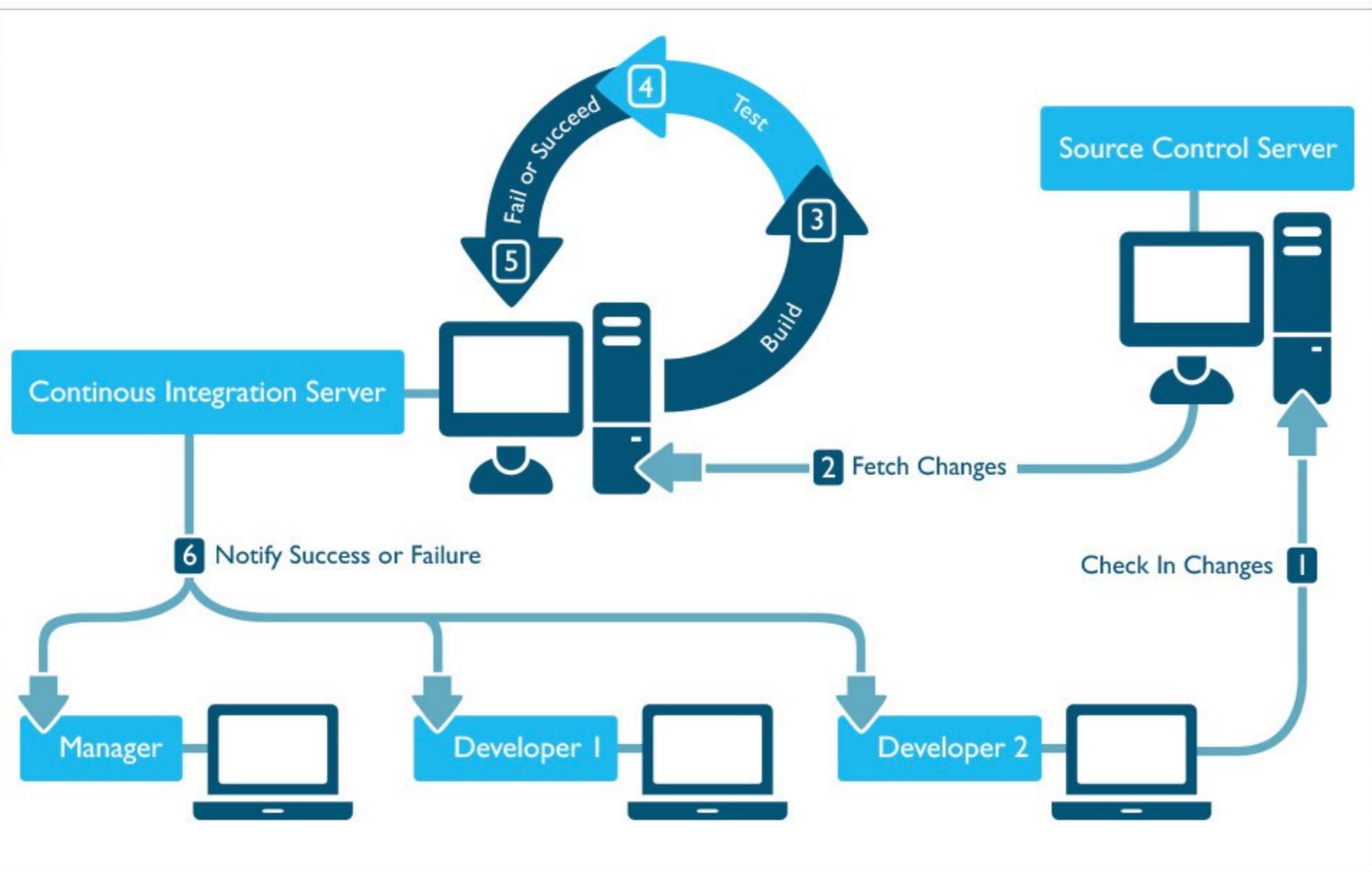
CONTINUOUS DELIVERY

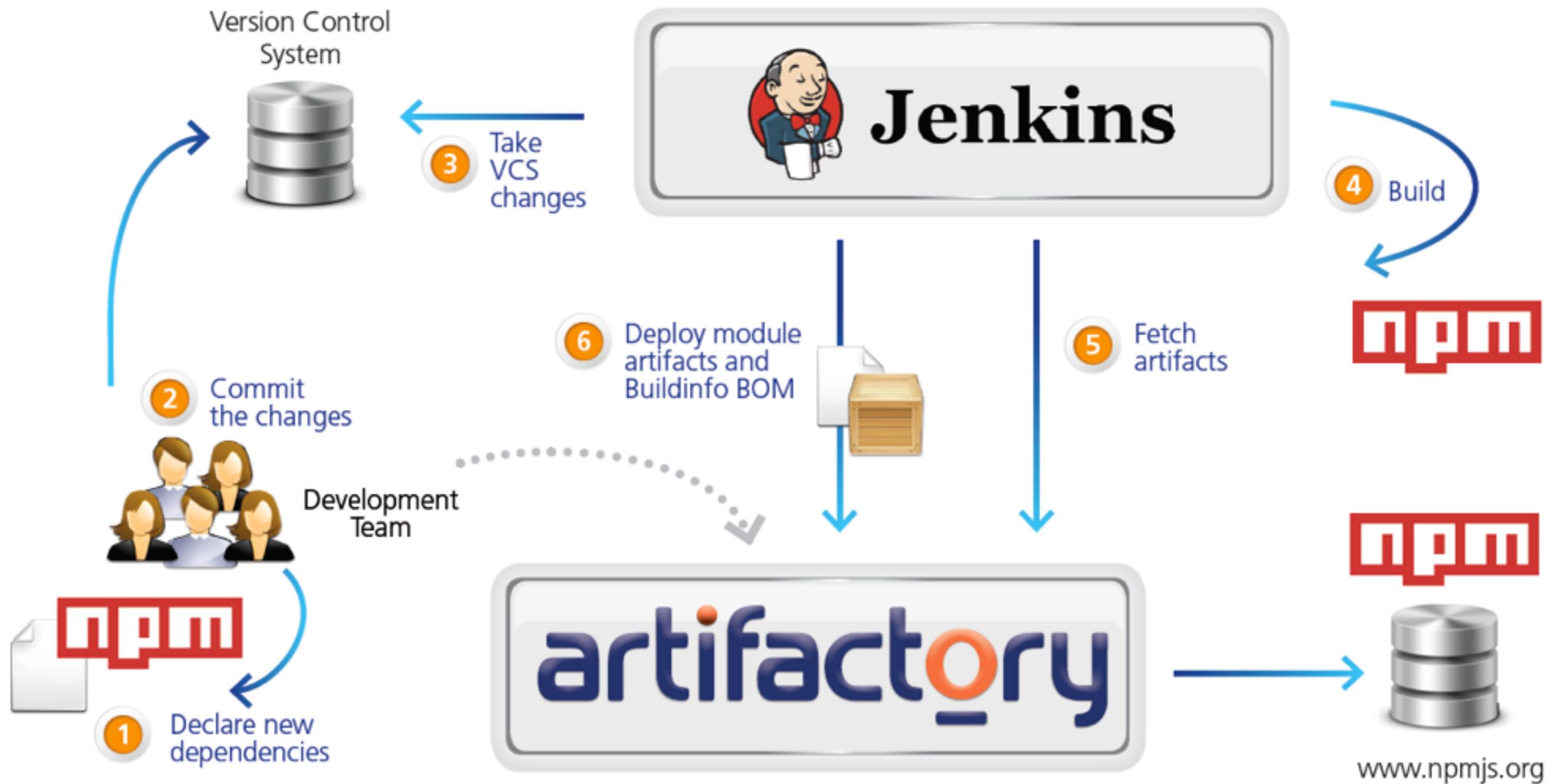


CONTINUOUS DEPLOYMENT



<http://blog.crisp.se/2013/02/05/yassalsundman/continuous-delivery-vs-continuous-deployment>



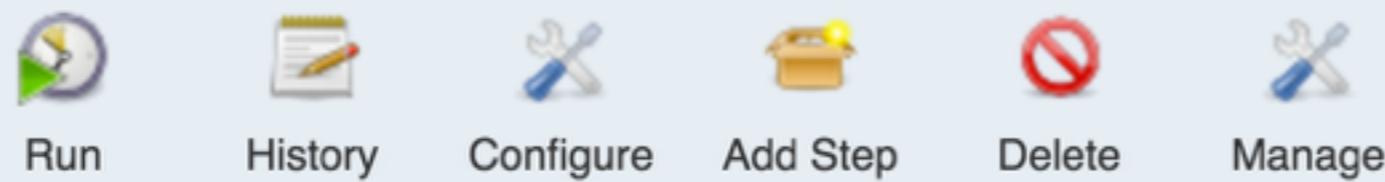


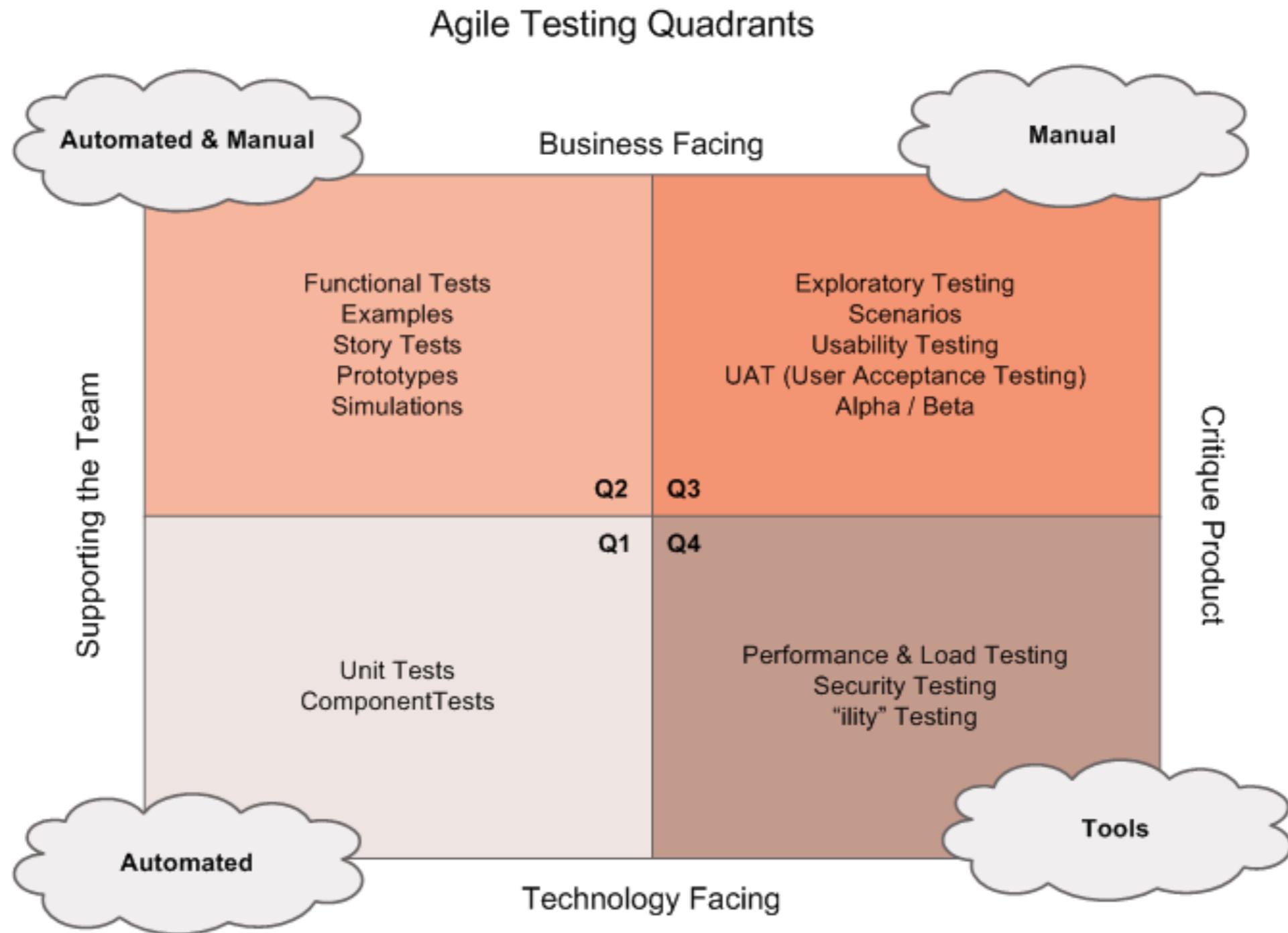
Workshop #3

Continuous Integration

Build pipeline

Build Pipeline





Are you too busy to improve?





บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่



LEGACY CODE

บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

“Quality is the responsibility of everybody.”