

EXECUTABLE REQUIREMENTS

Testing in the language of the business

Tim Wingfield

@timwingfield



Hockey, hockey, hockey,
hockey...hockey all the
time...



Tuesday, January 18, 2011

Central OH
Wendy
Boys
Kona
Hockey
local/regional speaker

How many projects have you seen fail because of the code?

-Chad Fowler, SCNA 2010

Tuesday, January 18, 2011

Dev1: If only we had better requirements.

Dev2: You guys have requirements? Wow.

Crap code ships often – internals of nHibernate?

If you beautifully code the wrong solution, who cares how beautiful it is?



Tuesday, January 18, 2011

Can't express problem to the delivery team.
“Passes down” their ideas
Software is easy

Add a process for importing HR data. This will need to run about once each month.

Functionality for printing receipts

Tuesday, January 18, 2011

Do you want it to actually print receipts? What should be on those receipts?

Please have it load as
fast as it can

Just make it look nice

Tuesday, January 18, 2011

Other end...the features filibuster: Need data import, user registration, high scalability, and a cafe-latte.

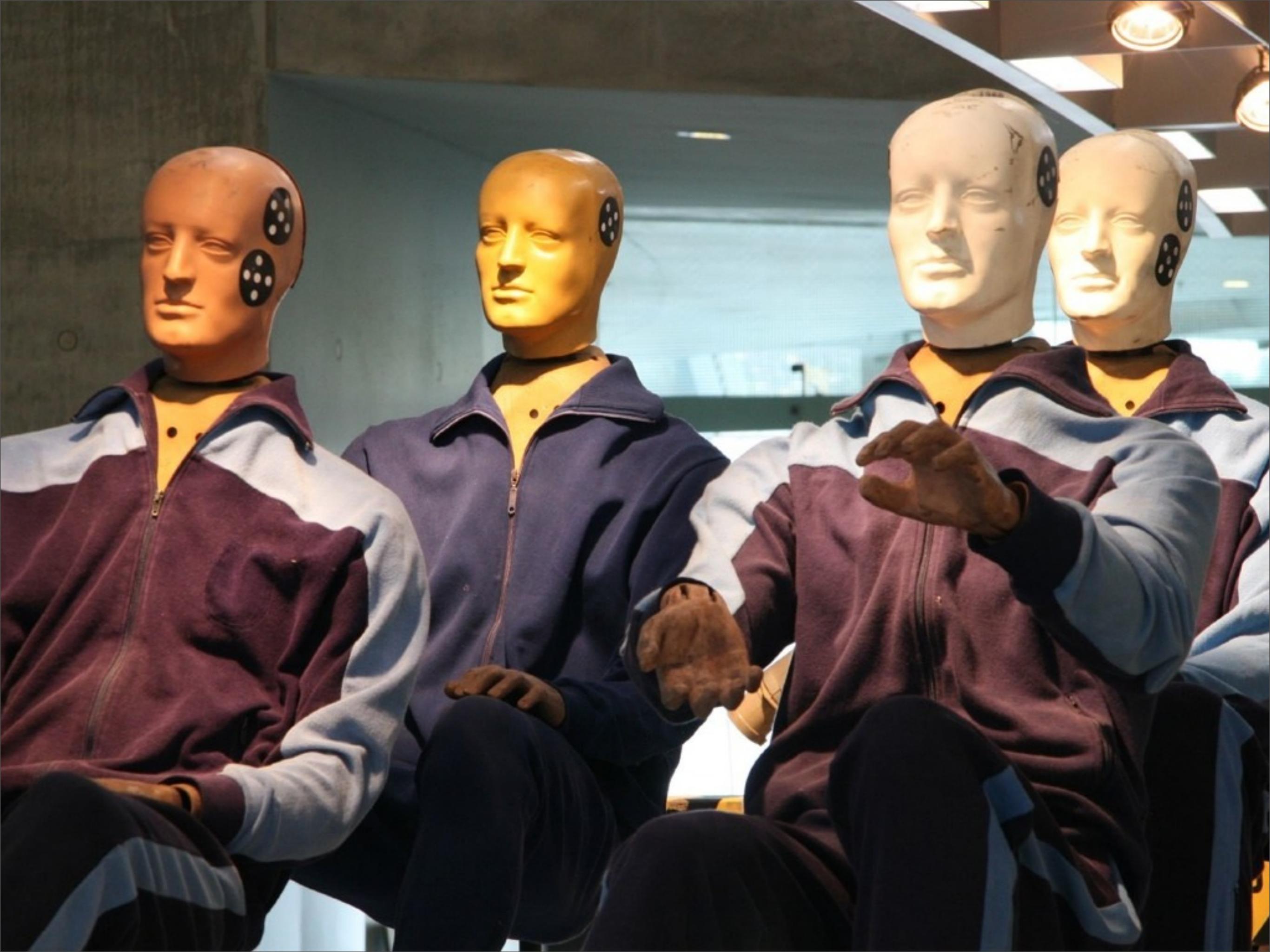


Tuesday, January 18, 2011

Fear-mongoring – “Can’t change SQL trigger, will affect lots of ADO code, may too difficult to update the MVC models, and could keep us from re-implementing the flux capacitor”

Can’t, too difficult, major change

Solving, not listening – “Import Data from...”



Tuesday, January 18, 2011

Room for interpretation

“Fail Gail” – if you’re solely accountable (bottleneck) for quality nothing goes out the door
Seen Kanban boards stack up for that reason
Quality is on the whole team, not just the QA



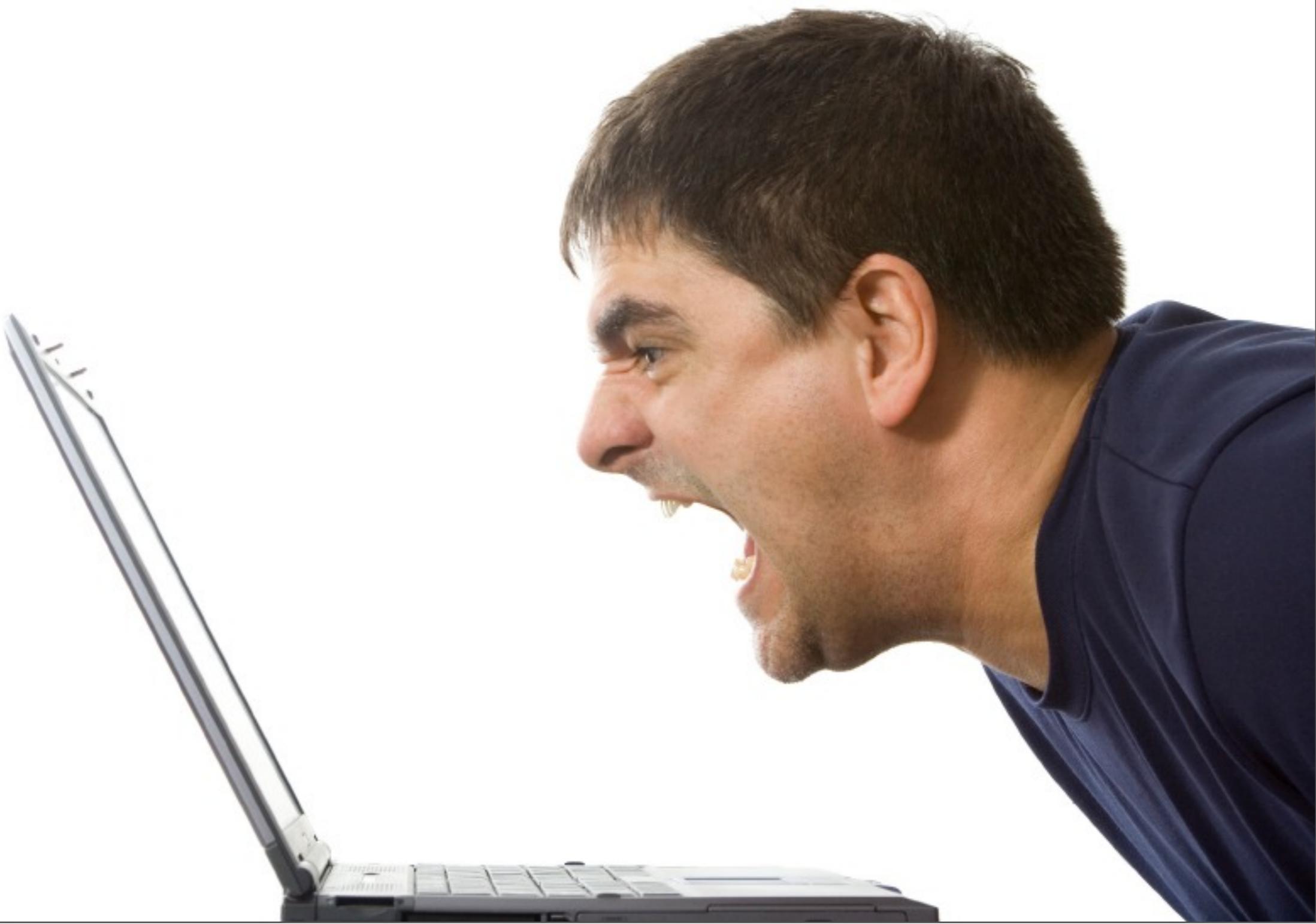
Tuesday, January 18, 2011

Enter Behavior Driven Development
Define and implement behaviors of an application

Imagine a world where you as a developer get detailed requirements in a format and mechanism that is readable by the business facing people...

...completely devoid of
ambiguity in terms of what the
correct behavior is...

...and can also be used as an automated test within your team's continuous integration processes.



Tuesday, January 18, 2011

**app is composed of the things that it does
it's just a collection of behaviors**



Tuesday, January 18, 2011

Limit up front planning. No BDUF.

Doesn't mean NO design up front – basic arch, framework decisions

Very Lean – Last responsible moment

This is Codemash...could've gone with "know when to say when" and add any random picture from the bar.



Tuesday, January 18, 2011

**Very lean
add value or remove cost or why do it?**

Who are our stakeholders?



Tuesday, January 18, 2011

customer is always right...but sometimes room for improvement
devs should be willing to challenge the story requests
NOT fear mongering
will the cost of impl justify the value?



Tuesday, January 18, 2011

Whole team!

Dev, tester, BA, PO

Everyone responsible for quality

Everyone responsible for value

All together we plan out our stories



Tuesday, January 18, 2011

Connextra style
As as...
I want...
So that...

Introduces Context

Given

Tuesday, January 18, 2011

The context or the “As a” portion

When

Tuesday, January 18, 2011

The action, or the “I want”

Then

Tuesday, January 18, 2011

The outcome, or the “So that”

connextra == story
composed of many g/w/t
g/w/t == acceptance criteria
acceptance criteria is agreed by team as story passes
acceptance criteria marker for quality/value



Tuesday, January 18, 2011

Customer based acceptance criteria
coding against data and expectations
inputs and outputs
user adder example => $2+2=4$ is different from “As a mathematician...”
Fit/Fitness are good ATDD tools.



Tuesday, January 18, 2011

Engineering practice
Emergent Design
Tests the internals of your app.
Not a replacement, it's a significant other
TDD == unit/method tests, BDD == functional tests
BDD drives app, TDD drives design



Tuesday, January 18, 2011

Test behaviors. The end points or the hooks to your app.
Think of executing specs as a user
Not for internals – Validating a user example.
Northwoods: Refactor to MVP, test the presenters



Tuesday, January 18, 2011

Java – JBehave

.Net – SpecFlow, MSpec, StoryTeller

javascript – jasmine, jspec

ruby – rspec

plant
sket

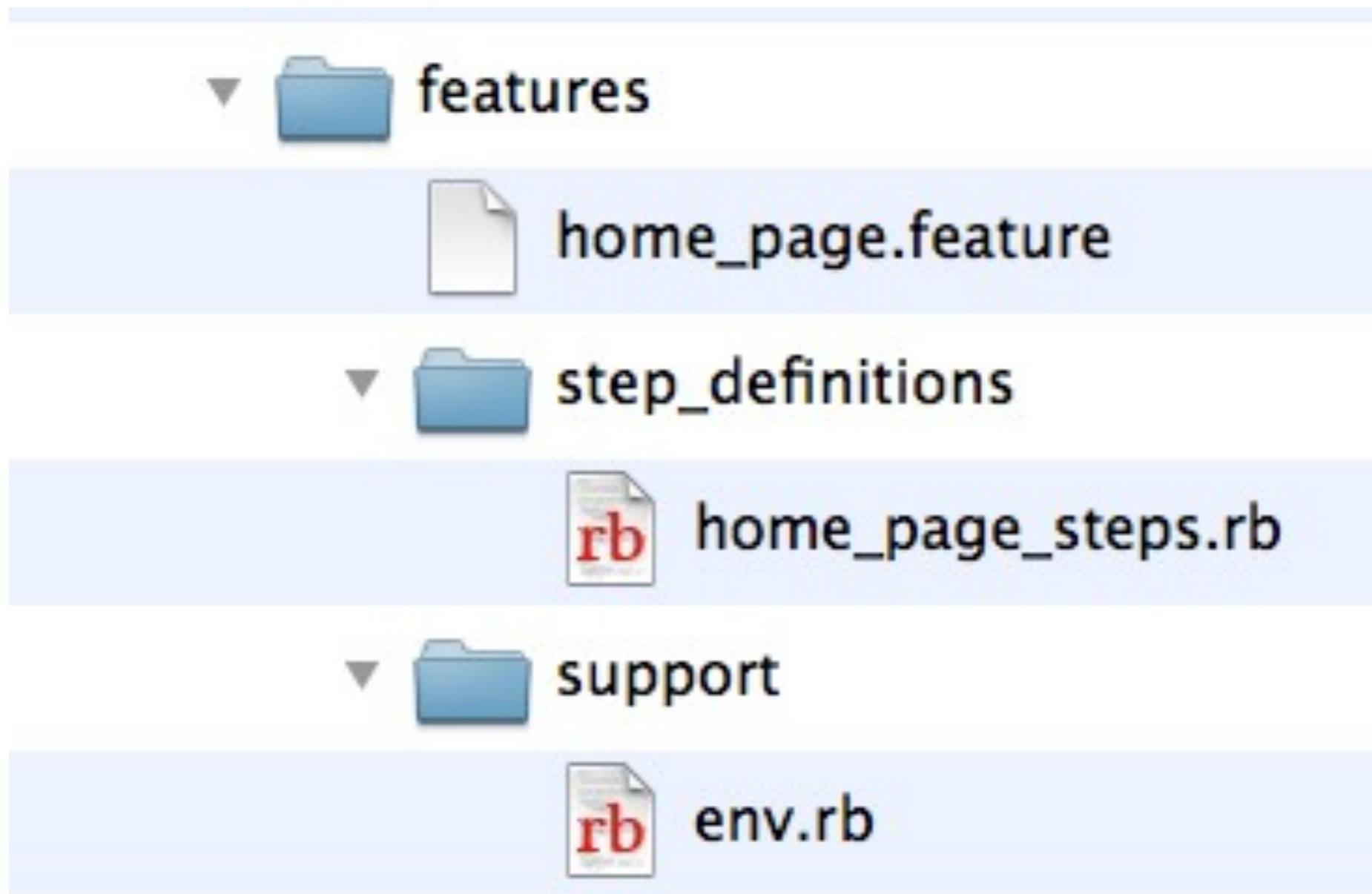
Cucumbers

2. 00 / basket

1

Tuesday, January 18, 2011

Aslak Hellesoy
2008 came from RSpec Story Runner
Plain text specs
Cuke4Duke and Cuke4Nuke



Tuesday, January 18, 2011

Convention over configuration

.feature file

step_definitions home_page.steps.rb

Support – env.rb



Tuesday, January 18, 2011



- Feature



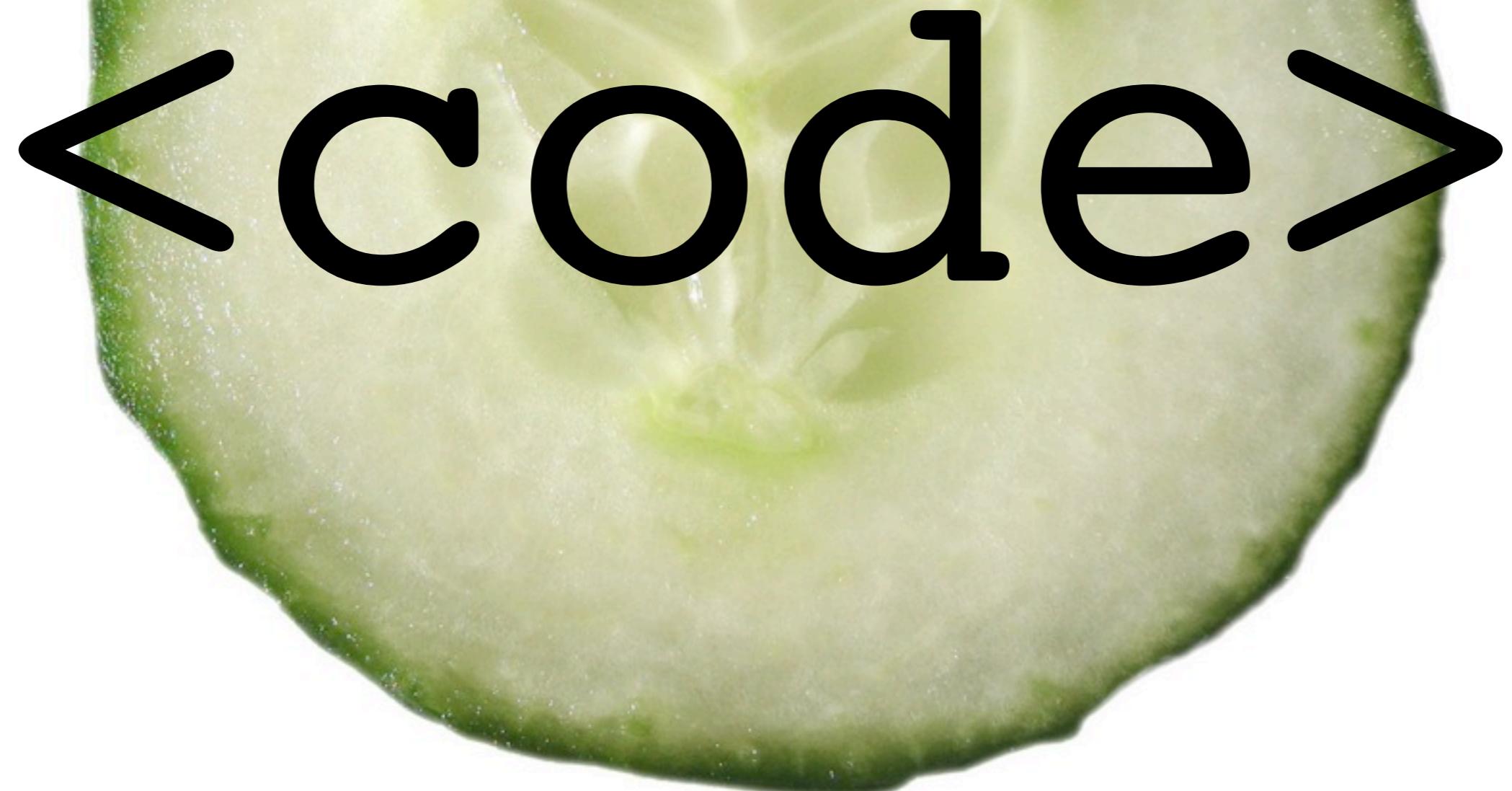
- Feature
- Scenario



- Feature
- Scenario
- Given/When/Then



- Feature
- Scenario
- Given/When/Then
- And/But



<code>

Tuesday, January 18, 2011

Millennium Falcon



Tuesday, January 18, 2011

nothing is
Tell Searls's story of scrapping the executable side



Tuesday, January 18, 2011

**Communication above all else
whole team: TALK
don't hoard knowledge**

photos

guy in suit - <http://www.flickr.com/photos/mytudut/5198146890/>
nerds - <http://www.flickr.com/photos/blyzz/5329090461/>
stop sign - <http://www.flickr.com/photos/thecrazyfilmgirl/3248283617/>
smart car - <http://www.flickr.com/photos/yourdon/3087402587/>
books - <http://www.flickr.com/photos/hadevereux/4626037013/>
yield sign - http://www.flickr.com/photos/question_everything/4202305936/
party - <http://www.flickr.com/photos/st3f4n/4231173553/>

Jeremy Miller's InfoQ article - <http://www.infoq.com/news/2009/09/storyteller>

Tim Wingfield

tim@timwingfield.com

blog.timwingfield.com

[@timwingfield](https://twitter.com/timwingfield)