



Agile
Firestarter

Introduction to Test-Driven Development



Alex Hung
CTO

Ballyhoo Software

Agenda

Why Test-Driven Development?

What is It?

Demo & Lab

Why Test-Driven Development is “agile”?



Transmission change \approx days

Why Test-Driven Development is “agile”?



Transmission + rear suspensions + rear subframe
change \approx 3.5 hours!

How is Test-Driven Development “agile”?



Transmission change \sim days



Transmission + rear suspensions +
rear subframe change \sim 3.5
hours!

How is Test-Driven Development “agile”?

Ability to responds to change

TDD allows code to be changed/refactored quickly, easily, and **safely** in response to change in requirement, architecture, etc.

How is Test-Driven Development “agile”?

Feedback on change

TDD provides quick feedback on code changes,
especially non-obvious, unexpected effect.

The Rationale for Unit Tests

Non-Debatable Software Engineering Facts:

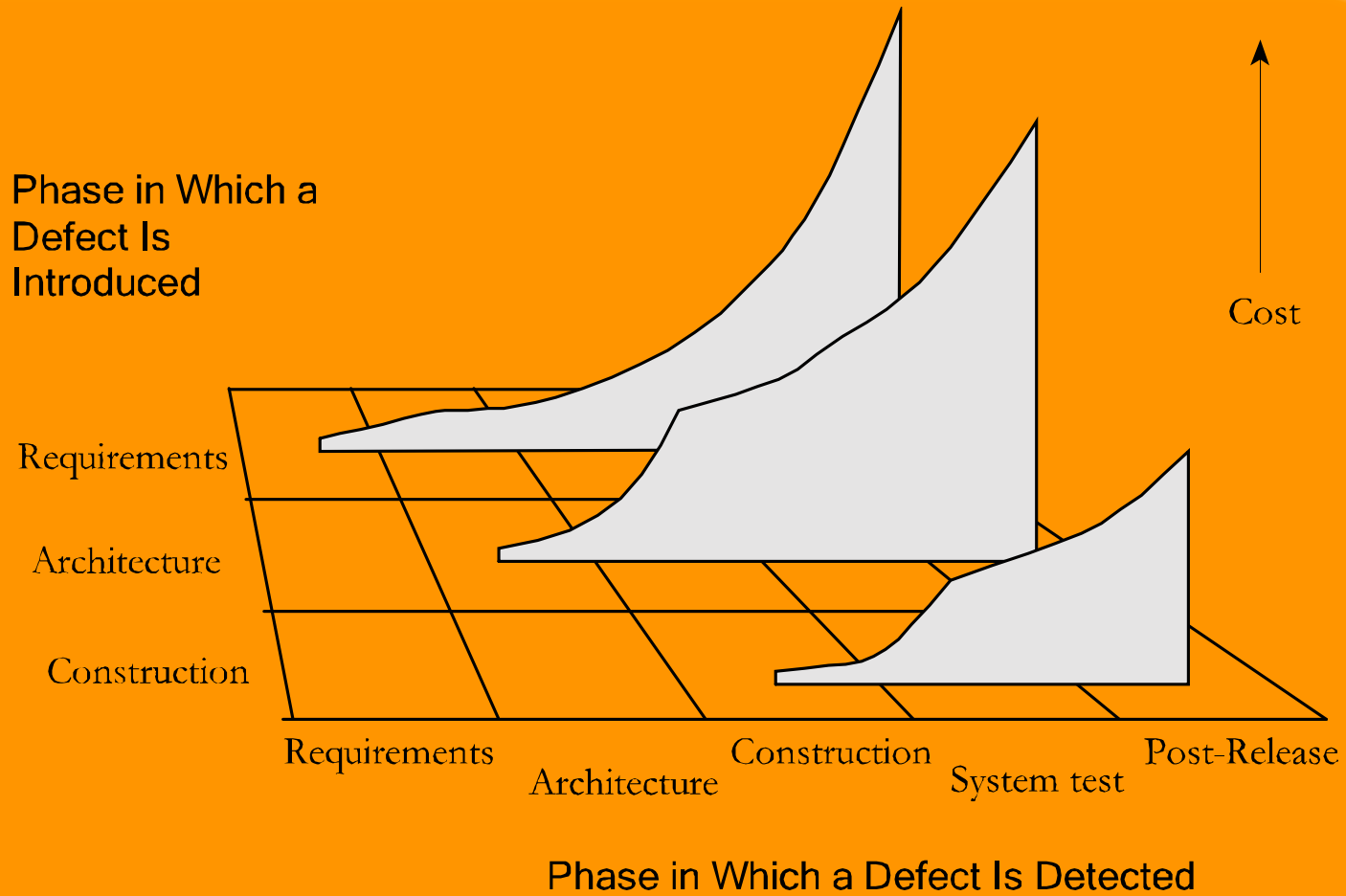
There will always be bugs

Complex programs have more bugs than simple programs

Code is more maintainable when its divided into bite-sized chunks

The cost of fixing a bug escalates non-linearly over time as the project progresses

'Code Complete' Defect Cost Graph



What is Unit Testing?

Primary purpose is to test public interface of one or more classes

As Isolated as possible (hence 'unit')

Independent of other system components

If your test touches something else you didn't develop but your app depends on to run (database, web service, etc.) then it's not a unit test

Sandboxed

No side-effects to the environment allowed

Typically run via a unit test framework (NUnit, MbUnit, etc.)

What is Test-Driven Development?

Testing *while* coding

not *before* or *after*

What is Test-Driven Development?

Think “Test-Driven Design”

or

“Specification-Driven-Development”

Consider your design from the perspective of the consumers of your methods, classes, etc.

Outside-In design instead of Inside-Out design

What is Test-Driven Development?

“How-Do-I-Use-This-Driven-Development”

For every line of code you write, ask yourself a simple (but powerful) question:

“HOW WILL I TEST THAT?”



Wait, why do you mean by that?!

CODE EXAMPLE

Benefits of TDD

Every class and method immediately has at least **TWO** consumers:

Your **APP** and your **TEST!**

Better-isolated code leads to easier to extend, enhance, replace, maintain (lifecycle costs)

Benefits of TDD

Less likely to write code that you eventually don't need

Write nothing that you *think* you will need

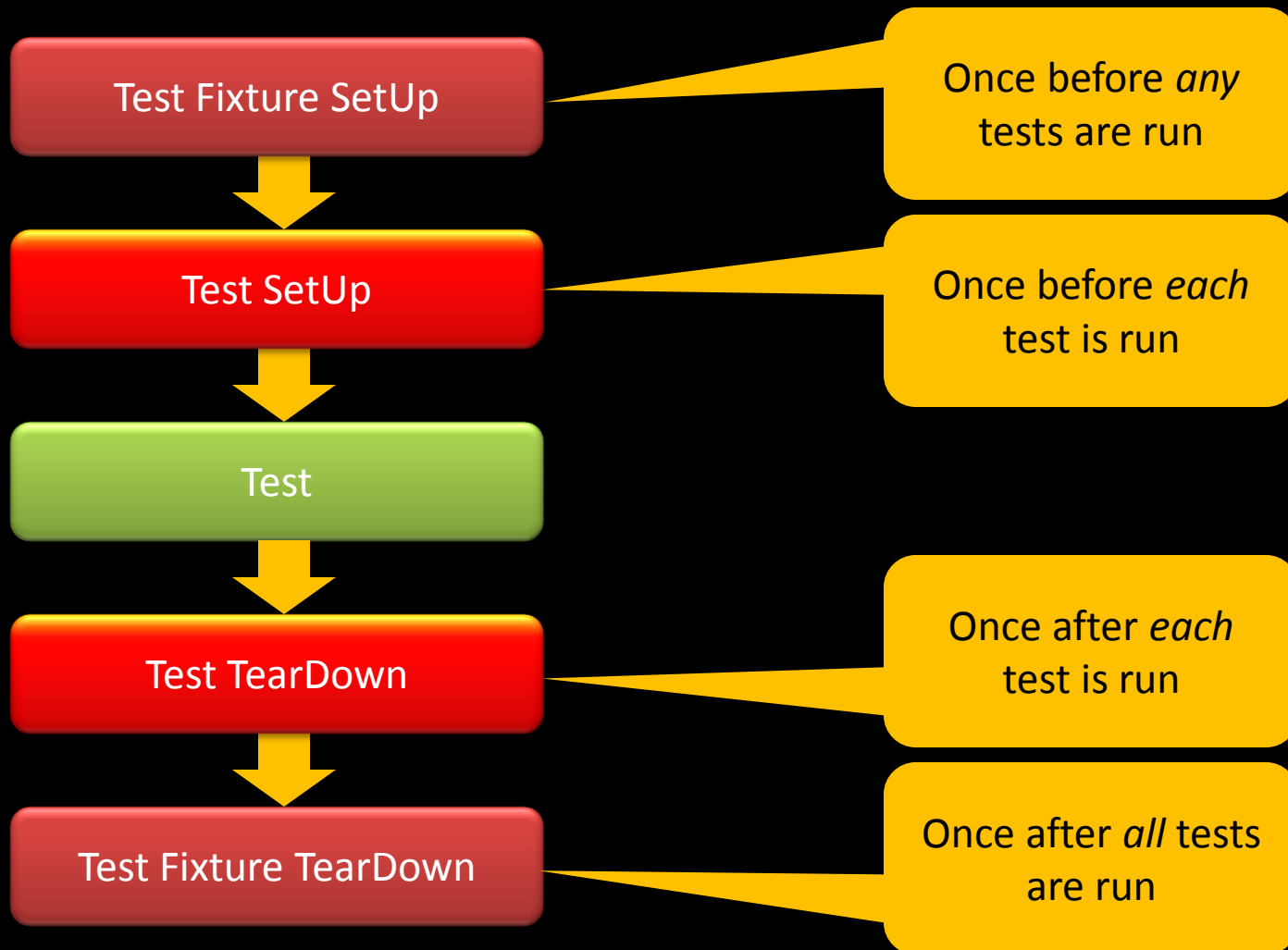
(YAGNI – You Ain't Gonna Need It!)

Waiting until code is written to write the tests often means hard (impossible?) to test code!

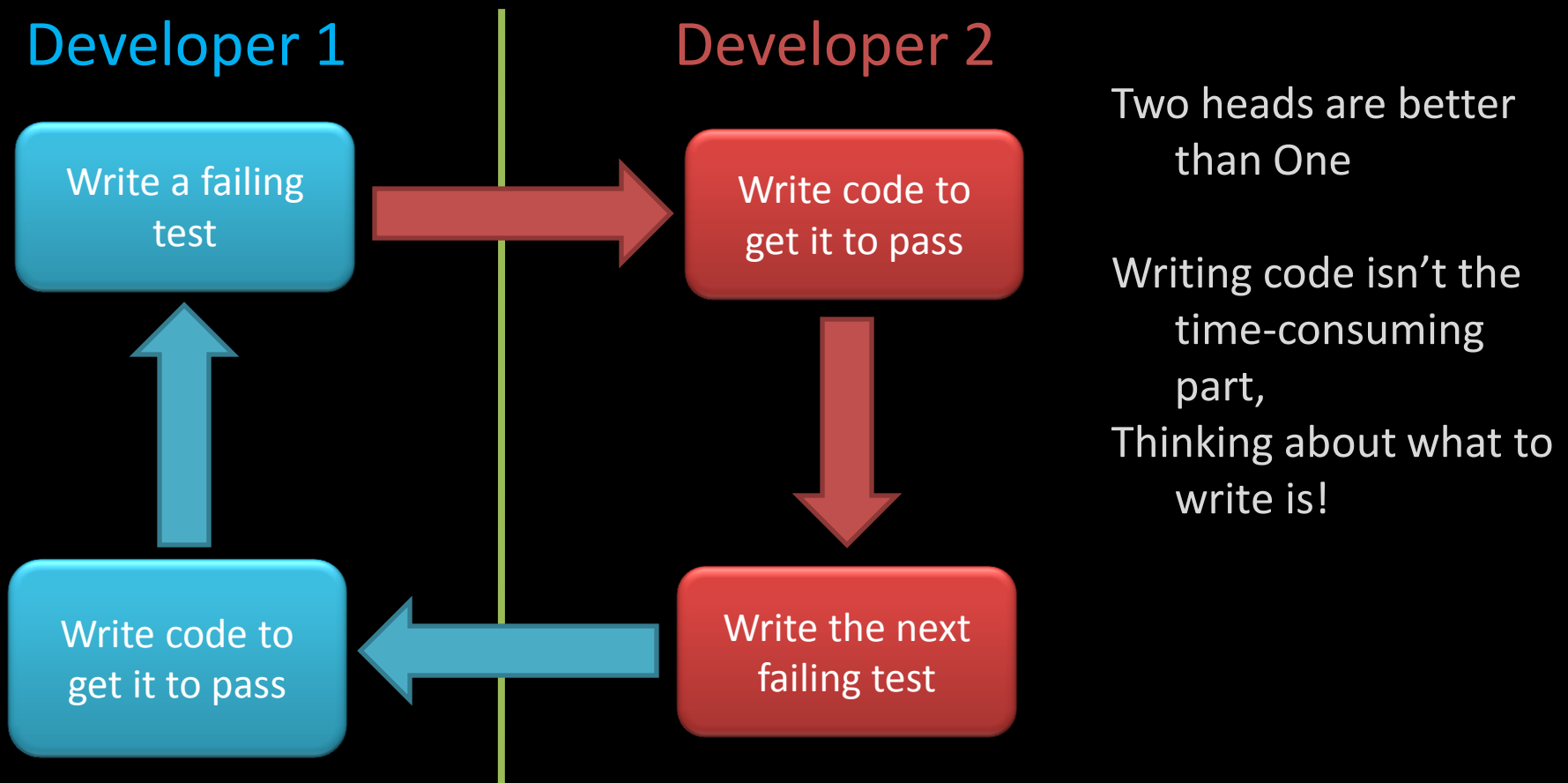
Enough talking, let's write some code!

OUR FIRST TEST

Unit Test Flow



Ping-Pong Pair Programming



Let us show you how that's done

PING-PONG DEMO

The Process

REFACTOR

...mercilessly!

Coding Lab

CALCULATING PRIMES

Problem Statement

Calculate Prime Numbers between 1-10,000

A positive integer divisible only by 1 and itself

By definition, 0 and 1 are *non-prime*

Write results to the console as follows:

Multiple lines of 5 comma-separated values each

Every 10 lines, insert a line indicating the count of the primes output thus far

2,3,5,7,11

n,n,n,n,n

n,n,n,n,n

n,n,n,n,n

n,n,n,n,n

n,n,n,n,n

n,n,n,n,n

n,n,n,n,n

n,n,n,n,n

n,n,n,n,n

Count: 50

n,n,n,n,n

n,n,n,n,n

....

Write a Failing Test

Get it to Pass

Refactor!

Now GO!

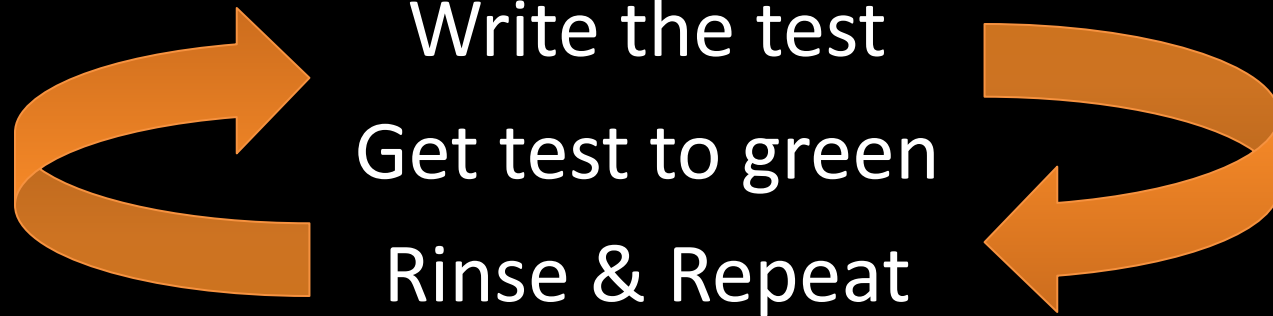
Find someone to pair with

Decide on features to implement

Write the test

Get test to green

Rinse & Repeat



Remember: **Failure** *IS* an option!

Ways to think about Unit Testing

Unit Testing is...

An alternative to spending your life in the debugger

A way to validate that your code is aligned with your intent

‘it compiles’ is a validation of the syntax of your code

‘it passes the unit tests’ is a validation of the behavior of your code

Ways to think about Unit Testing

Unit Testing is...

An hedge against future bugs

Prevents regression bugs (oops, I broke it!)

A safety-net that allows you to experiment with your design

‘What-if’ can be safely explored

No code is ‘hands-off for fear of something breaking’

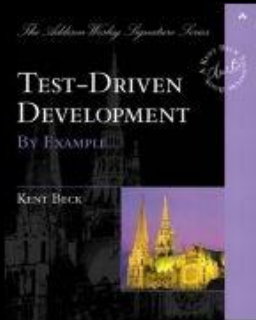
Ways to think about Unit Testing

Unit Testing is not...

Going to save you any time in the project

...until someone changes the project requirements

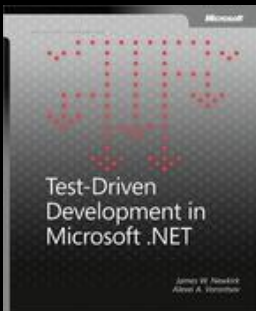




Test-Drive Development: By Example – Kent Beck
<http://amzn.com/0321146530>



The Art of Unit Testing – Roy Osherove
<http://amzn.com/1933988274>



Test-Drive Development in Microsoft .NET – James Newkirk,
Alexei Vorontsov
<http://amzn.com/0735619484>



Agile
Firestarter

Questions?

Twitter: @AlexHung
alex.y.hung@gmail.com