cucumber ltd

An Introduction to Behaviour Driven Development with Cucumber for Java

Seb Rose

seb@cucumber.io

Agenda

- What is BDD?
- What is Cucumber?
- Demo *Cucumber for Java*
- Cucumber variants
- Putting it all together

What is BDD?

Behaviour Driven Design
Development

BDD defined

BDD is a second-generation, outside-in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology.

It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters.

Dan North

http://skillsmatter.com/podcast/java-jee/how-to-sell-bdd-to-the-business

The power of "*should*"

- Diverts developers from their distaste for testing
- Encourages people to ask "Should it really?"

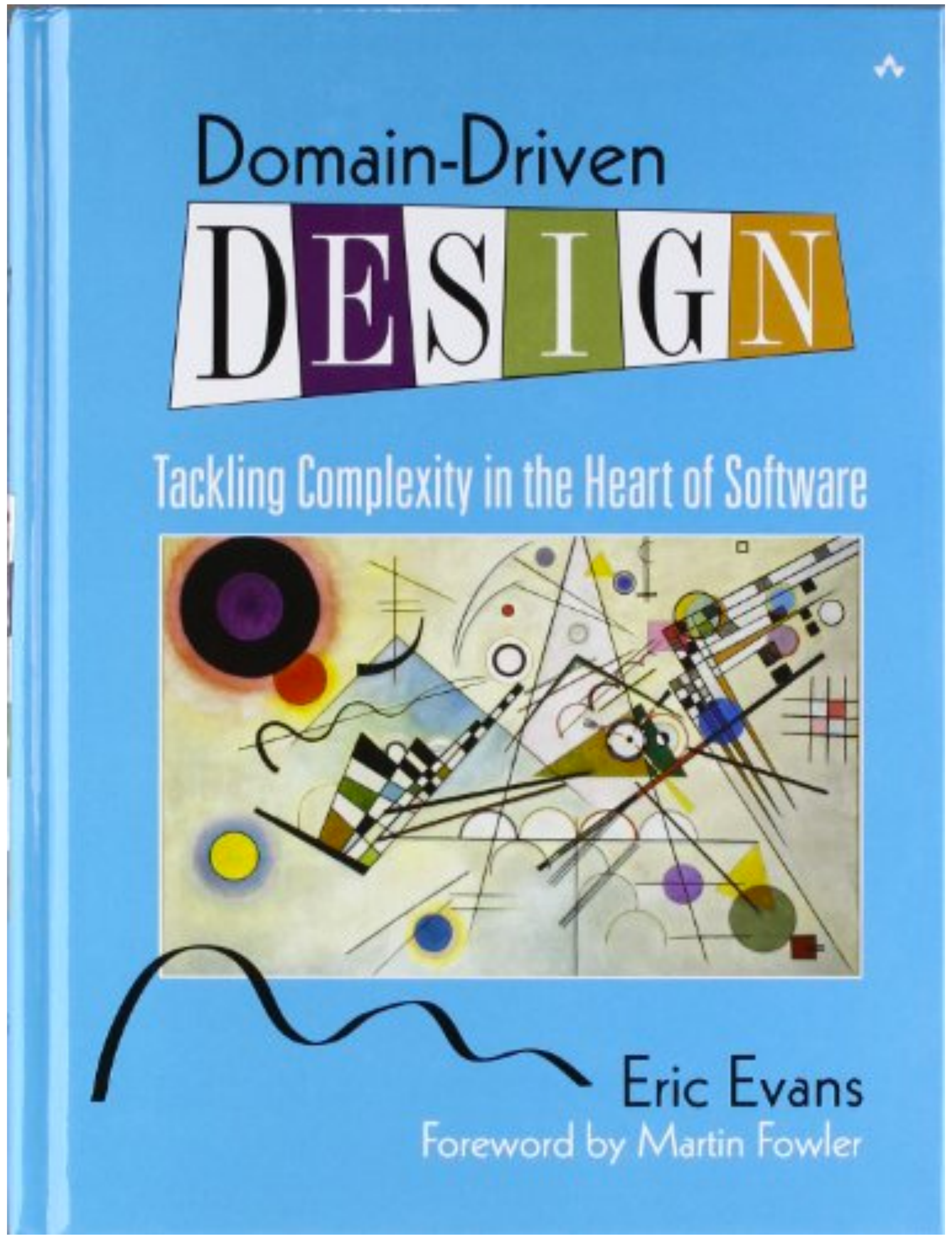
Deliberate discovery

"… during an inception, when we are most ignorant about most aspects of the project, the best use we can possibly make of the time available is to attempt to identify and reduce our ignorance …"

Dan North

http://dannorth.net/2010/08/30/introducing-deliberate-discovery/

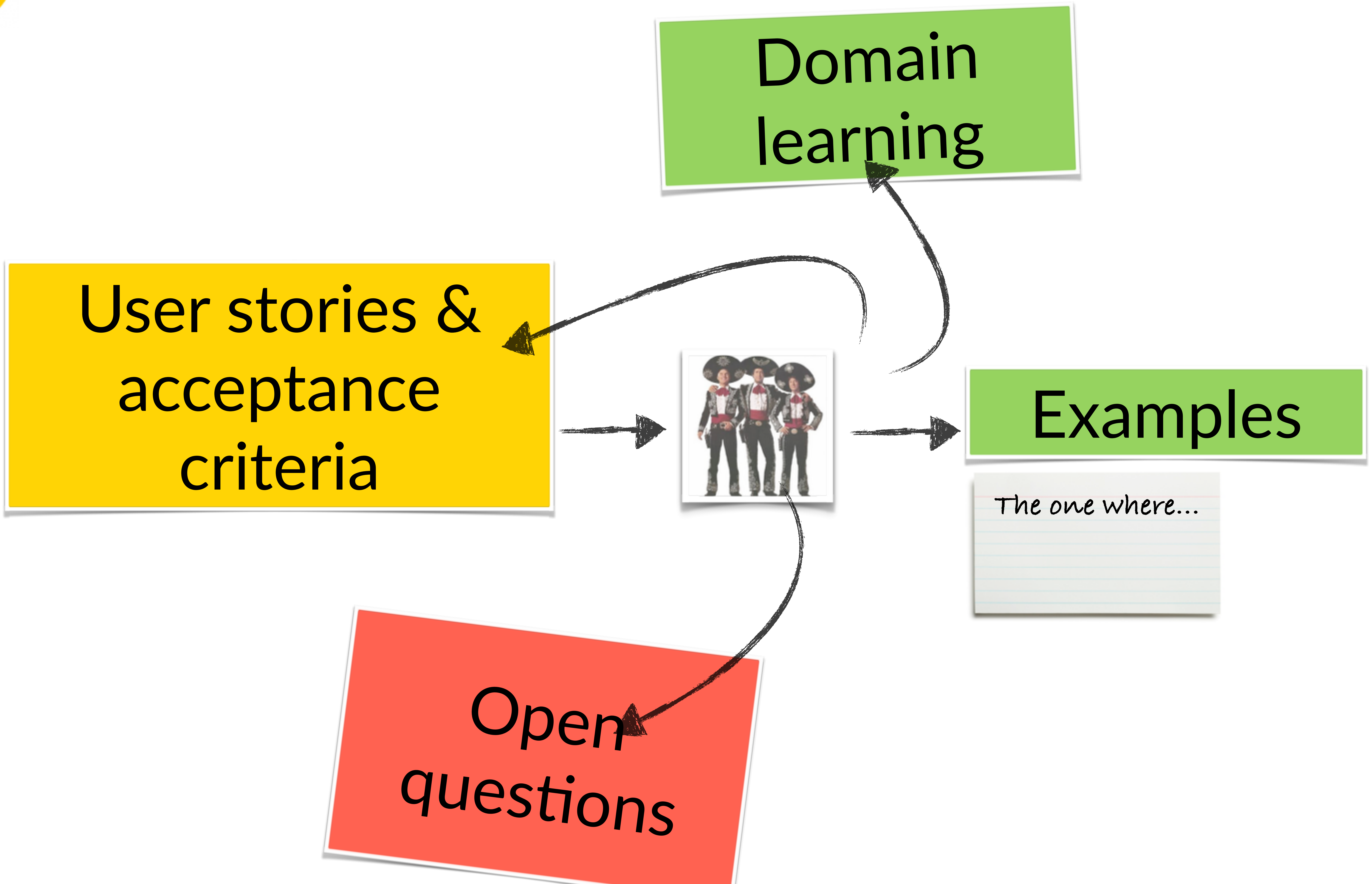
Ubiquitous language


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

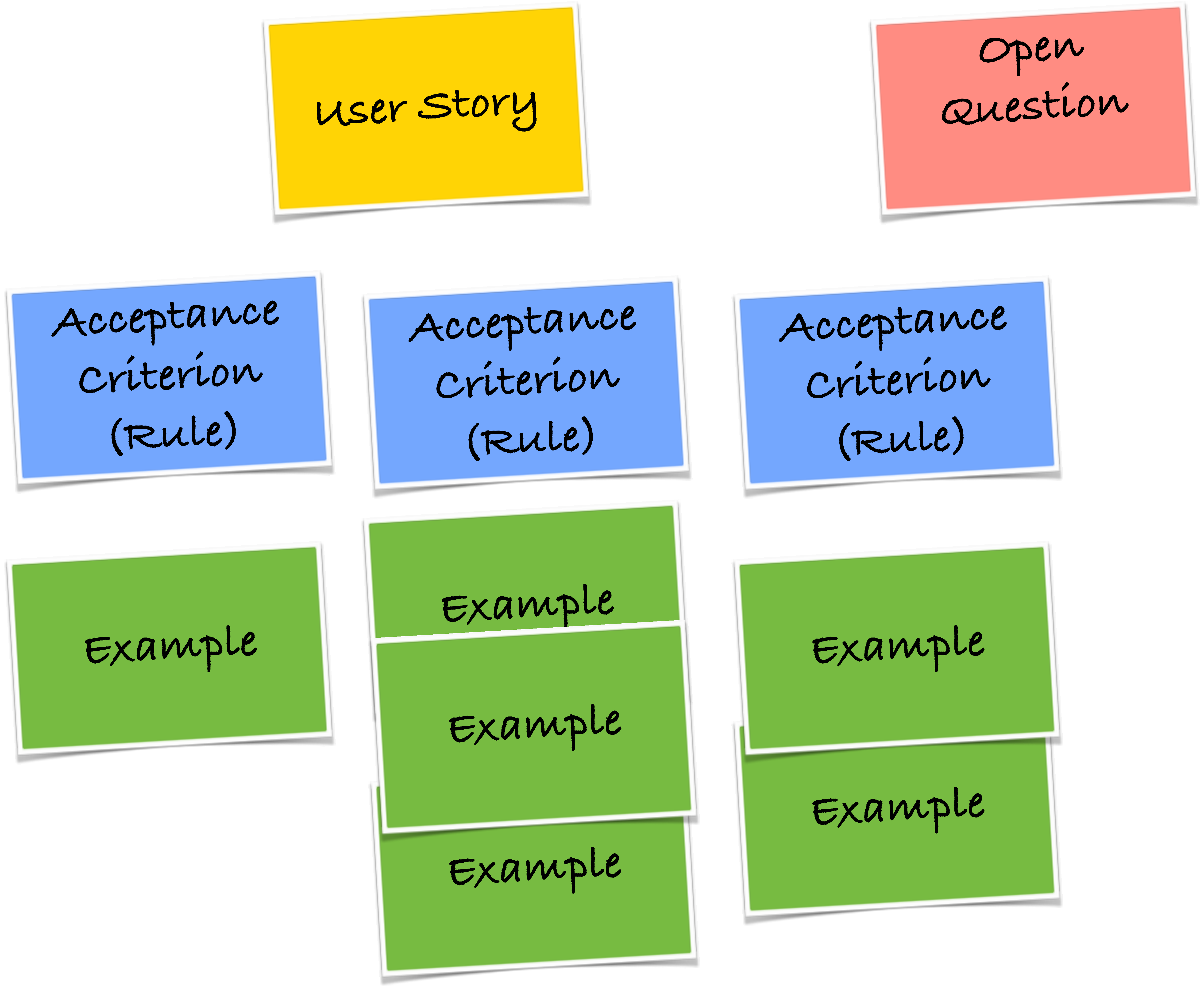
"The practice of building up a common, rigorous language between developers and users"

Martin Fowler

The 3 Amigos


Domain learning
User stories & acceptance criteria
Examples
The one where...
Open questions

Example mapping


User Story
Open Question
Acceptance Criterion (Rule)
Acceptance Criterion (Rule)
Acceptance Criterion (Rule)
Example
Example
Example
Example
Example
Example
Example

Important conversations

having conversations
is more important than
capturing conversations
is more important than
automating conversations

Liz Keogh

http://lizkeogh.com/2014/01/22/using-bdd-with-legacy-systems/

Living documentation

a.k.a Executable specification

Living documentation is a reliable and authoritative source of information on system functionality, which anyone can easily access.

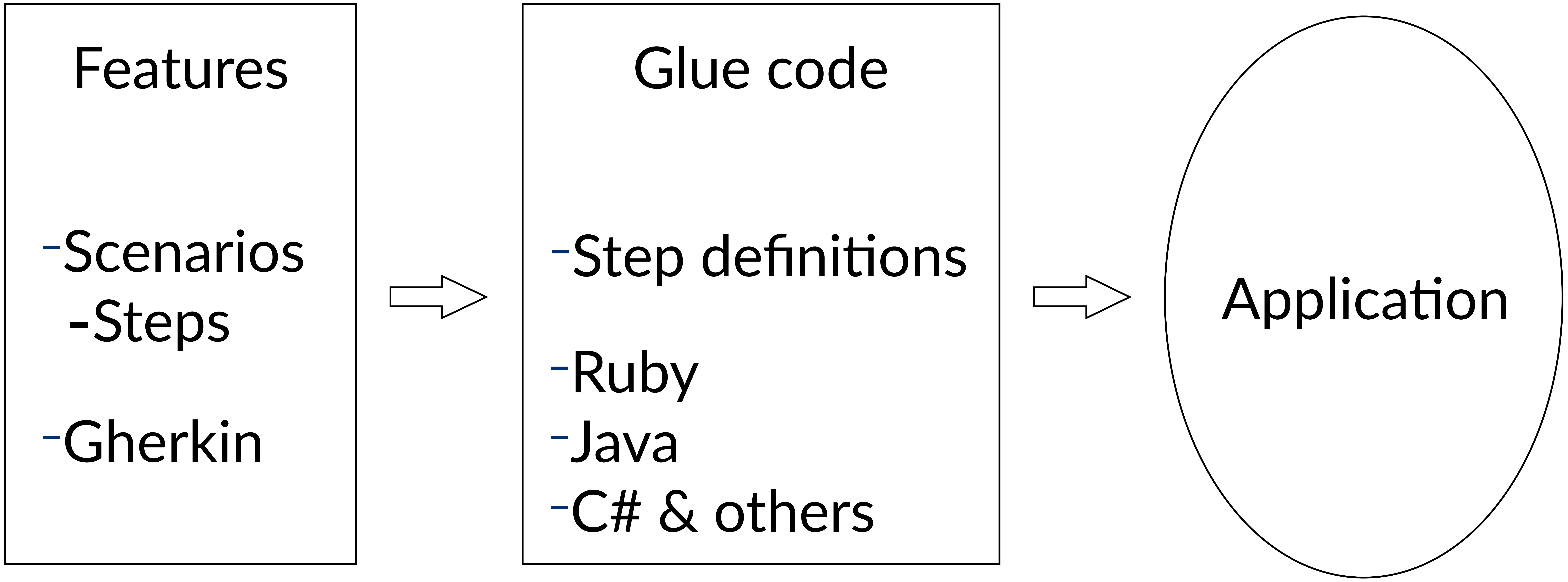
It is as reliable as the code, but much easier to read and understand.

Gojko Adzic

What BDD is not….

- Using “Given/When/Then”
- The responsibility of testers
- An alternative to manual testing

What is Cucumber?



```
Feature: Team Scoring
  Teams start with zero score.
  Correct answer gets points depending on
  how difficult it is.

Scenario: New teams should not have scored yet
  Given I register a team
  Then my score is 0

Scenario: Correctly answering a question scores points
  Given I register a team
  When I submit a correct answer
  Then my score is 10
```

Gherkin fundamentals

```
Feature: Feature name
  Description of feature goes here

  Scenario: Scenario name
    Description of scenario goes here

    Given a certain context
    When something happens
    Then an outcome
    And something else
    But not this though

  Scenario: Another scenario name
    …
```

Ruby

```
Given(/^I register a team$/) do
  # Glue code goes here
end
```

Java

```
import cucumber.api.java.en.*;

public class MyStepDefinitions {

  @Given("^I register a team$")
  public void iRegisterATeam() throws Throwable {
        // Glue code goes here
  }
}
```

DEMO

Cucumber variants

- Ruby
- C# (Specflow)
- Java (& other JVM languages)
- Javascript
- PHP
- Python
- C++

Given/When/Then namespaces

Global namespace
- Given/When/Then interchangeable

Separate namespaces
- Given/When/Then distinct
- And/But bind to preceding namespace
- [StepDefinition] for compatibility

Binding & test frameworks

Behind the scenes
- may need to specify paths
- select required plugin(s)

Some magic code generation
- NUnit by default
- configuration changes for others
- several output options

Sharing data between steps

Varies by implementation
- Ruby, Javascript: World object
- Java: Dependency Injection

Context object(s)
- Injected
- Scenario
- Feature

Hooks

Before & After
Ruby: Around, AfterStep

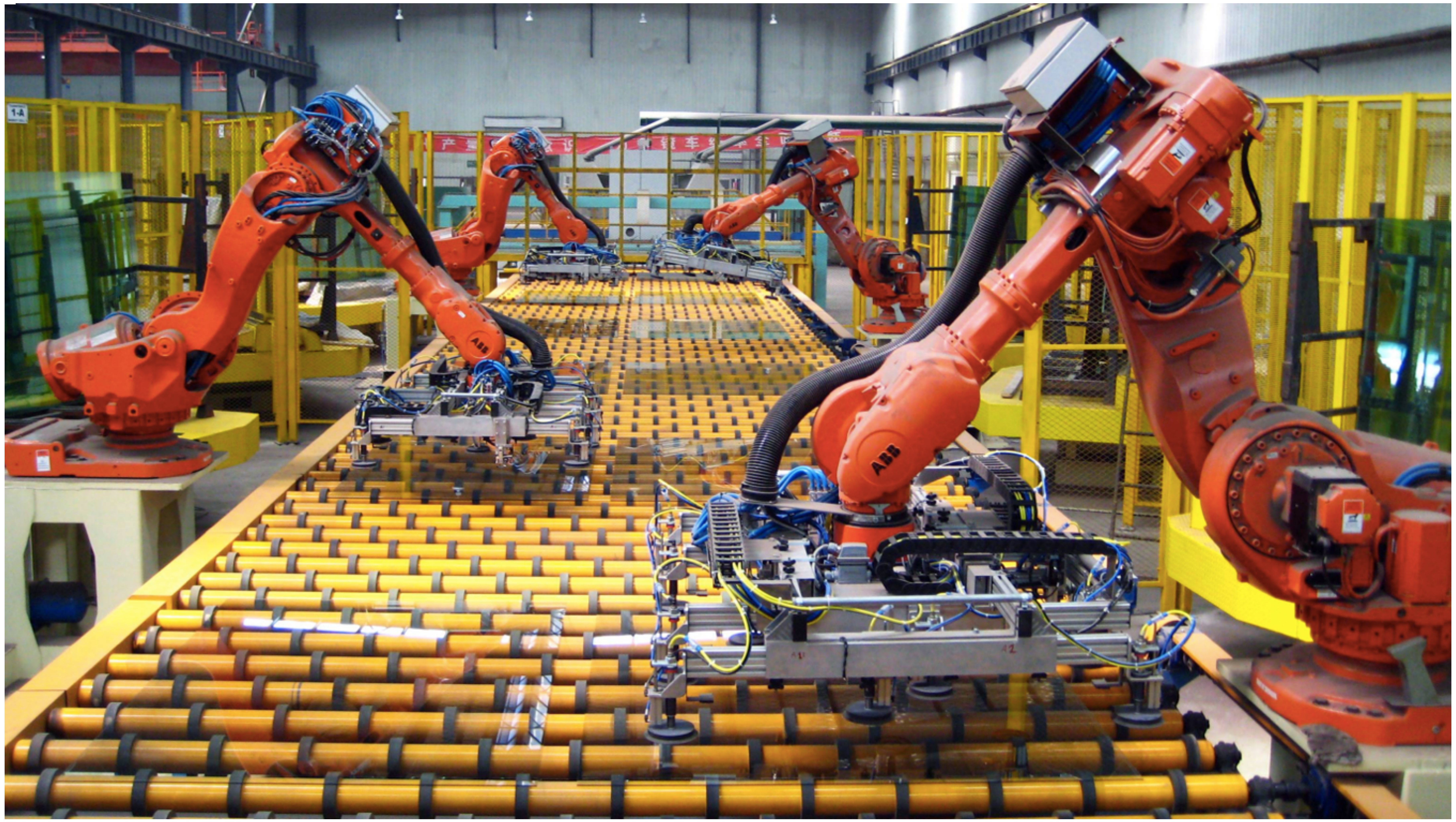
Before & After
BeforeStep & AfterStep
BeforeFeature & AfterFeature
BeforeTestRun & AfterTestRun
BeforeScenarioBlock &
AfterScenarioBlock

Putting it all together

Software development is hard.

Understanding what your
methods and tools
are good for
can make it easier.

Tools are not essential



BDD redefined

BDD is a second-generation, outside-in, pull-based, multiple-stakeholder, multiple-scale, high-automation, agile methodology.

It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters.

Dan North

http://skillsmatter.com/podcast/java-jee/how-to-sell-bdd-to-the-business

It’s about collaboration

When you do BDD/Specification by Example and Outside-in, regression tests fall out at the other end. They are a by-product of those activities. Testing isn't the activity itself.

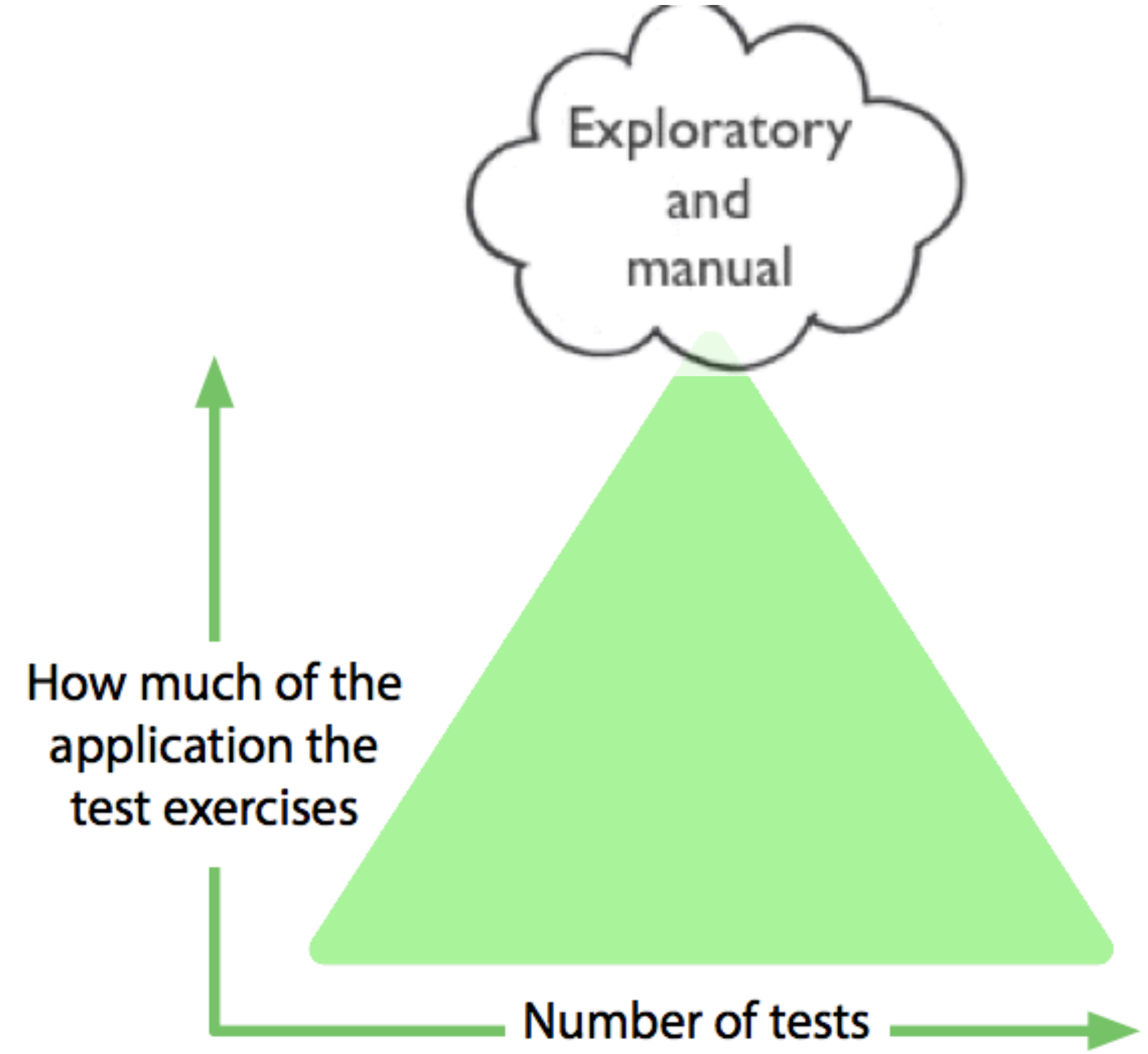
Cucumber is first a foremost a collaboration tool that aims to bring a common understanding to software teams - across roles.

Aslak Hellesøy

https://cucumber.pro/blog/2014/03/03/the-worlds-most-misunderstood-collaboration-tool.html

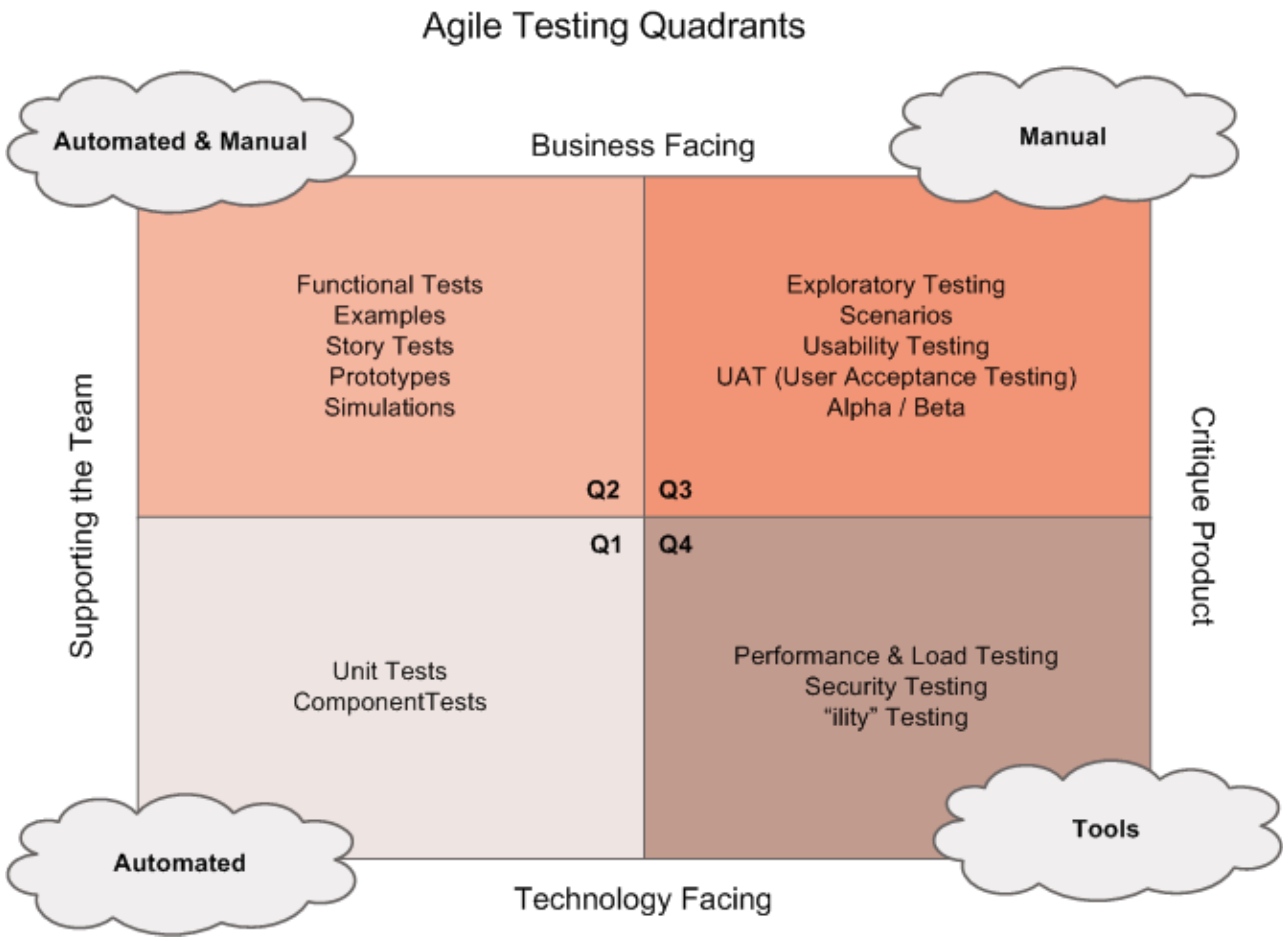
We still need testers

Test = check + explore



http://claysnow.co.uk/architectural-alignment-and-test-induced-design-damage-fallacy/

So much can go wrong!

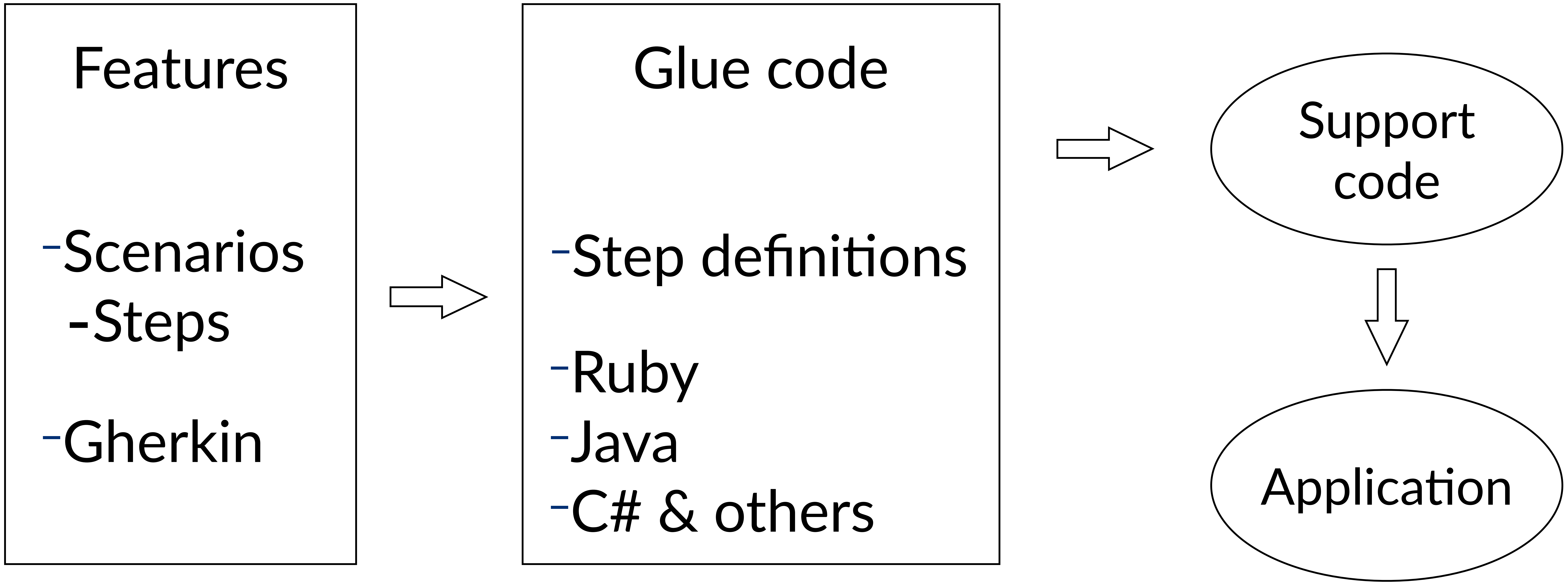


Automation is development

THERE, I'VE JUST BUILT THE LATEST BROKEN FEATURE. LET ME KNOW WHEN YOU FIND THE BUGS.
THANK YOU. I'M ETERNALLY GRATEFUL FOR YOU. MAY YOU LIVE FOR EVER AND BLESSED BE YOUR OFFSPRING TO THE 10TH GENERATION.

http://1.bp.blogspot.com/_YzKCMr-tcMM/TFLIGeqTJfI/AAAAAAAAARQ/AhNW62KX5EA/s1600/cartoon6.jpg

Understand designs



<role> will write scenarios

This page intentionally blank.

More does not mean better

It's about confidence

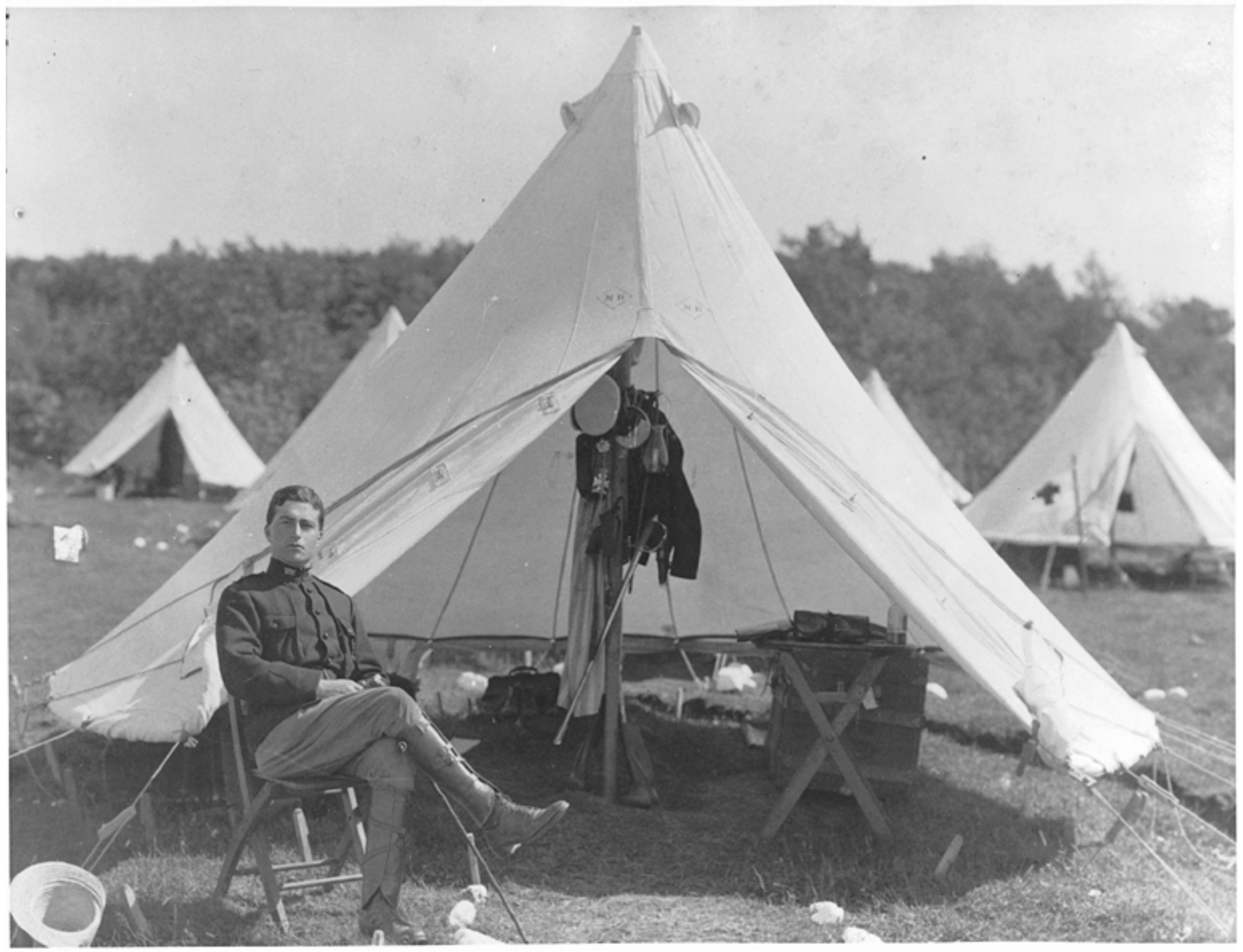
"I get paid for code that works, not for tests, so my philosophy is to test as little as possible to reach a given level of confidence …

"I suspect this level of confidence is high compared to industry standards"

Kent Beck

http://stackoverflow.com/questions/153234/how-deep-are-your-unit-tests/153565#153565

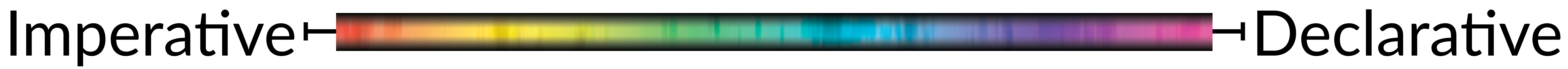
How many guy ropes?



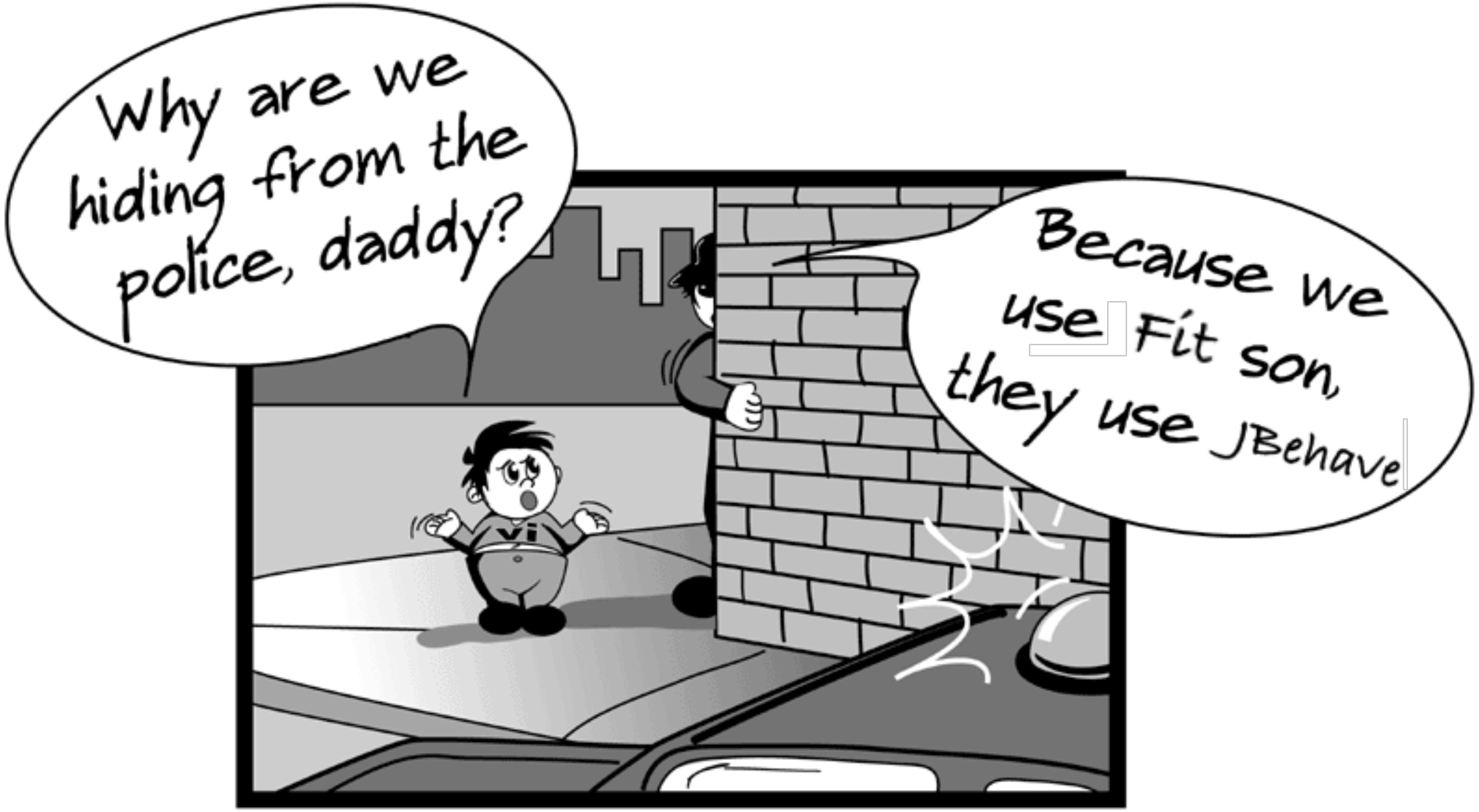
Clarity over detail

Keep it focussed

Avoid incidental details



TDD, ATDD, BDD, SBE …


Why are we hiding from the police, daddy?
Because we use Fit son, they use JBehave
vi

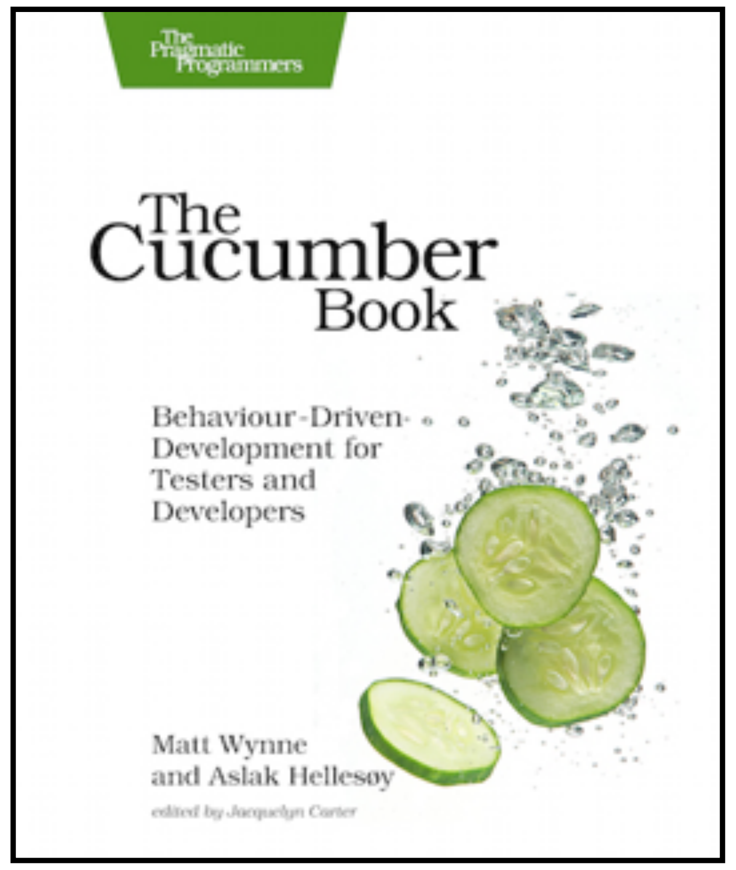
What's the difference between TDD, ATDD, BDD and SbE?
They're called different things

http://lizkeogh.com/2011/06/27/atdd-vs-bdd-and-a-potted-history-of-some-related-stuff/

The best TDD practitioners…

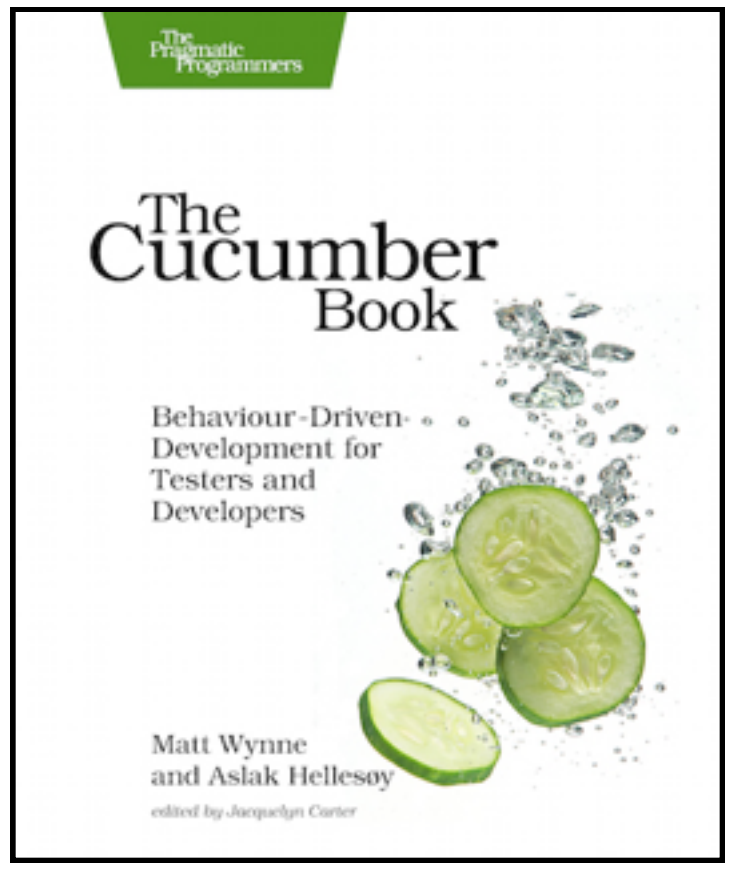
- Work from the **outside-in**, ***i.e. test-first***
- Use **examples** to clarify their requirements
- Develop and use a **ubiquitous language**

cucumber ltd


The Pragmatic Programmers
The Cucumber Book
Behaviour-Driven-Development for Testers and Developers
Matt Wynne and Aslak Hellesøy

The best BDD practitioners…

- Focus on **value**
- Discover examples **collaboratively**
- Create **living documentation**


The Pragmatic Programmers

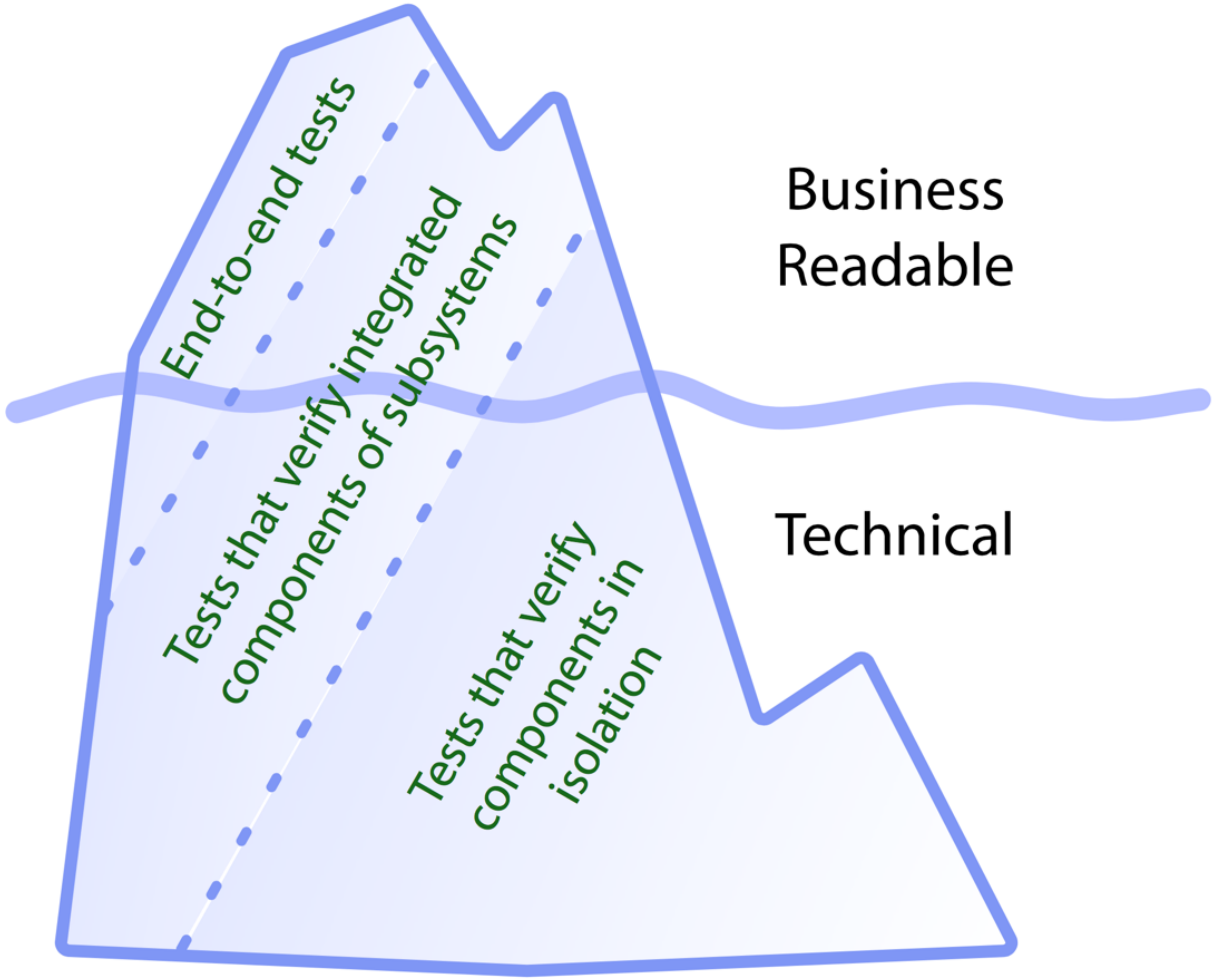
The Cucumber Book

Behaviour-Driven-Development for Testers and Developers

Matt Wynne and Aslak Hellesøy

edited by Jacquelyn Carter

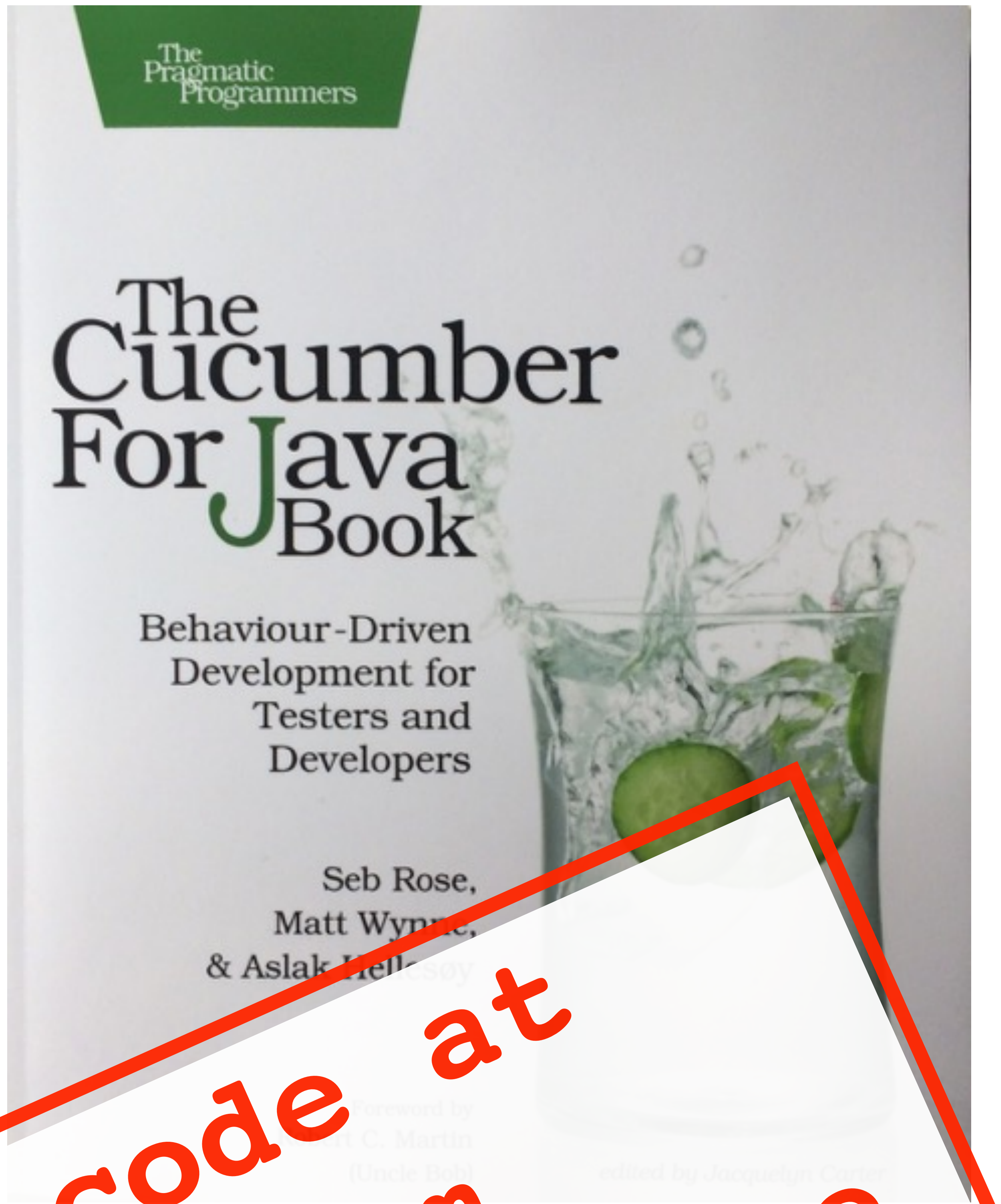
JUnit or Cucumber?


End-to-end tests
Tests that verify integrated components of subsystems
Tests that verify components in isolation
Business Readable
Technical

Take aways

- BDD is about collaboration
- You need a ubiquitous language
- Cucumber can power conversation
  - and produce living documentation
  - but it's no substitute for testing!

cucumber ltd


The Pragmatic Programmers
The Cucumber For Java Book
Behaviour-Driven Development for Testers and Developers
Seb Rose, Matt Wynne, & Aslak Hellesøy

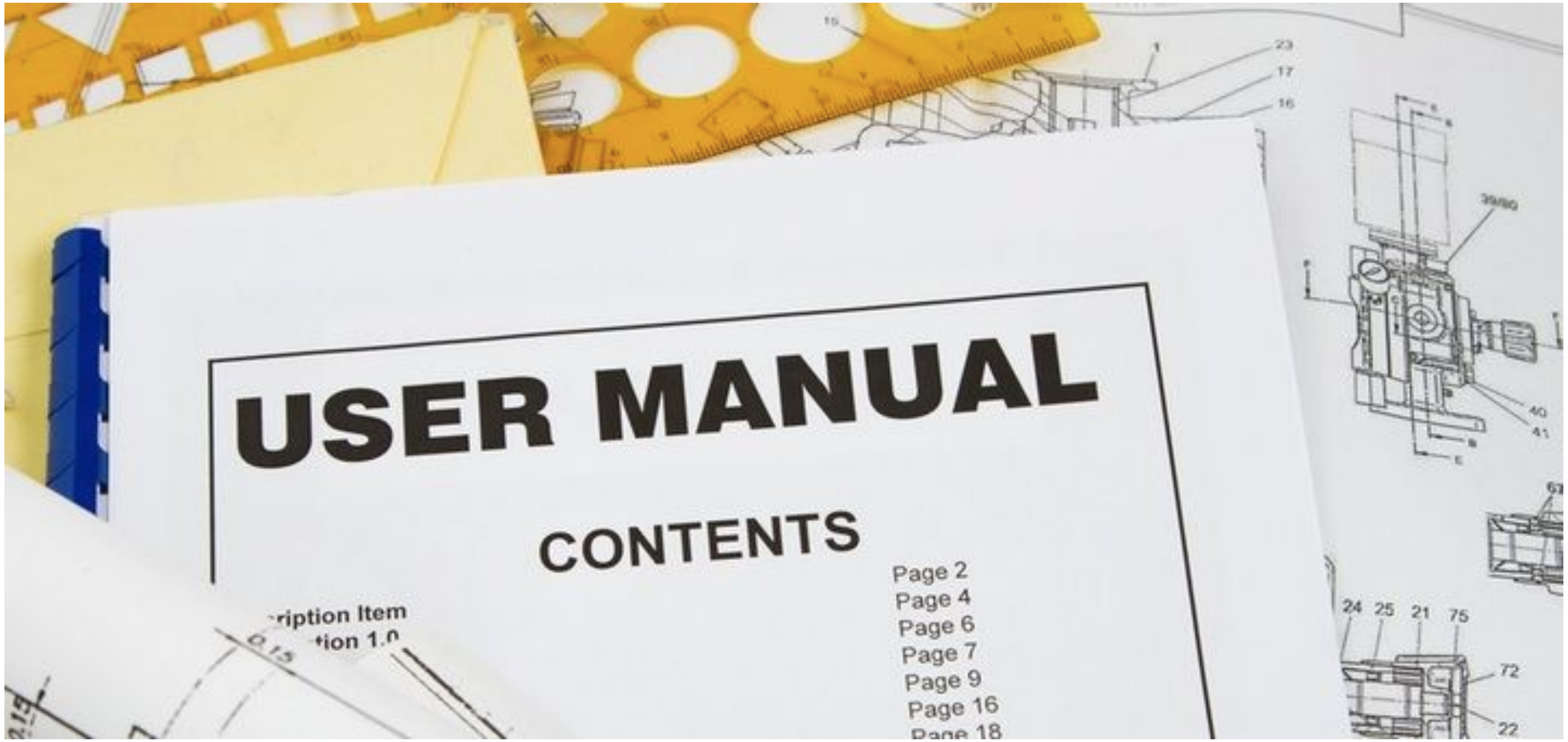
Questions?

Discount code at pragprog.com
Java_One_25%_Off_Cuke
Valid for 1 month

Seb Rose

Twitter: @sebrose
Blog: www.claysnow.co.uk
E-mail: seb@cucumber.io

Documentation



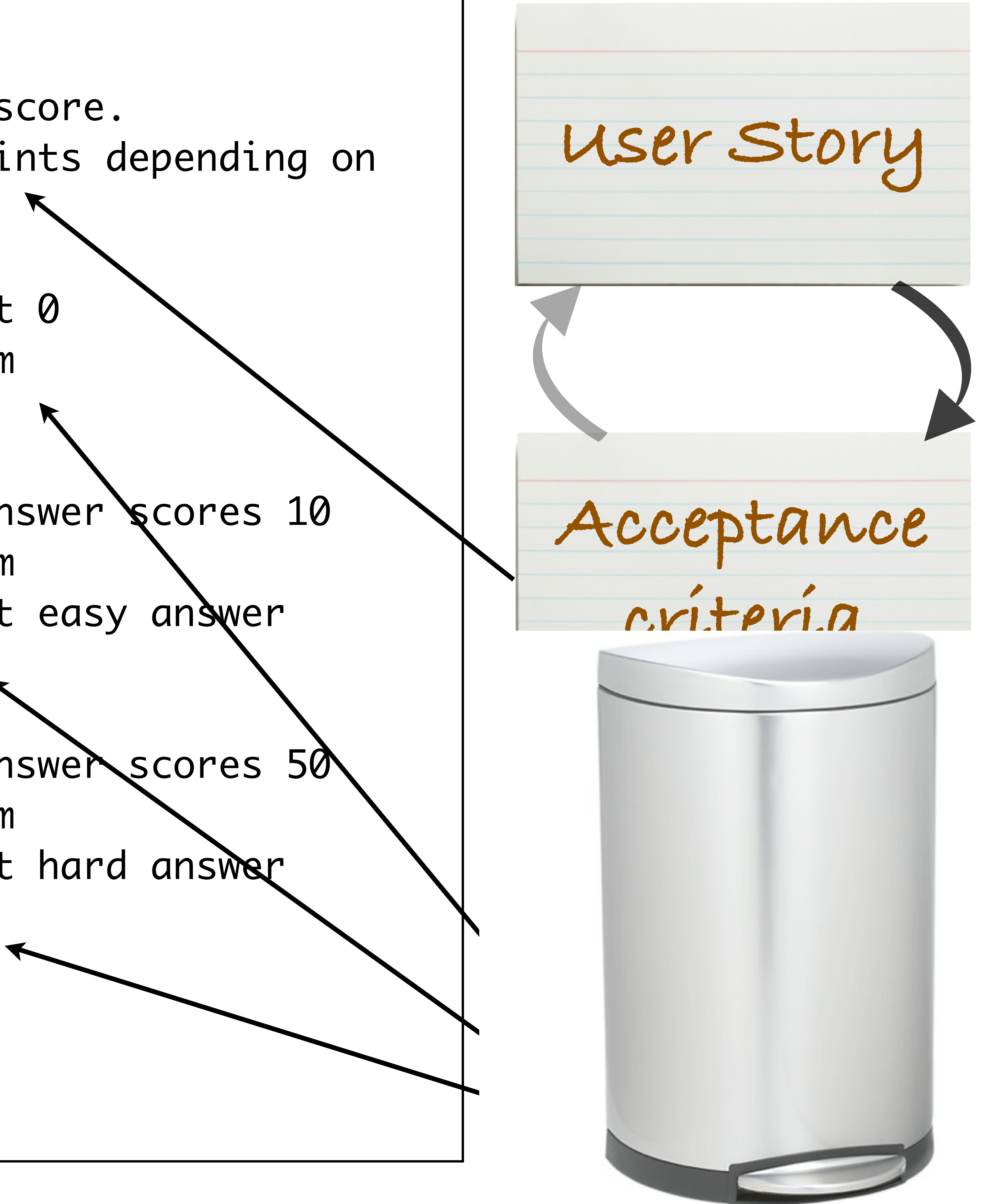
https://cemarking.net/wp-content/uploads/2014/01/User-Manual_featured.jpg

```gherkin
Feature: Team Scoring
  Teams start with zero score.
  Correct answer gets points depending on
  how difficult it is.

Scenario: Score starts at 0
  Given I register a team
  Then my score is 0

Scenario: Correct easy answer scores 10
  Given I register a team
  When I submit a correct easy answer
  Then my score is 10

Scenario: Correct hard answer scores 50
  Given I register a team
  When I submit a correct hard answer
  Then my score is 50
```



Step outcome

- Passed
  - No exception thrown
- Pending
  - Pending exception thrown
- Undefined
  - No matching step definition found
- Failed
  - Any other exception thrown
- Skipped
  - Steps that follow a Pending, Undefined or Failed step
- Duplicate
  - Step matches more than one step definition

Scenario outcome

- Passed
  - All steps passed
- Failed
  - Any step failed
- Undefined
  - No failed steps
  - At least one undefined step
- Pending
  - No failed or undefined steps
  - At least one pending step