

# Busting the BDD myths

<http://gojko.net>

@gojkoadzic

Behaviour Driven Development

Specification by Example

Story Testing

Acceptance Test Driven Development

**“automation applied to an efficient operation will magnify the efficiency....**

...automation applied to an inefficient operation will magnify the inefficiency”

- Bill Gates.

# Cucumber in rear was 'failed suicide bid'

★ Save this story to read later

 Tweet

0

 Recommend



Be the first of your friends to recommend this.

**A HONG Kong man, taken to the hospital to have a cucumber removed from his bottom, told doctors he inserted it in a suicide attempt.**

*The Sun* reported [Chin Wei](#), 62, said the method was a variation of the Japanese ritual suicide hara-kiri - usually carried out with a sword plunged into one's own stomach.

myth #1:  
Instoolation

Instoolation (n): belief that process problems can be solved by installing a tool

As you can see, Cucumber allows you to easily describe the behaviour of your new feature. In fact, it's that good that some of our customers are using Cucumber descriptions as the acceptance criteria on their project's stories. So where's the catch? Well, not everything in cucumber is as easy as it may look. Even though cucumber has lot of pre-defined "phrases" to describe the behaviour of an application, you soon find out that sometimes it can be really tricky to preserve the readability of stories that are more complex than just: "And I should see 'a yellow box" for example. When it gets tricky it gets really time consuming. And even if you know that implementing the related functionality is really easy - writing the proper, readable cucumber test that makes sense is sometimes very hard.



## Fatal flaws

Let me first summarize the fatal flaws of Fitnessse, and address them in more detail immediately thereafter:

- In practice it turns out to be extremely hard to get actual business involvement for the still rather technical (Wiki based) integration tests.
- "Programming" in Fitnessse has a steep learning curve and remains a nuisance due to its obscure syntax constructs
- Search the web on Fitnessse topics gives "health" related results.
- Resolving classpath issues suddenly becomes a daily practice (again?)
- Exceptions thrown in constructors disappear into oblivion

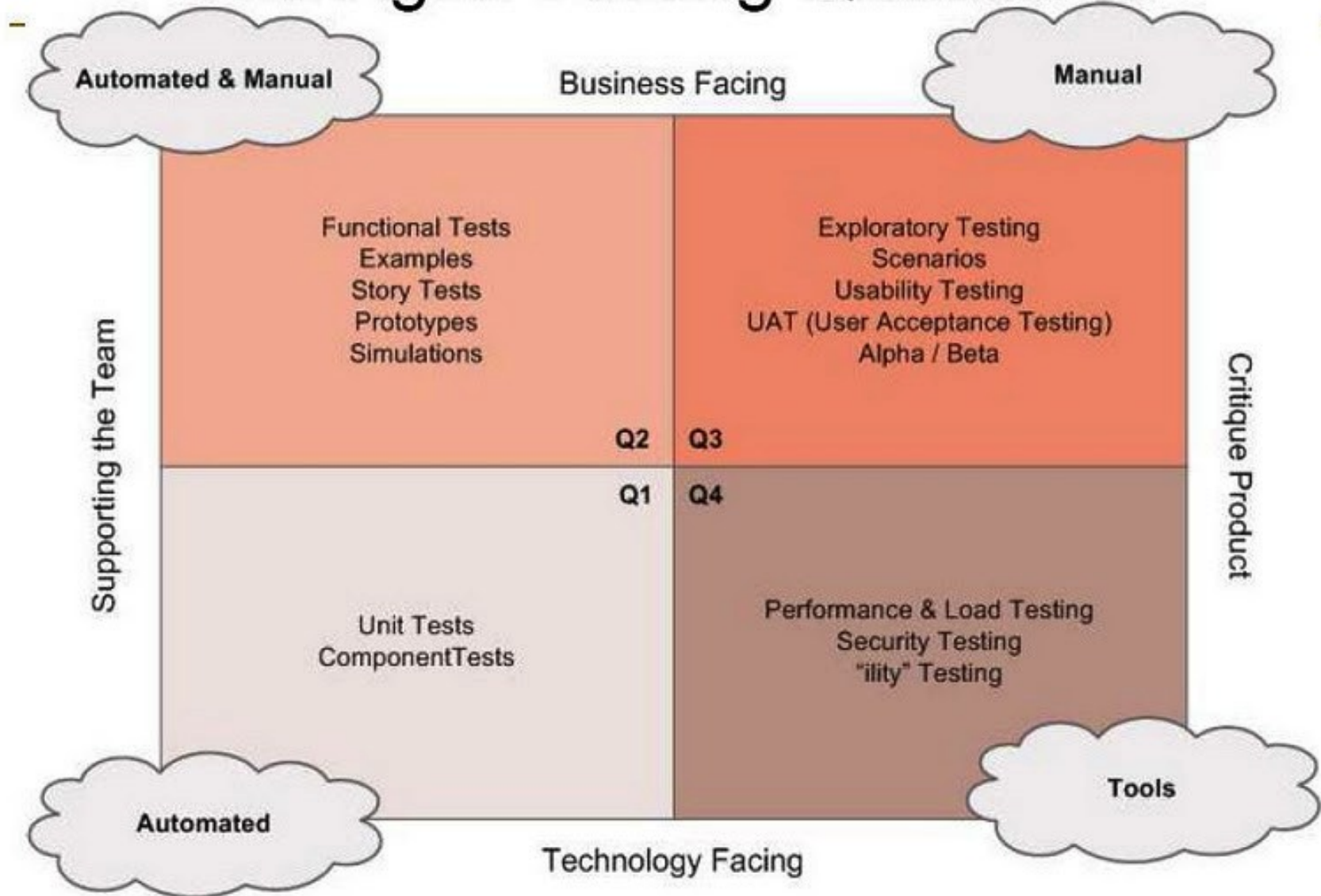
**java** javascript jdbc jetty jpa  
**junit linux maven**  
**portal** proxy roller scrum  
**spring** tomcat  
**websphere wicket**

### Links

[IT-Essence](#)  
[JavaPassion.com](#)  
[TheServerSide](#)  
[Zeger's home](#)  
[+Blogs](#)  
[+IBM DeveloperWorks](#)

[Front Page](#)  
[Weblog](#)  
[Login](#)

# The Agile Testing Quadrants



From "Agile Testing" Crispin/Gregory

myth #2:  
Businesting

**businesting (n): belief that business users should write acceptance tests**

Does cucumber suck? | Illite x

www.software-testing.com.au/blog/2010/08/31/does-cucumber-suck/

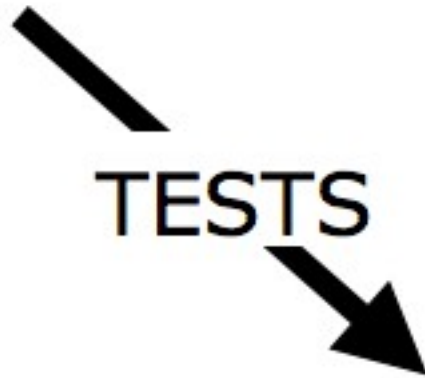
In my experience, I'm seeing -

- Customers/non-programmers never write the tests (because everything is given-when-then. They just want to tell us what to specify everything in that format anyway).
- Customers/non-programmers write the tests but it focuses on readability rather than other testing that seems to add more value.
- The tests are written in English, but what the test actually does is to convert english phrases into code (so there's no guarantee it tests what it's supposed to).
- Avoidance of conversation (ie. tests as contracts).
- Cucumber and the related tools (through the toy examples they provide).



**REQUIREMENTS**





**TESTS**



myth #3:

Acceptegration



Acceptegration (n): belief that tests are either unit or “other”, a mix of acceptance, integration and everything else



**@lroal**

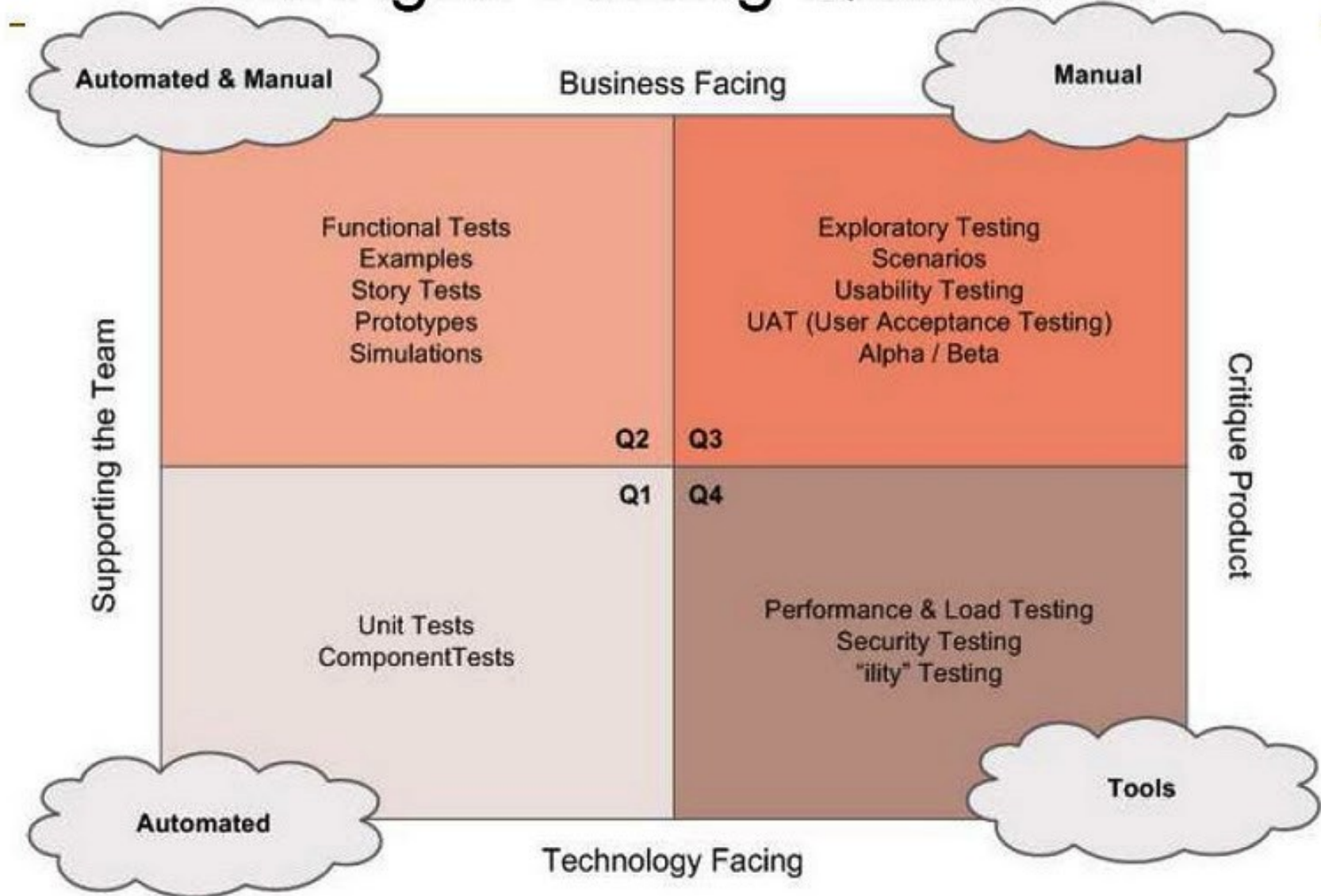
Lars-Erik Roald

After all, isn't unit testing about isolating the class for test. Testing a set of classes is called by another name: acceptance tests.

8 Jan via web ☆ Favorite ↻ Retweet ↩ Reply

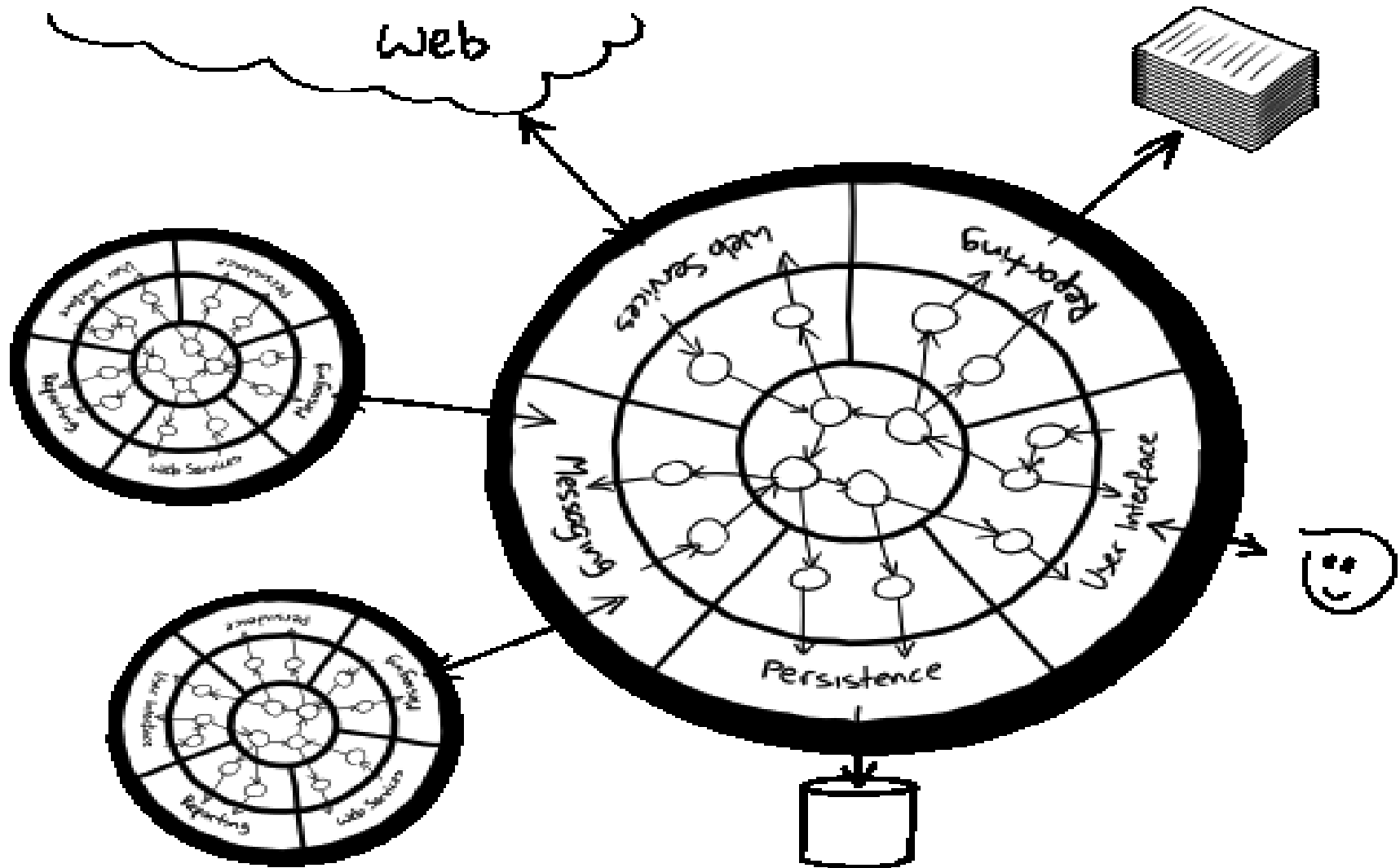
<http://twitter.com/#!/lroal/status/23754744904294400>

# The Agile Testing Quadrants



From "Agile Testing" Crispin/Gregory

# Ports & adapters



From "Growing Object Oriented Software" Pryce/Freeman

myth #4:  
Resolution

Resolution (n): <insert role> should do BDD in isolation (eg testers do everything)

▲ bryanlarsen 331 days ago | link | parent

Cucumber is very expensive -- it requires you to maintain a set of rules to map English to ruby. And it causes people to continually fiddle with stuff just so that the English reads write. And it adds an abstraction layer, which are also "expensive".

Great expense, little gain -> actively harmful to projects.

You end up spending lots of "testing time" fiddling with cucumber stuff. It feels productive, but you'd have more, better tests writing straight into webrat

<http://apps.ycombinator.com/item?id=1944179>

Stories

Programming

Testing

---



Functional gaps  
Inconsistencies

Misunderstood  
requirements



Stories

Programming

Testing

---



Functional gaps  
Inconsistencies  
Misunderstood  
requirements

Unexpected  
stuff

myth #5:  
Longression

Longression (n): belief that the long term value of BDD is in regression testing



## Using FitNesse rather than NUnit



As I've understood it, there's a couple of cases when you may want to use FitNesse.

6



- You want to do acceptance testing rather than unit testing.
- You want to use it as a communication tool with the stake holder.
- You want to do large scale tests rather than granular tests.
- You want non-technical people to write the tests.



1

My personal opinion and experience is:

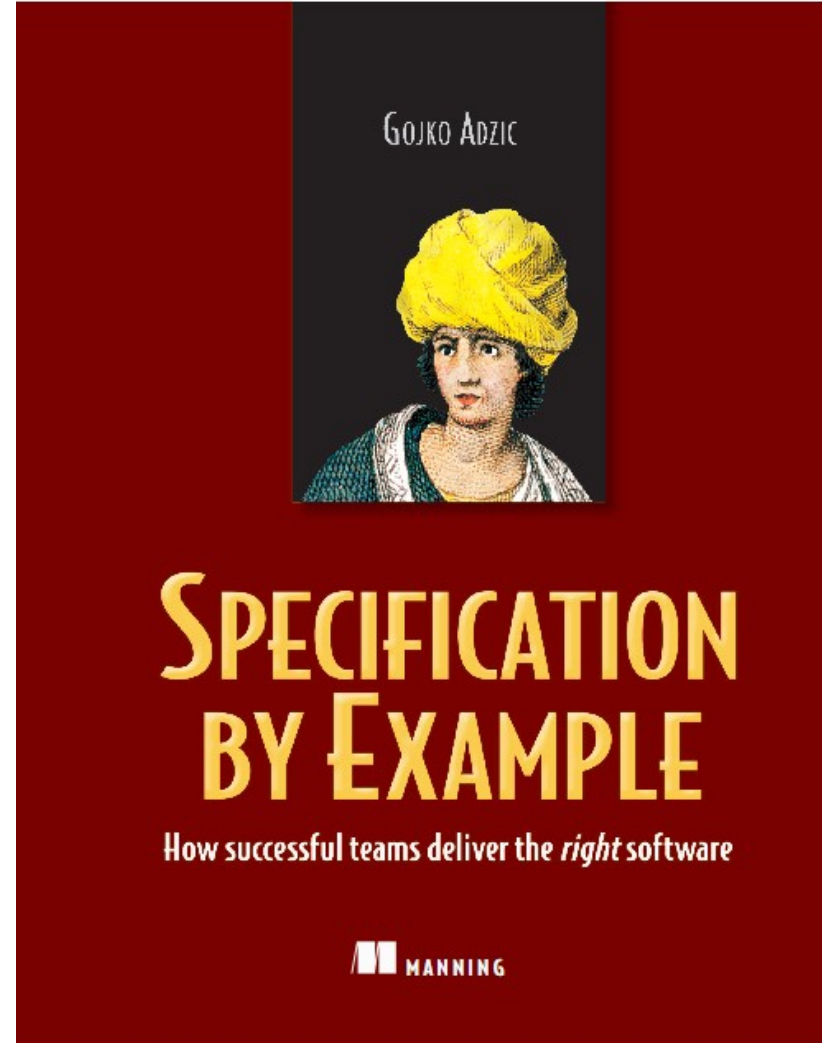


- I can do acceptance testing using tools such as NUnit. Fitnesse is basically just a framework for calling public methods on a .NET assembly, which can be done using NUnit to.
- The stakeholders tell me what they want. They are rarely interested in writing Fitnesse tests. They seem to have other/better/more important things to do.

# Five ways to misuse FIT

- Misuse #1: Use Fit for Test Automation
- Misuse #2: Customer? What Customer?
- Misuse #3: Write Integration Tests
- Misuse #4: Spin Your Propeller
- Misuse #5: Specify Everything

<http://jamesshore.com/Blog/Five-Ways-to-Misuse-Fit.html>



SpecificationByExample.com  
gojko.net