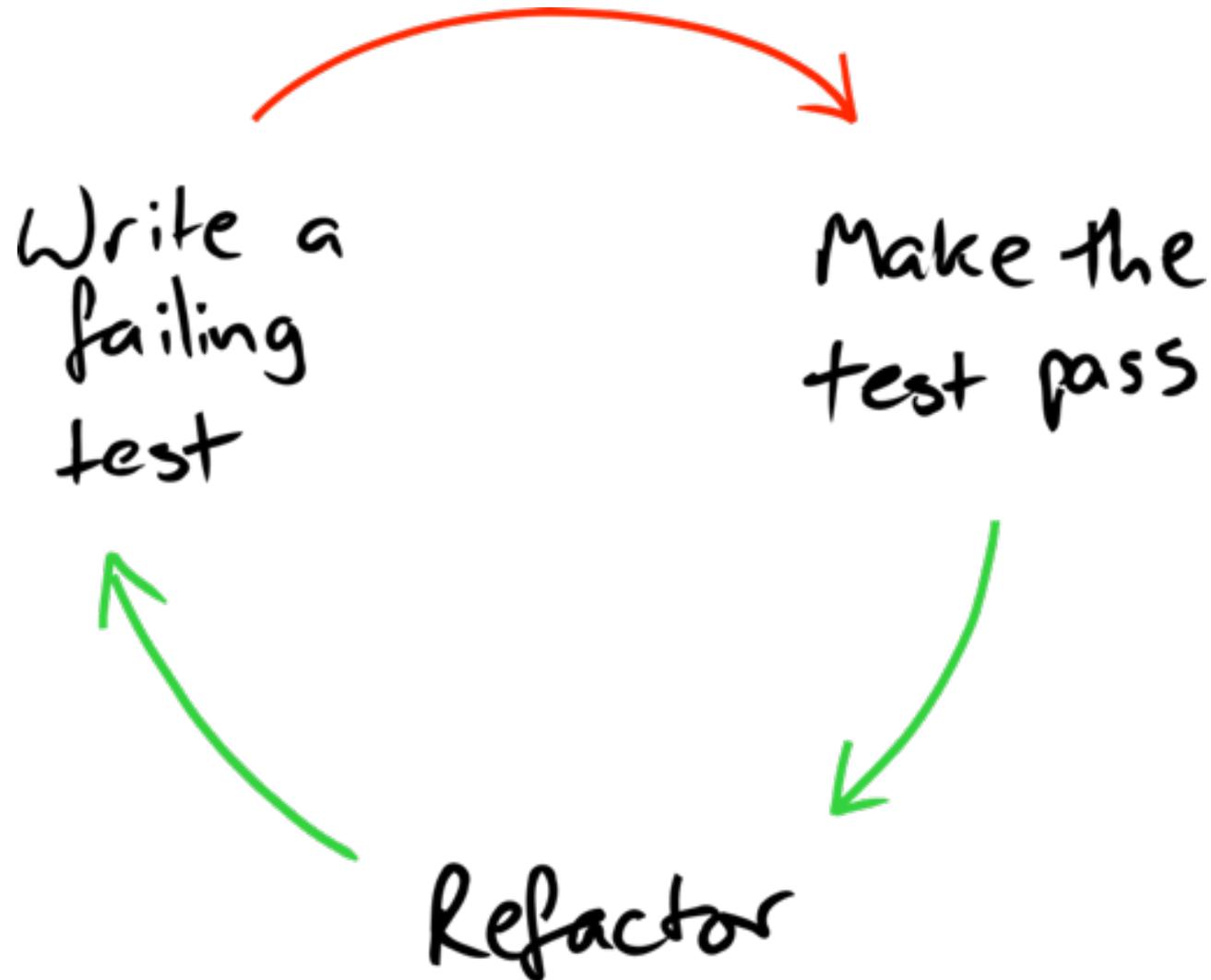


# Ten Years of Test-Driven Development

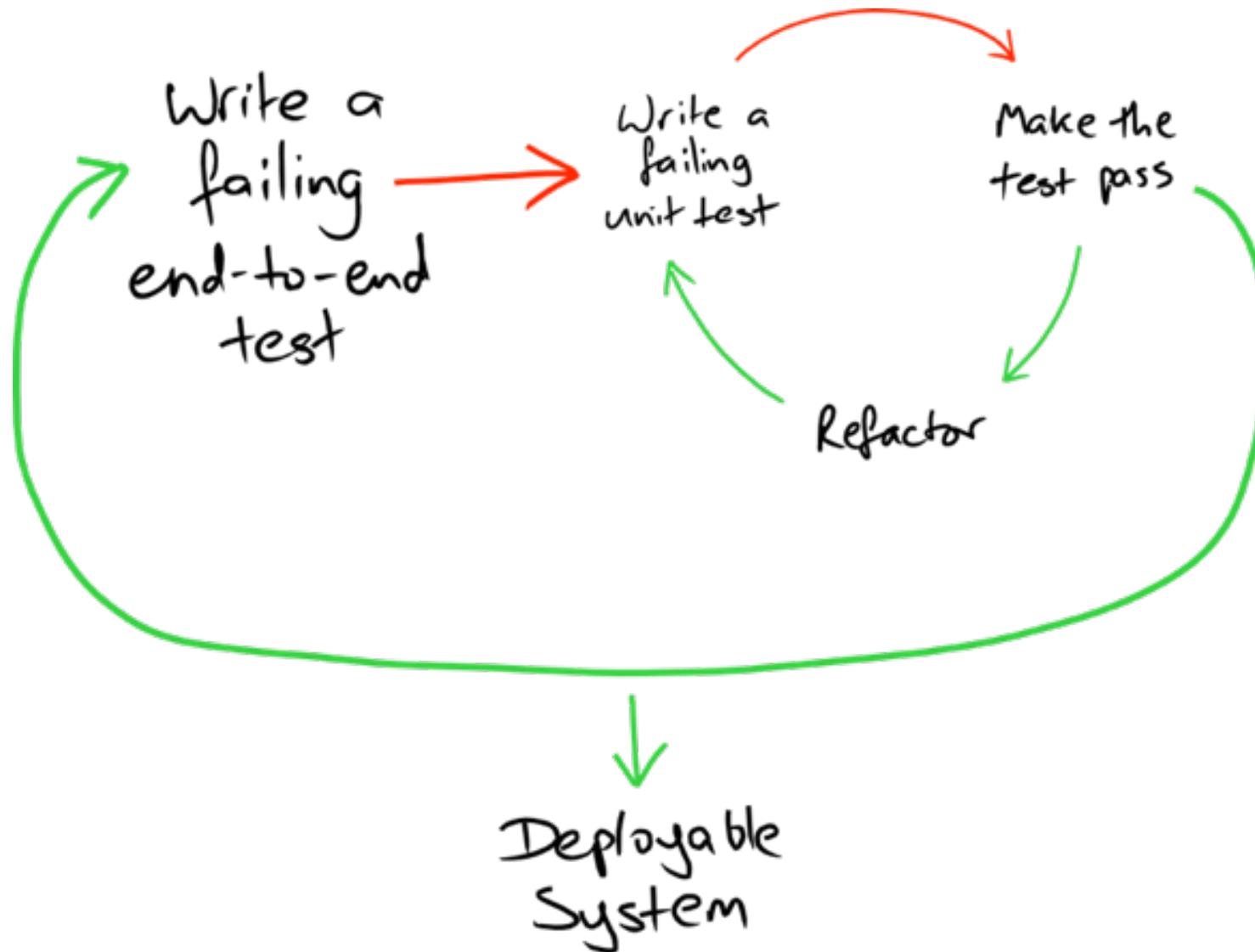
**Michael Feathers**  
**Steve Freeman**



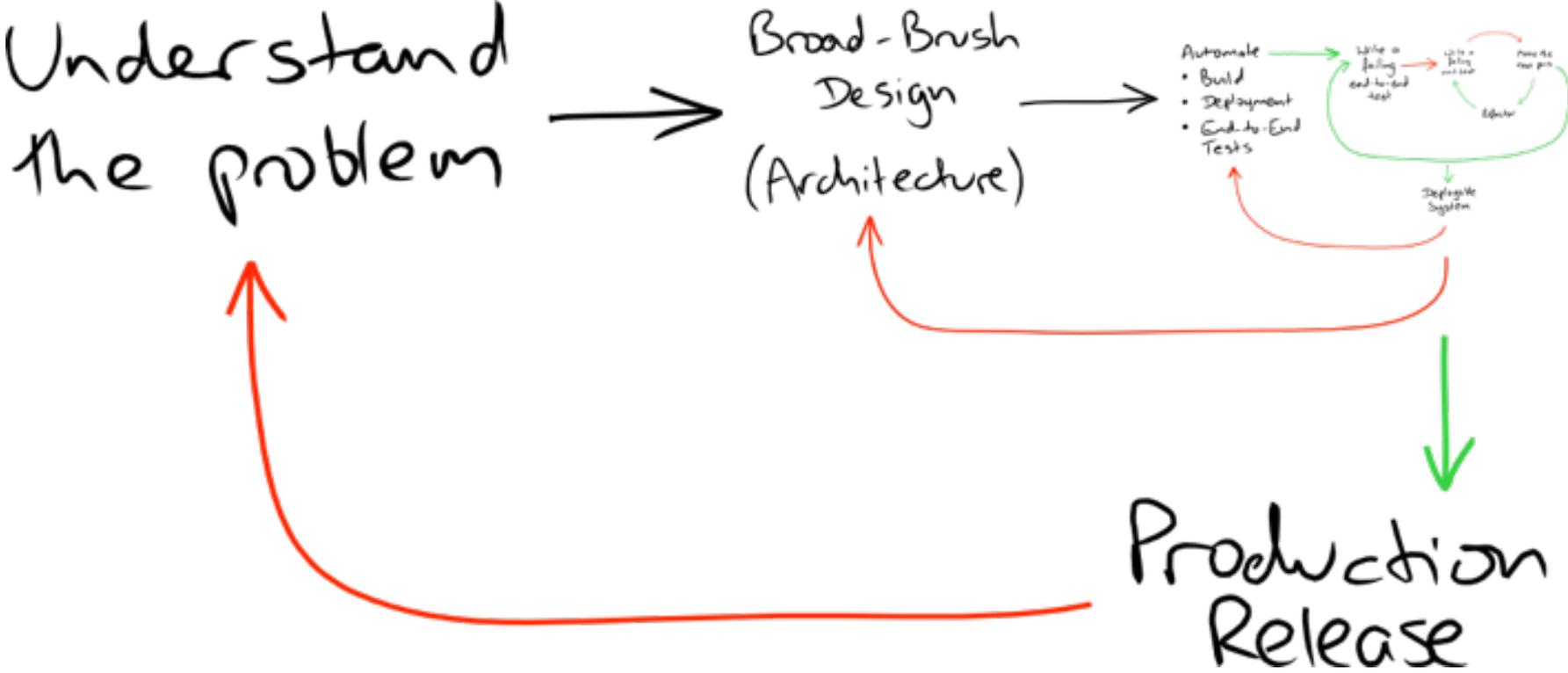
# Review: TDD in one slide



# Review: Acceptance Test-Driven



# Review: The larger feedback loop



1960's

# The secrets of the ancients



It was just assumed that any pro would do a damn good job of this activity, not to be embarrassed by unit bugs found in integration test or system test—or god help us, in production.

Jerry Weinberg

<http://secretsofconsulting.blogspot.com/2008/12/how-we-used-to-do-unit-testing.html>

# Professionals test their code

1970's

# Kent Beck reads his Dad's books



<http://www.flickr.com/photos/mwichary/2376639066/>

**Kent Beck**

15:10

anyway, i had been using sunit for a couple of months when i remembered this passage from a programming book i'd read as a kid. it said the way to program is to look at the input tape and manually type in the output tape you expect. then you program until the actual and expected tapes match.

15:10

i thought, what a stupid idea. i want tests that pass, not tests that fail. why would i write a test when i was sure it would fail. well, i'm in the habit of trying stupid things out just to see what happens, so i tried it and it worked great.

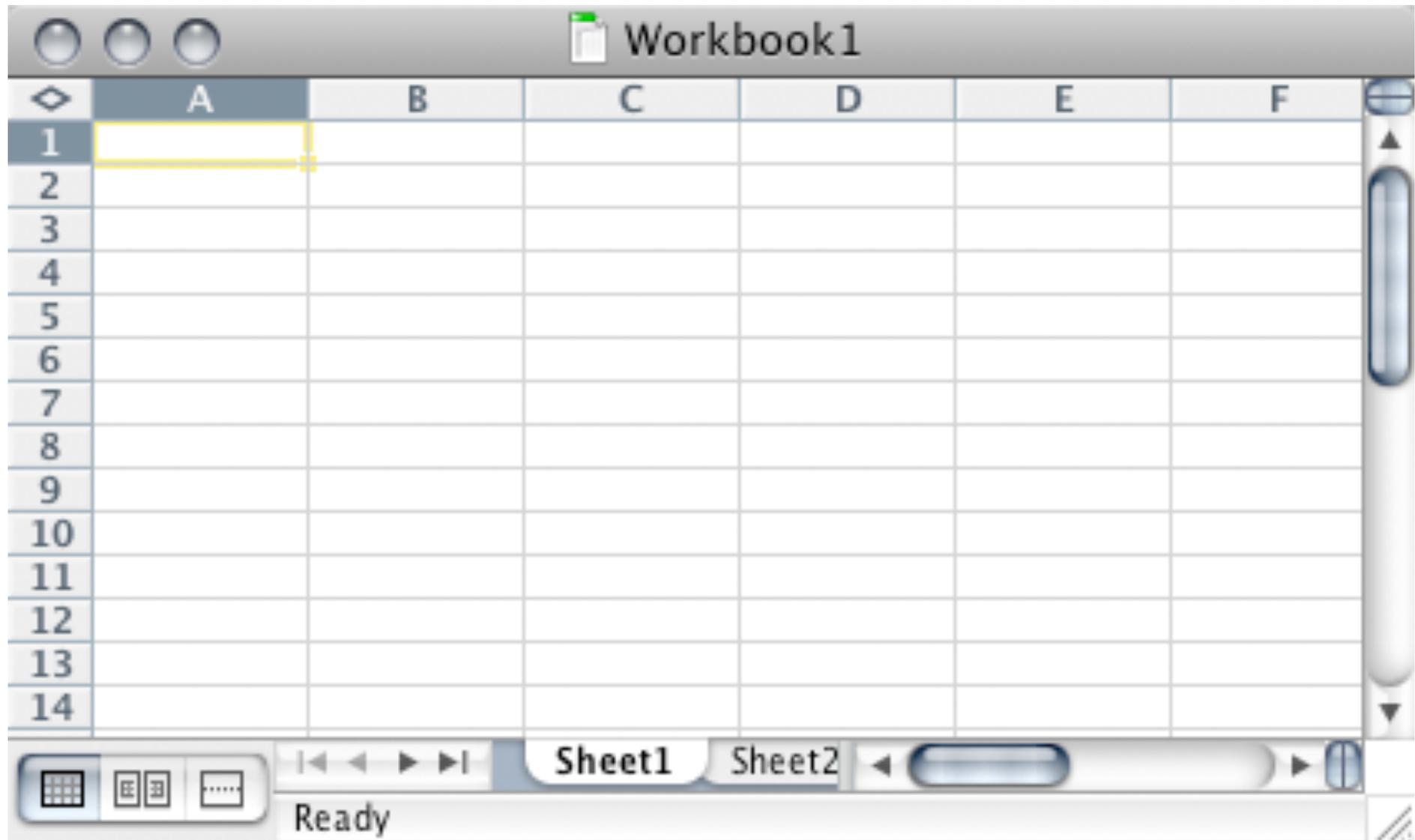
15:11

i was finally able to separate logical from physical design. i'd always been told to do that but no one ever explained how.

Separate what from how

1989

# Ward short-circuits the process



I decided to short-circuit the process by reading his spreadsheets into a test framework I wrote in a day.

[...] we sailed through a big five audit when the auditor admitted that he'd never seen anything like our test browser and passed us before lunch.

# Automated user-tests confirm features

1996

# Scaling up on the C3 project



Ten-Years of Test-Driven Development. Michael Feathers and Steve Freeman

# Ron remembers

We had lots of unit tests on C3. We agree that we didn't START with TDD, though we had been told tests first was good.

My personal [recollection] is that I didn't really get into it until Kent's book started to come out.

It's a change in culture

XUnit

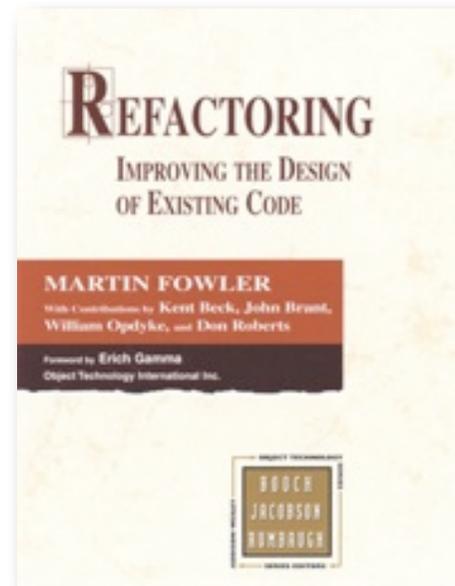
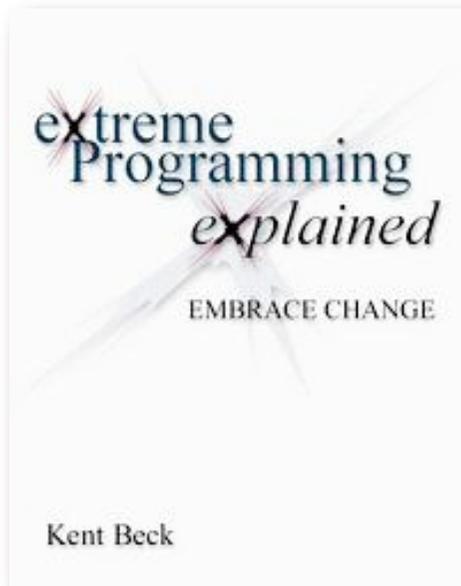
SUnit

JUnit

Nunit

# Spreading the word

...open discussions on C2 wiki



codified in books

# This isn't music

Pos. I. II

Pos. III. IV

III

III. IV

III. IV

ff

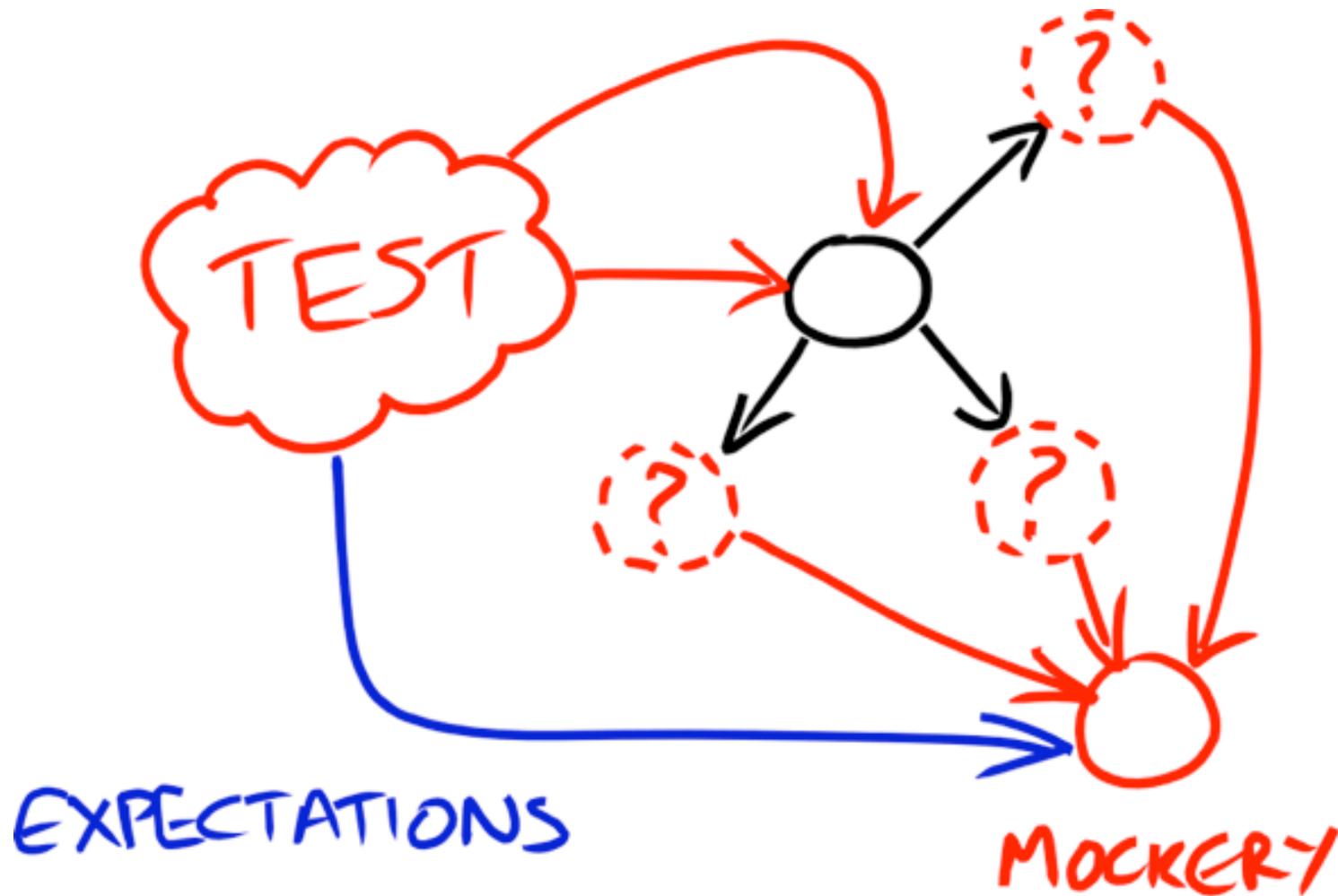
ff

ff

“Working”  
isn't good enough

1999

# Mock Objects



# The origins of Mock Objects

- “These getters we write for testing are cluttering up the design”
- Difficulties with integration testing
- Endless discussion at the eXtreme Tuesday Club



# What Connextra learned

- Composition over inheritance (again)
- Self-explaining failures
- Roles based on behaviour

```
public class Person {  
    public void printDetails(PrintWriter writer) {  
        writer.println(myName);  
        writer.println(myAge);  
        writer.println(myTelephone);  
    }  
}
```

```
public class Person {  
    public void handleDetails(PersonHandler handler) {  
        handler.name(myName);  
        handler.age(myAge);  
        handler.telephone(myTelephone);  
    }  
}
```

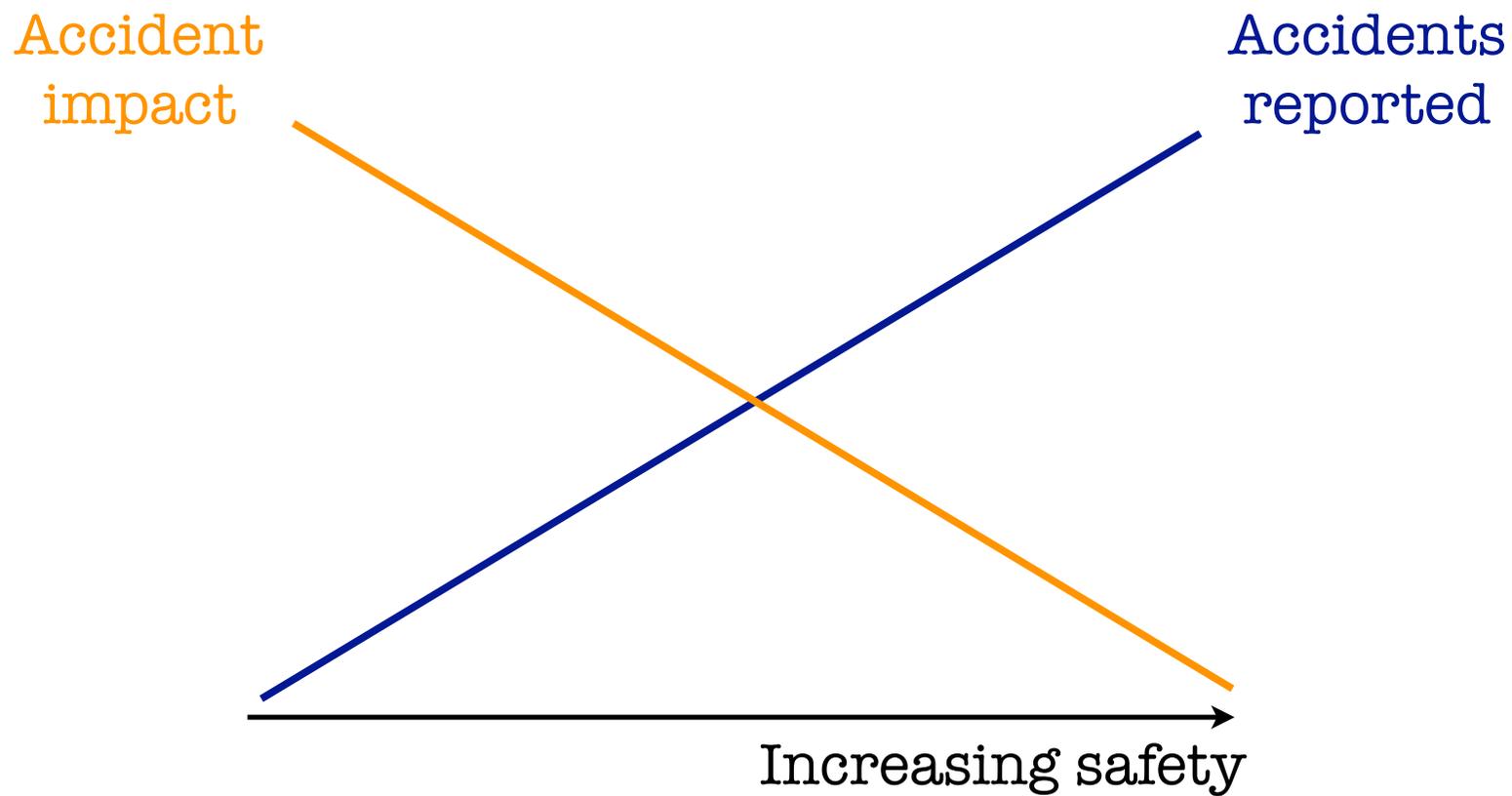
Test style  
drives code style

2001

# Continuous Build



# Safety Culture



(Thanks to Keith Braithwaite for the concept)

The system always works

# Chris Stevenson's TestDox style

**A Sheep** eats grass

**A Sheep** bleats when frightened

**A Sheep** produces delicious milk

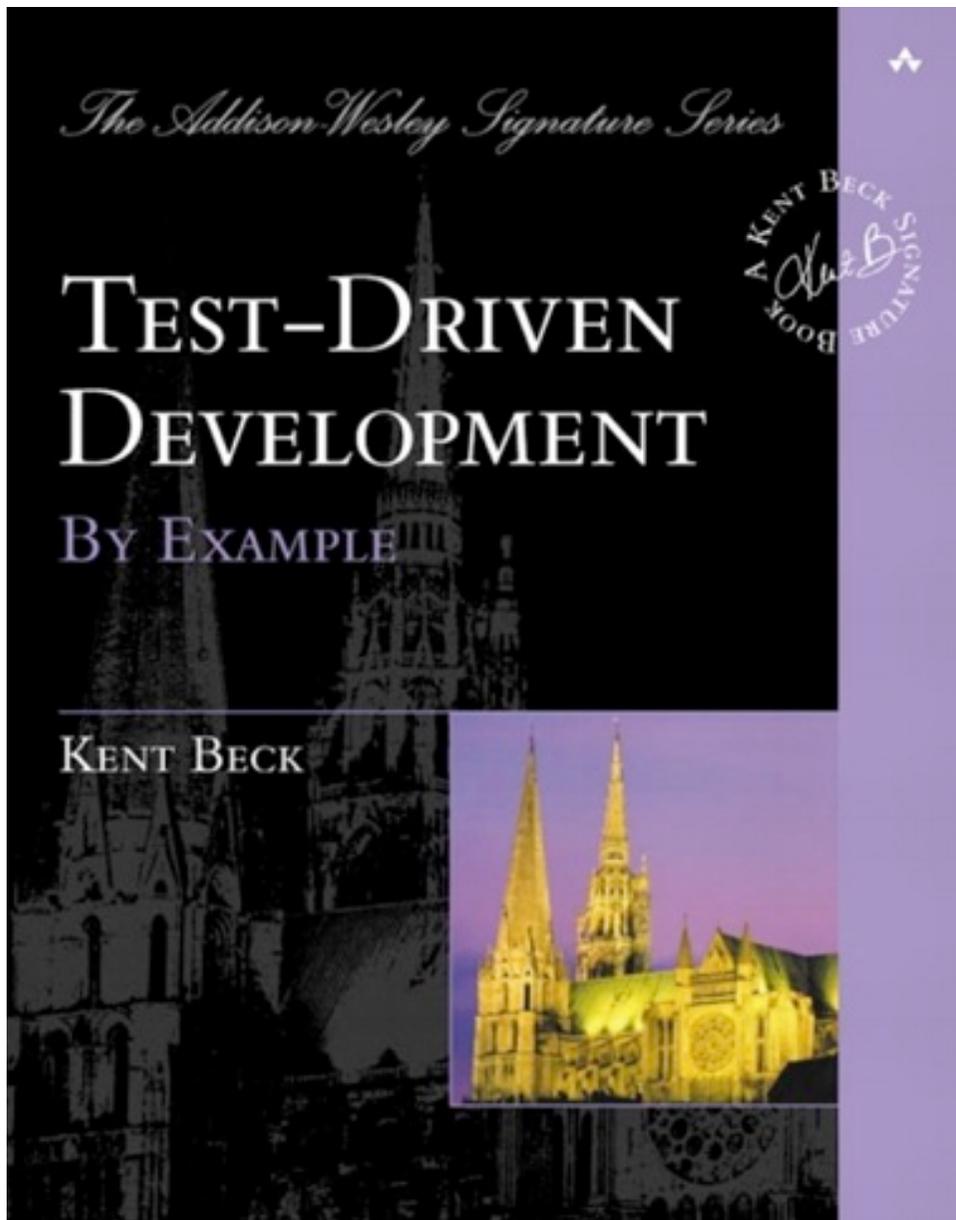
**A Sheep** moves away from sheep dogs

```
public class SheepTest {  
    @Test public void eatsGrass() { ... }  
    @Test public void bleatsWhenFrightened() { ... }  
    @Test public void producesDeliciousMilk() { ... }  
    @Test public void movesAwayFromSheepdogs() { ... }  
}
```

# Focus on intent

2092

# Kent Beck scales down

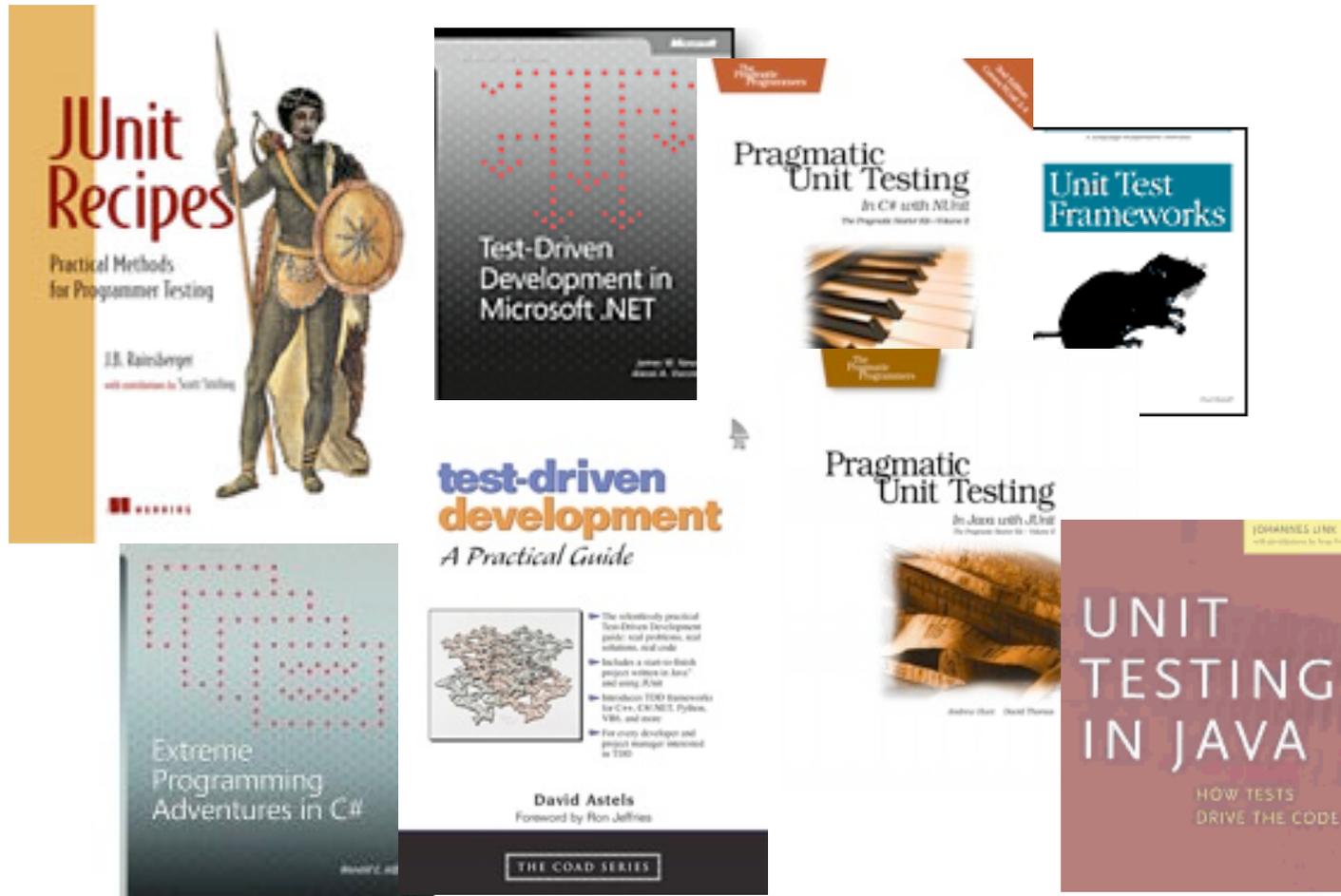


- Write a failing automated test before you write any code
- Remove duplication

When you're lost,  
slow down

2093

# The next generation



Ten-Years of Test-Driven Development. Michael Feathers and Steve Freeman

2004

# TDD for the Customer



result.htm - Microsoft Word

File Edit View Insert Format Tools Table Window Documents To Go Help X

Basic Employee Compensation

For each week, hourly employees are paid a standard wage per hour for the first 40 hours worked, 1.5 times their wage for each hour after the first 40 hours, and 2 times their wage for each hour worked on Sundays and holidays.

Here are some typical examples of this:

Compensation	HolidayHours	Wage	Pay()
	0	20	\$800
	0	20	\$950
	8	20	\$1360
			<i>expected</i>
			\$1040 <i>actual</i>

At 1" Ln 1 Col 1 REC TRK EXT OVR E

ZBrew: Disconnect

Project Edit Search Suite Storytest Table Row Column Font Format Help

Back Forward Undo Redo Test Clear Reports

Report Storytest Tags

right: 10 wrong: 0 ignored: 0 error: 0

Within

ChatStart

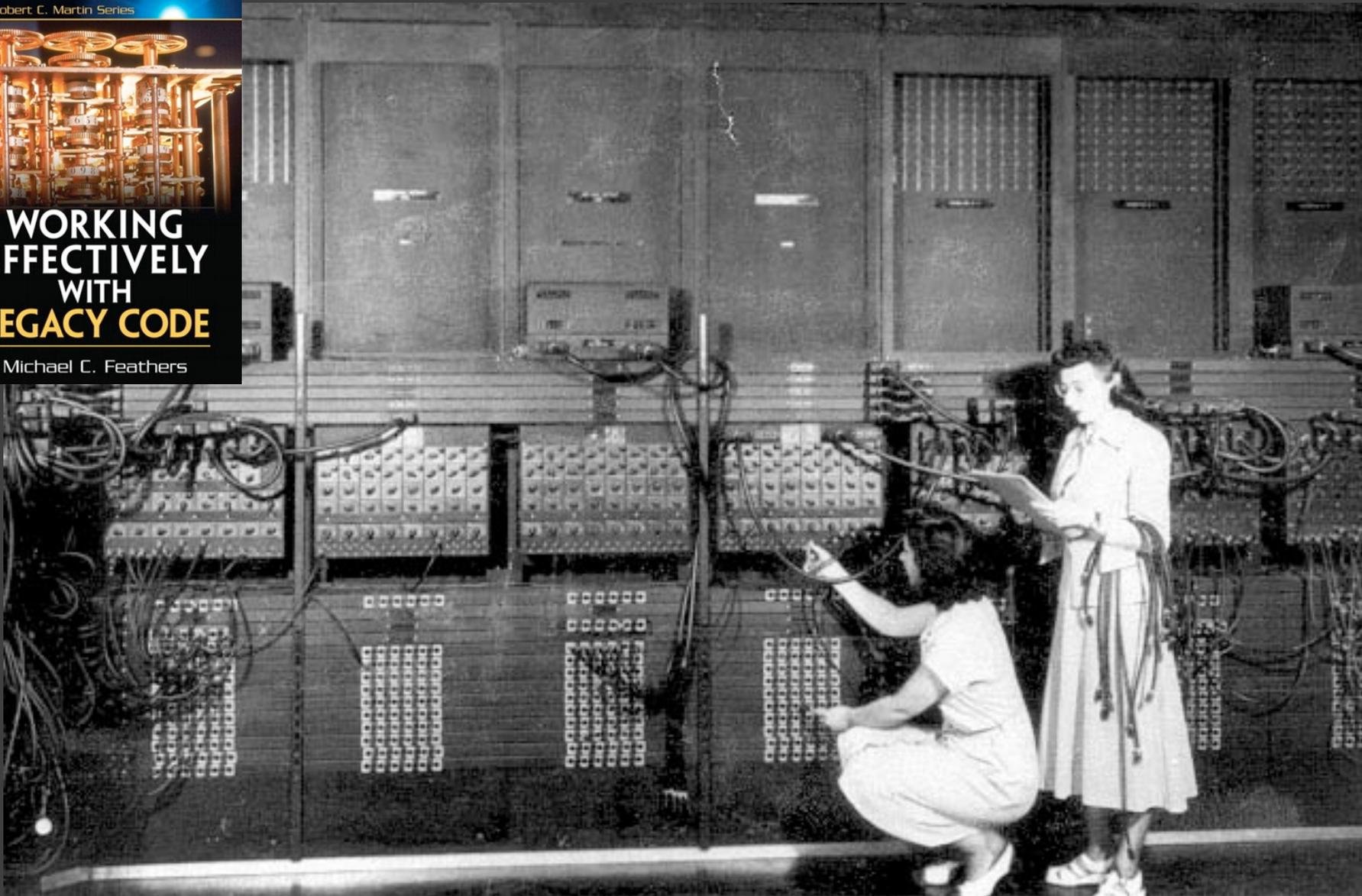
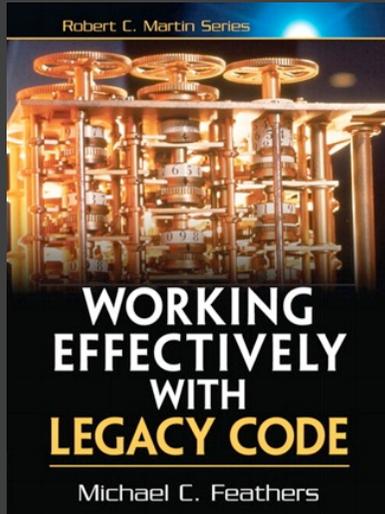
On disconnecting, a user is no longer an occupant of a room.

```
connect user anna
user anna creates job room
user anna enters job room
users in room job
disconnect user anna
check occupant count job 0
```



# Confirm understanding with examples

# Living in the Real World



Ten-Years of Test-Driven Development. Michael Feathers and Steve Freeman

Legacy code is code  
without tests

# Rails. TDD inside



# RSpec “No testing here...”

```
module Mastermind
  describe Game do
    context "starting up" do
      it "should send a welcome message" do
        messenger.should_receive(:puts).with(
          "Welcome to Mastermind!")
        game.start
      end
    end
  end
end
```

It's not only about testing

# and another generation...



Ten-Years of Test-Driven Development. Michael Feathers and Steve Freeman

2007

# The research suggests...



- More effort, better quality
- Effects more visible on real projects
- Very hard to get meaningful results
- Some dissent about the results

# Poor-man's Formal Methods

ScanPT

Δ Field

```
calibrated' pt_factor = pt_formula(calibrated press,calibrated temp)
computed' pt_factor = pt_formula(computed press,computed temp)
```

```
(let pt_corr == if computed pt_mode = automatic
  then calibrated' pt_factor else computed' pt_factor •
  computed' calvolt1 = pt_corr * calibrated calvolt1 ∧
  computed' calvolt2 = pt_corr * calibrated calvolt2)
```

```
{ pt_factor } ≡ calibrated' = { pt_factor } ≡ calibrated
{ pt_factor, calvolt1, calvolt2 } ≡ computed' = { pt_factor, calvolt1, calvolt2 } ≡ computed
```

```
prescribed' = prescribed
accumulated' = accumulated
overridden' = overridden
```

# Measuring TDD

Project	Slope
Jasml 0.1	0.95
Sunflow 0.06.3	1.59
m-e-c scheidung α3-10	1.69
NanoXML 2.2.1	1.77
Syncbuilder1999	1.84
Itext 1.4.8	1.88
Xcool 0.1	1.93
Ant 1.7.0	2.25
Jfreechart 1.0.3	2.30
MarsProject 2.79	2.33
Log4j 1.2.14	2.43
JUnit 3.8.1	2.49
JMock 1.1.0	2.79
Spring 2.0.1	2.78

<http://www.keithbraithwaite.demon.co.uk/professional/software/index.html>

Ten-Years of Test-Driven Development. Michael Feathers and Steve Freeman

# Understand the principles behind the practices

Professionals test their  
code

Separate what from  
how

Automatic user-tests  
confirm features

It's a change in culture

“Working” isn't good  
enough

Test style drives code  
style

The system always  
works

Focus on intent

When you're lost, slow  
down

It's not only about  
testing

Legacy code is code  
without tests

Understand the  
principles behind the  
practices

# Thanks to...

Kent Beck

Keith Braithwaite

Chet Hendrickson

Ron Jeffries

J.B. Rainsberger

Tim Mackinnon

Dan North

Nat Pryce

Chris Stevenson

[youDevise.com](http://youDevise.com)

# Steve Freeman

[steve.freeman@m3p.co.uk](mailto:steve.freeman@m3p.co.uk)



# Ten Years of Test-Driven Development

**Michael Feathers**  
**Steve Freeman**

