Challenges of adopting Behaviour-driven Development (BDD)

Fernando Bernardino
<tony.bernardino@gmail.com>

Advance Programming SG

13th April 2017

Contents

- What is Behaviour-driven Development (BDD)?
- Different approaches to a problem
- Demo
- What are the costs of using BDD?
- What are the impacts of BDD?

What is Behaviour-Driven Development?

Wikipedia:

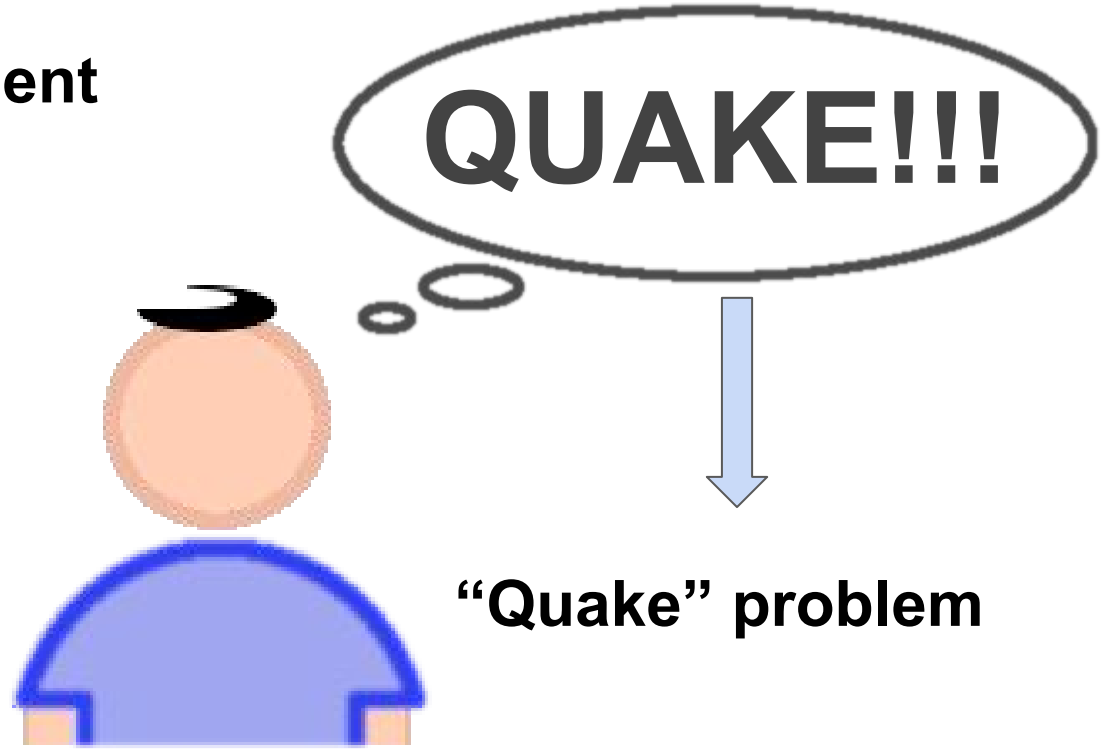
> *"(...) behavior-driven development (BDD) is a* **software development process** *that (...) combines the general techniques and principles of TDD with ideas from domain-driven design and object-oriented analysis and design. (...) As such it is a* **natural extension of TDD testing** *in general."*

Who am I?

- Fernando (Tony) Bernardino
- Senior Java Developer at Deontics. Ltd
  - Decision Support in Health
- Main areas of interest:
  - Agile/Lean methodologies
  - Continuous Delivery

Who am I?

As a Student



Who am I?

“Quake” Problem

Given a 2D point and a closed 2D shape defined by a set of vertex, determine if the point is contained inside the shape

Who am I? v1.0

“Quake” Solution

- 20 lines of code
- High performance
- Highly optimized

Who am I? v1.0

Problems:

- No one understood it, not even **me** after a while
- No use
- No value

Who am I? v2.0

Worry about readability

- 👍 Documentation
- 👍 Comments
- 👍 Structure
- 👍 Aesthetic

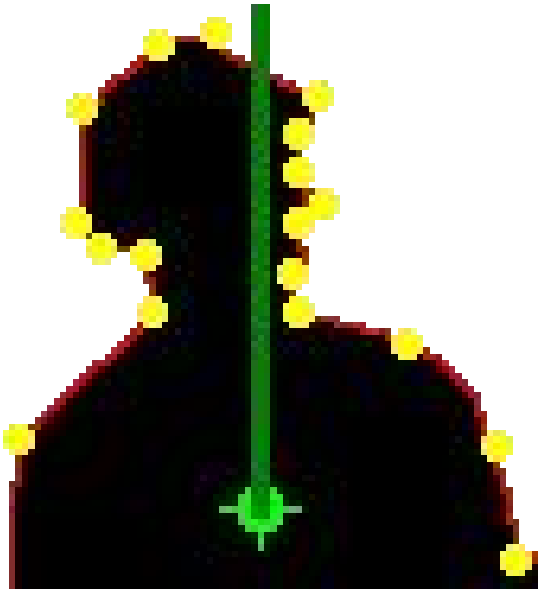
Who am I? v2.0

“Quake” solution

```
/*

Drawing an ascending vertical
line starting at the point,
count the number of times it
intercepts the limits of the
shape. Pair means it is OUT,
odd means it is IN. …

*/
```



Who am I? v2.0

Problems

- Over documenting
  - Easy to get out-of-date
  - Hard to change
- Over-engineering
- **Code ownership**

Who am I? v3.0

Test-driven Development (TDD)

- 👍 Test-first development
- 👍 Red, Green, Refactor
- 👍 Almost 100% code coverage
- 👍 Less bugs pass the development phase

Who am I? v3.0

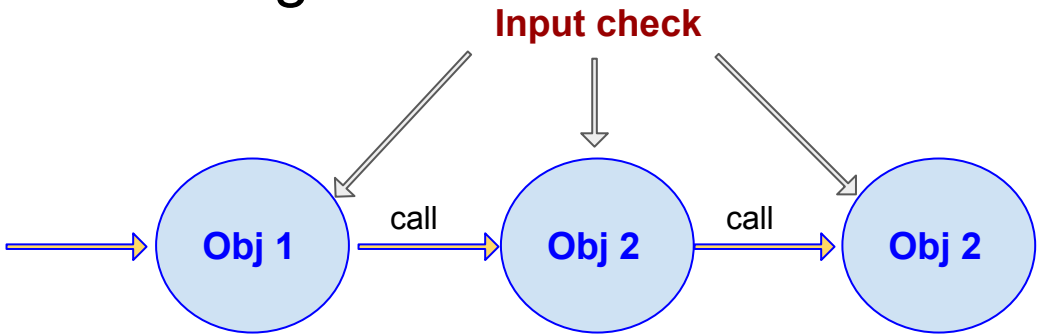
“Quake” Solution

- Tests
  - isIn_nullArray_throwException
  - isIn_emptyArray_throwException
  - isIn_1PointArray_throwException
  - isIn_2PointArray_throwException
  - …

Who am I? v3.0

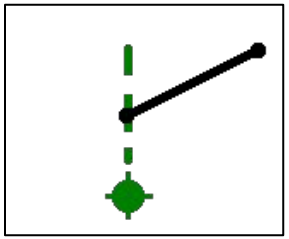
Problems

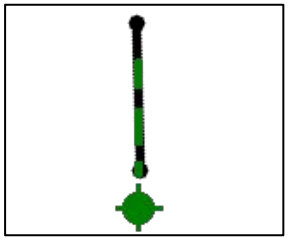
- Low level (isIn?! array?! null?!)
- Focus on little details
- Too much input checking



Who am I? v3.0

“Quake” Solution - corner case tests

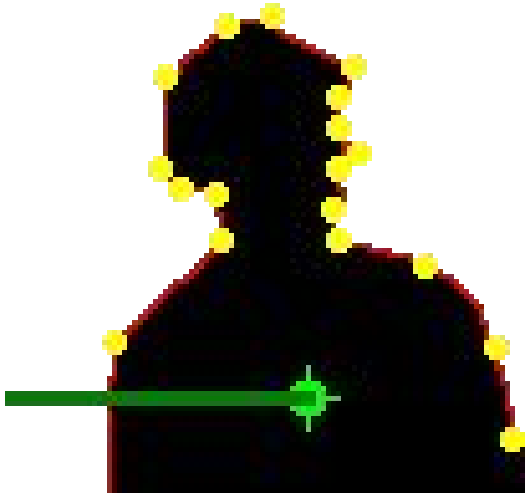




. . .

Who am I? v3.0

What if algorithm changes?



Who am I? v4.0

Behavior-driven Development

- 👍 Test behavior, not code
- 👍 Higher level of abstraction
- 👍 More focused on value
- 👍 Tests break less
- 👍 User friendly

Who am I? v4.0

“Quake” solution - Gherkin

```
Given a 2D shape When a contained point is passed  Then it returns IN

Given a 2D shape When a NOT contained point is passed  Then it returns OUT

Given a 2D shape When a point matching limits  Then it returns LIMIT

...
```

BUT .. a NORMAL user does not understand it ...

Who am I? v4.0

“Quake” solution - Gherkin revisited

Given an enemy **When** I fire and hit him/her **Then** the enemy dies

Given an enemy **When** I fire and miss him/her **Then** the enemy stays alive

Given an enemy **When** I fire and scrape him/her **Then** the enemy starts bleeding

Demo!

Demo!

Problem

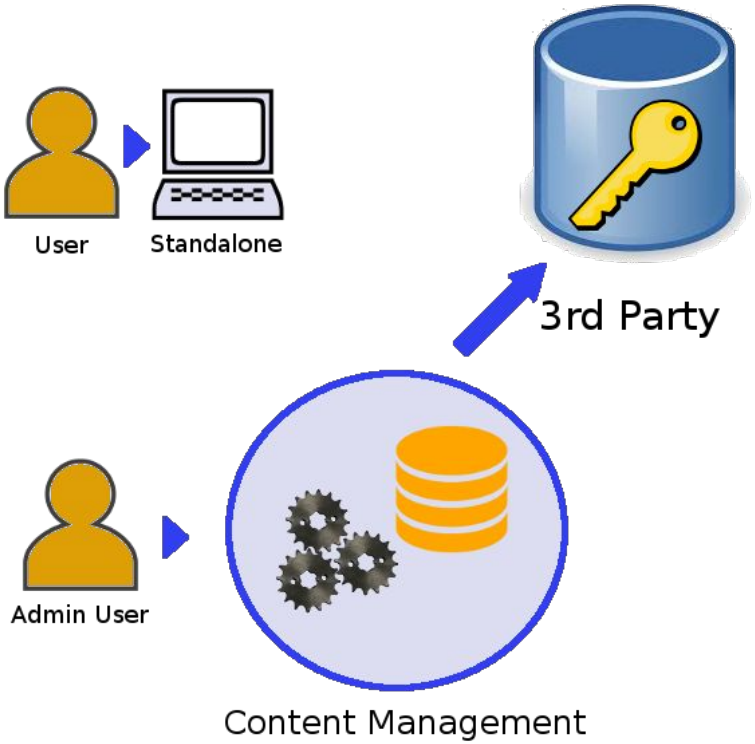
When a new user is registered in the system, how does the password gets chosen?

Avoid passing password over email.

Ideally the user should be pick it.

Demo!

Architecture



Demo!

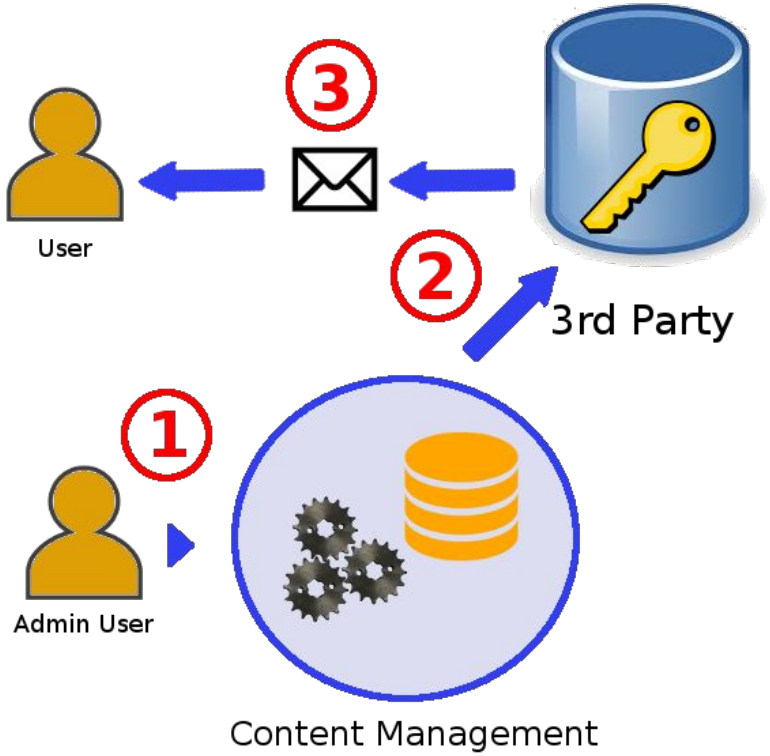
Requirement (after 2 days analysis)

When a user is created, it should be possible to provide her/him a location where the password can be changed without knowing any previous password.

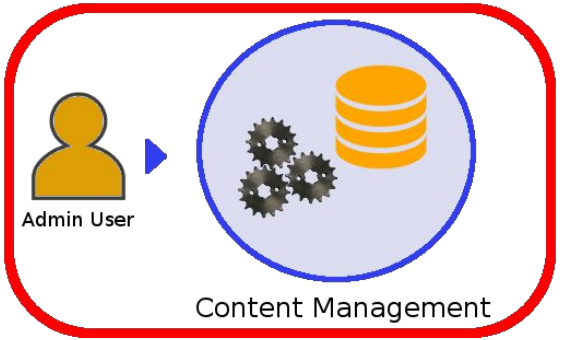
Demo!

In practical terms:

1. Create user with “Change Password” request
2. 3rd Party is requested to send “Change Password” email
3. 3rd Party sends email to user



Demo!



User Acceptance criteria - Gherkin

Given `John` in not a registered user

And `John` should pick his own password

When an Admin user registers `John`

Then the user for `John` gets created

And a “Change Password” email is sent to him

Demo!

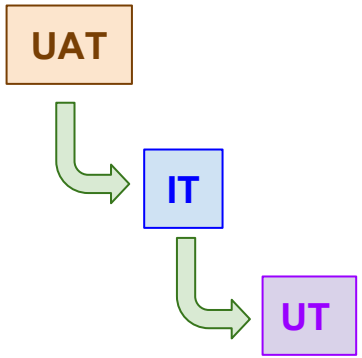
Level of tests in the demo:

- User acceptance UAT - Gherkin
- Integration (modules) IT
- Unit (class) UT

Demo!

Acceptance UAT **& Integration** IT **& Unit** UT **tests**

- Cascading implementation



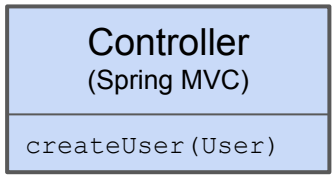
Demo!

Implement (User) Acceptance test UAT **:**

- Create 3rd party mocks
- Implement steps (lines)
- Refactor test code
- **Test passed (?!?!?)**
- **Make it fail**
- **0 lines of production code**

Demo!

“BDDing” Integration IT and Unit UT tests

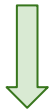



```
When I create a User

with a password change request

Then the user is created

And the request is sent
```



`createUser_withChangePasswordRequest_userCreatedAndRequestIsSent` IT

Demo!

Acceptance UAT **& Integration** IT **& Unit** UT **tests**

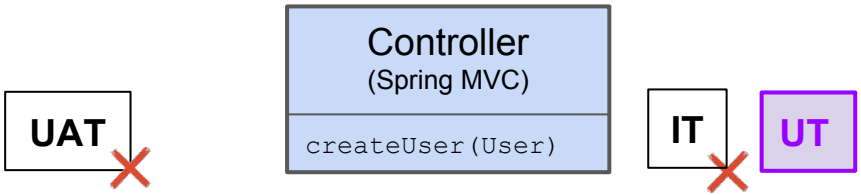
- Behaviour testing overlap

UAT ∩ IT ∩ UT ≠ 0

Demo!

- `Refactor: createUser_` **`withoutChangePasswordRequest`** `_userIsCreated` **`(renamed)`**
  - **Passing!**
- `New: createUser_withChangePasswordRequest_userCreatedAndRequestIsSent`
  - **Failing!**
  - Refactor “request”
  - Fix tests
  - Fix production code
  - **Other tests passing!**

Demo!



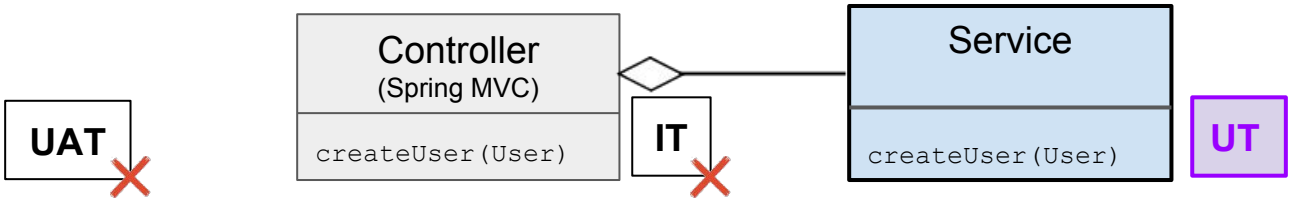

```
createUser_withChangePasswordRequest_userCreatedAndRequestIsSent
    createUser_validUserCreation_200OkReturned
```

IT

UT

(already exists)

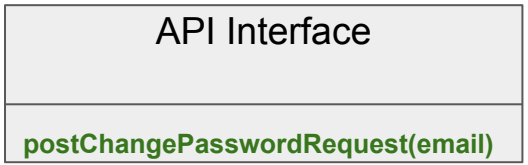
Demo!



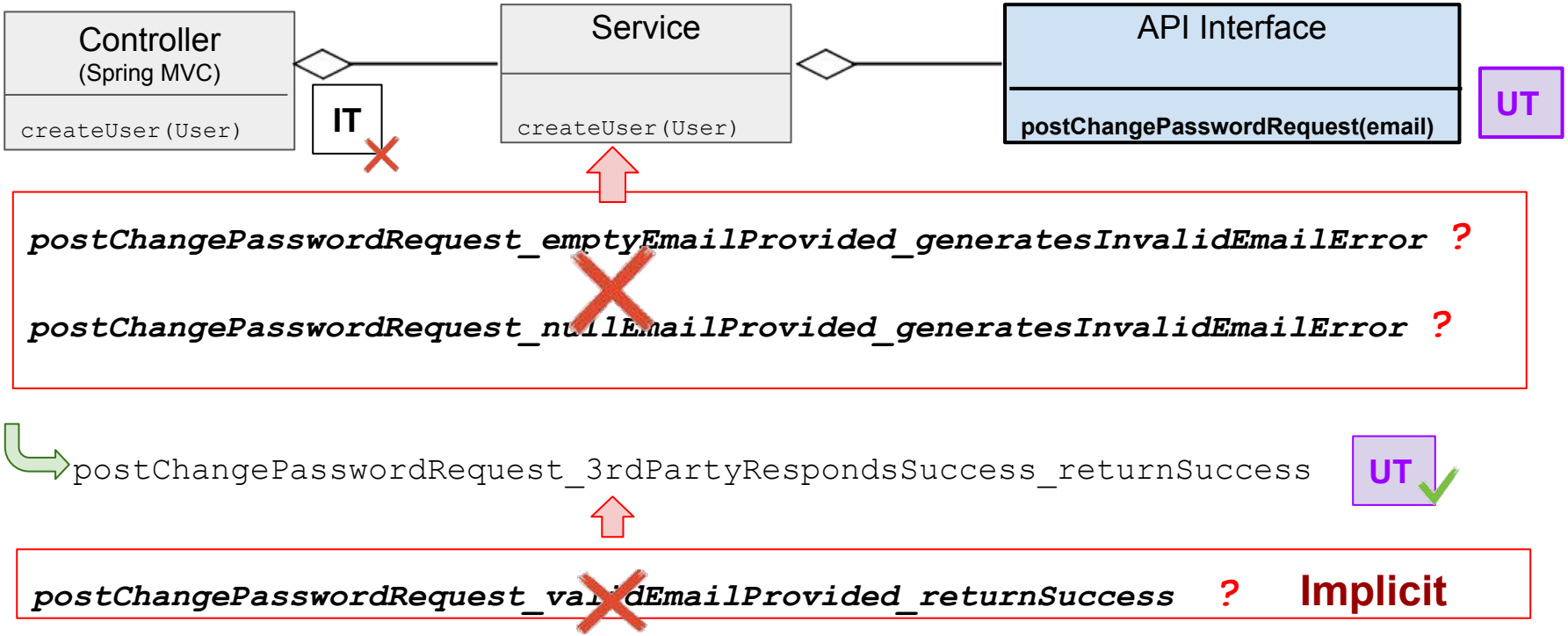
createUser_**withoutChangePasswordRequest**_returnedSuccess **(renamed)** UT

NEW: createUser_withChangePasswordRequest_3rdPartyRequestIsSent UT

New method in 3rd Party API Interface - **postChangePasswordRequest**

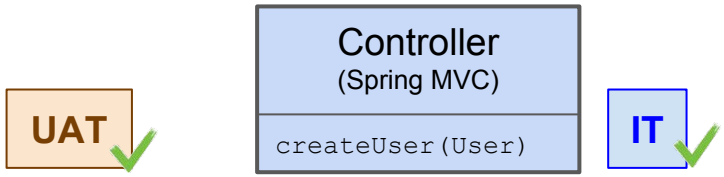


Demo!


Controller
(Spring MVC)
createUser(User)
IT
Service
createUser(User)
API Interface
postChangePasswordRequest(email)
UT
postChangePasswordRequest_emptyEmailProvided_generatesInvalidEmailError ?
postChangePasswordRequest_nullEmailProvided_generatesInvalidEmailError ?
postChangePasswordRequest_3rdPartyRespondsSuccess_returnSuccess
UT
postChangePasswordRequest_validEmailProvided_returnSuccess ?
Implicit

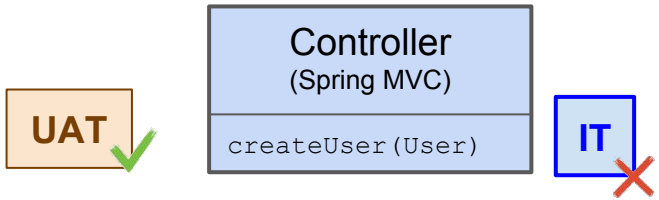
Demo!

Finished implementing first round!



Not always ...

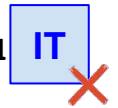
Demo!



Second round

- *Refactor: createUser_withoutChangePasswordRequest_userIsCreated* ***(renamed)***
- *New: createUser_withChangePasswordRequest_userCreatedAndRequestIsSent*

  - **Passing!**
- **New:createUser_3rdPartyChangePasswordRequestReturnsNotFound_notFoundReturned**



  - **Failing!**

Demo!

Done!

But wait, there is more ...

What are the costs of BDD?

Quantitative analysis of Demo:

Code scope	New and changed lines	%
Production	57	<12%
Testing	446	>85%

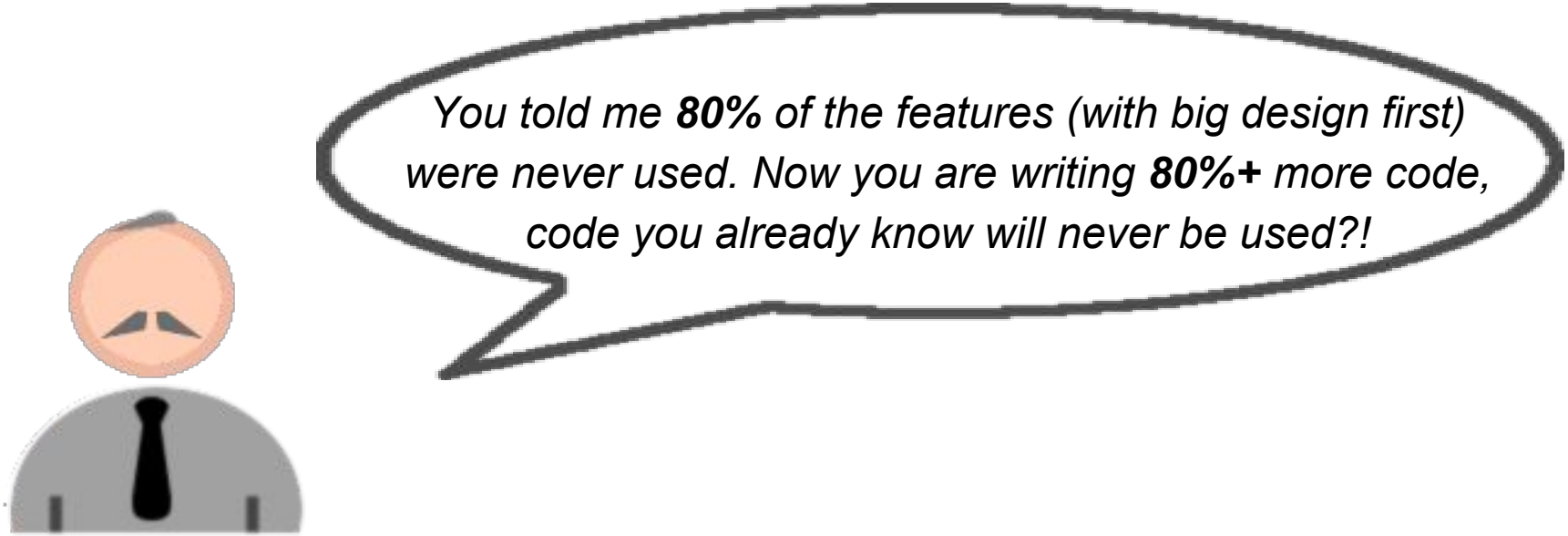
What are the costs of BDD?

Superficial analysis of Demo results:

- **More code (80+%)**
  - **more time**
  - **more complexity**
  - **no added feature**
  - ***more bugs (on development!)***

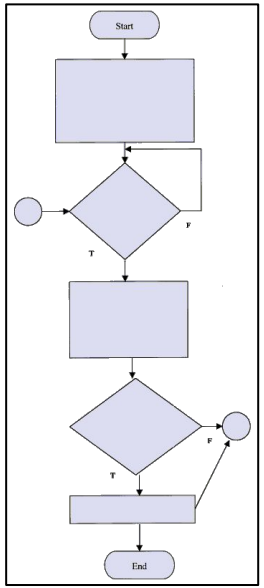
What are the costs of BDD?

Management analysis of Demo results:



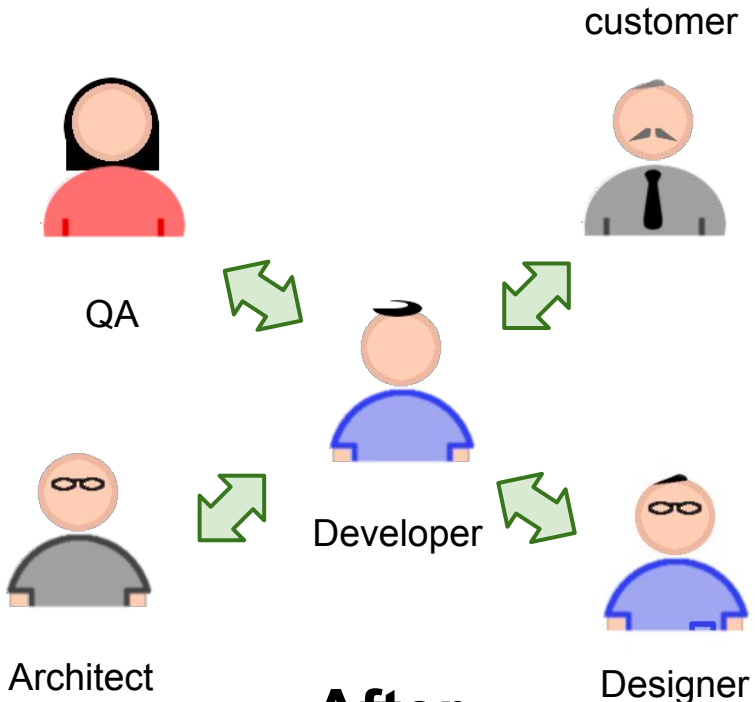
What are the impacts of BDD?

As a developer



Before

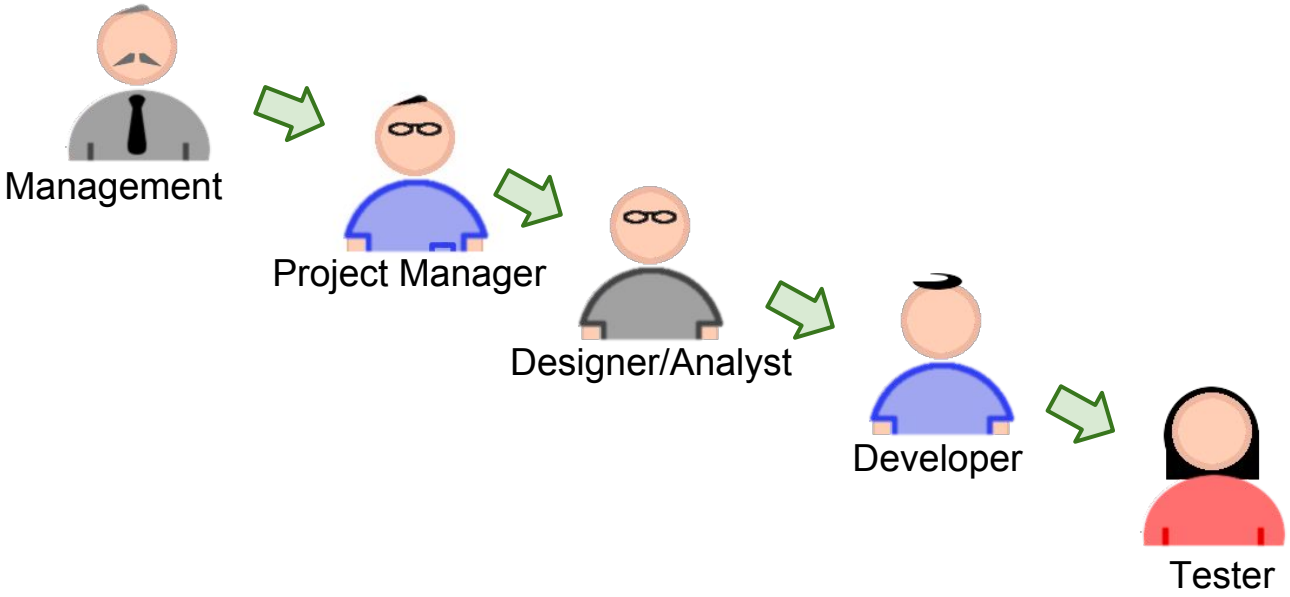
VS



After

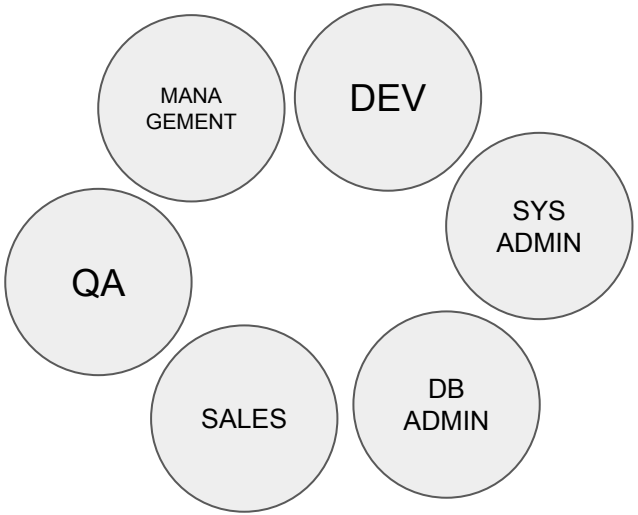
What are the impacts of BDD?

As a company - Before



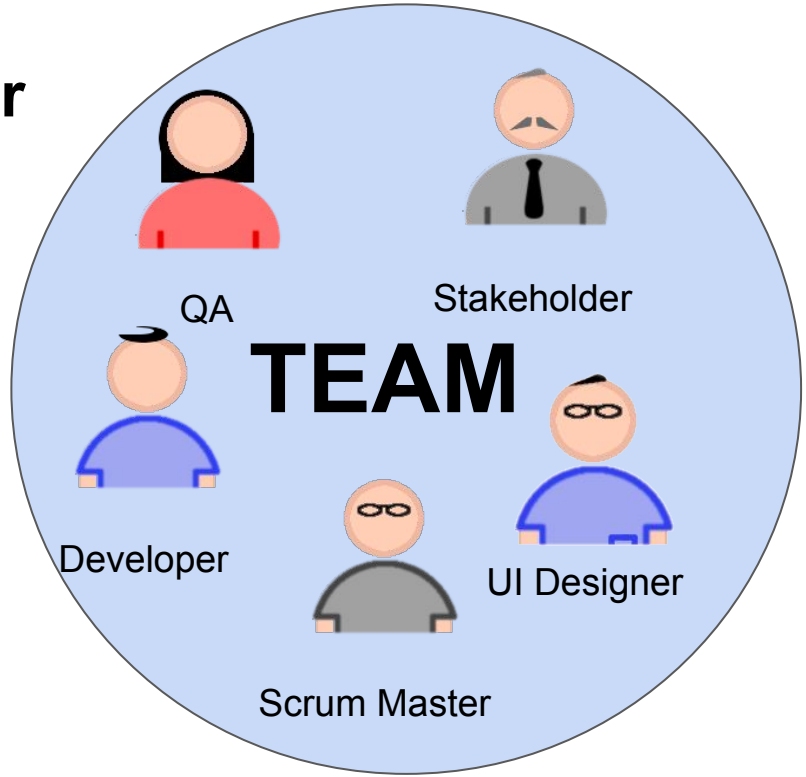
What are the impacts of BDD?

As a company - Before



What are the impacts of BDD?

As a company - After



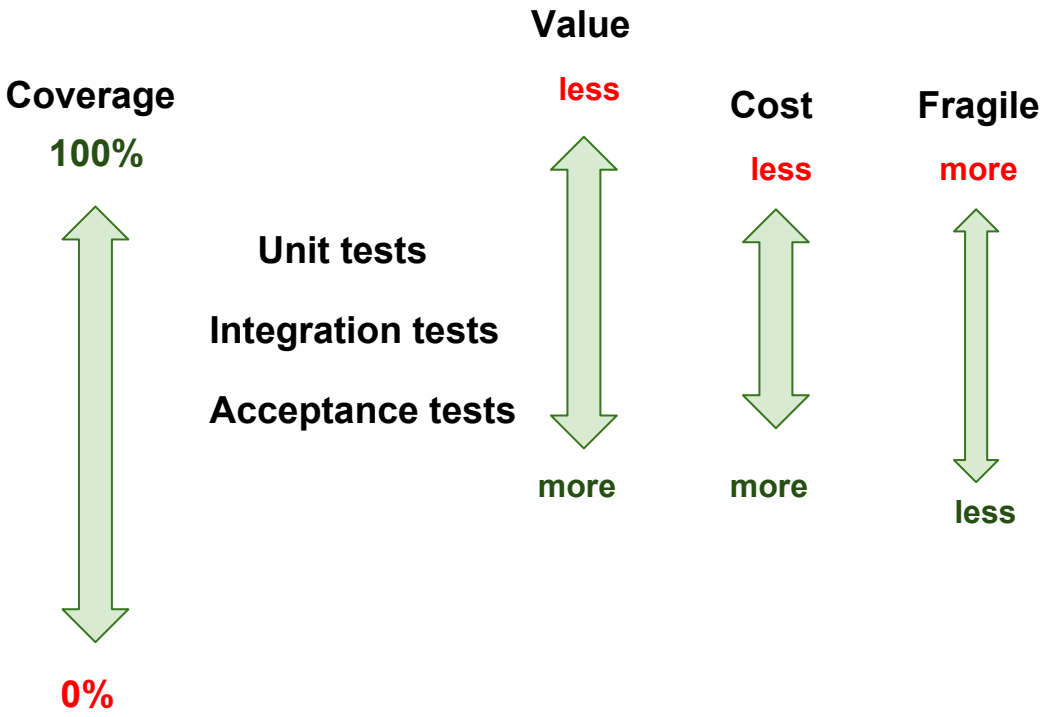
Conclusion

What are the challenges of adopting Behaviour-driven development?

Us - People

- Change is hard
- Learning curve (£££)
- Benefits hard to quantify
- Costs easy to quantify

Related ...


Coverage
100%
0%
Unit tests
Integration tests
Acceptance tests
Value
less
more
Cost
less
more
Fragile
more
less

Related ...

We are just covering simple coding! Some much more is out there...

What version are you?

References and further reading

1. “Clean Code: A Handbook of Agile Software Craftsmanship”, Robert C. Martin, Prentice Hall, 2009
2. “Domain-driven Design”, Eric J. Evans, Addison-Wesley, 2003
3. “The Cucumber Book: Behaviour-Driven Development for Testers and Developers”, Matt Wynne et al, Pragmatic Bookshelf, 2012
4. "The Art of Unit Testing: With Examples in C#", Roy Osherove, Manning Publications Company, 2013
5. "User Stories Applied: For Agile Software Development", Mike Cohn, Addison-Wesley Professional, 2004
6. “Examining the Agile Cost of Change Curve”, http://www.agilemodeling.com/essays/costOfChange.htm, Scott Ambler
7. “Defect-Driven Process Improvement”, http://www.neverletdown.net/2014/10/defect-driven-process-improvement.html