

Et si on arrêta d'enfoncer des clous avec un tournevis ?

Le BDD n'est pas une méthode de test



Rose Lutz

Paris Test Conf – 31/03/2026



GERAM-IT

A propos



www.linkedin.com/in/rose-lutz

- QA (2013)
- Coach Qualité (2023)
- Formations sur le BDD et Gherkin (2021)
- Ma devise :
« la Qualité c'est de l'Art »

Enfoncer des clous...

Clou / Vis : même finalité





BDD / Testing : même finalité

Qualité

Modalités différentes

- BDD :

- Construire le Produit
(Découvrir la spec)
- Avant de coder
- Devs

- Testing :

- Critiquer le Produit
(Découvrir des bugs)
- Après avoir codé
- Devs / QAs / POs / Clients / ...

« Test » == « Truc » ?

Le jour où ça a fait tilt



sebrose

08-28-2019 04:58 AM



Hi Alex,

I'm glad you got some value from the webinar - which was focussed on collaboration rather than Behaviour Driven Development (BDD).

BDD is not a QA activity, although one of the by-products of following all three BDD Practices (Discovery, Formulation, Automation) is a certain number of automated scenarios. The primary goal of BDD, however, is reaching a shared understanding of the required behaviour -- and capturing this using business-readable language.

The majority of the behaviours that we describe in BDD scenarios are quite narrow, but there is a place for "Journey Scenarios" that document wider, end-to-end processes (see our forthcoming book, [Formulation](#)). These are useful to give the reader of the documentation an overview of how users will interact with the system and are not intended to replace other forms of testing that your organisation may require.

Source : <https://community.smartbear.com/discussions/testcomplete-questions/the-secrets-of-successful-collaboration---bdd/189699>

Les QAs ont un rôle à jouer, mais ça n'est pas ça :

Scenario : Young drivers

Given a person :

Name	Birthdate	City	
Jane Doe	01/01/2004	Paris	

And a car :

Brand	Name	Registered	Engine	
VW	Beetle	2010	800cc	

And an insurance type :

Type	Collision	Usage	
Complete	Yes	Work	

When the person subscribes the insurance

Then the person should pay 495€

Source : https://www.linkedin.com/posts/alt-qa_e-pour-essential-activity-7204418885872267264-hBlo

Le BDD, petite méthode triste et incomprise

- « Nous on fait du **Business** Draiven Development »
- « Nos BDD sont cassés / flaky »
- « Comment écrire de bons BDD »
- « Oh la la, les tests BDD sont tout rouges »
- « C'est à cause de la QA qui n'a pas écrit les tests Gherkin »
- ...

« BDD », « scénarios Gherkin », « automatisations »

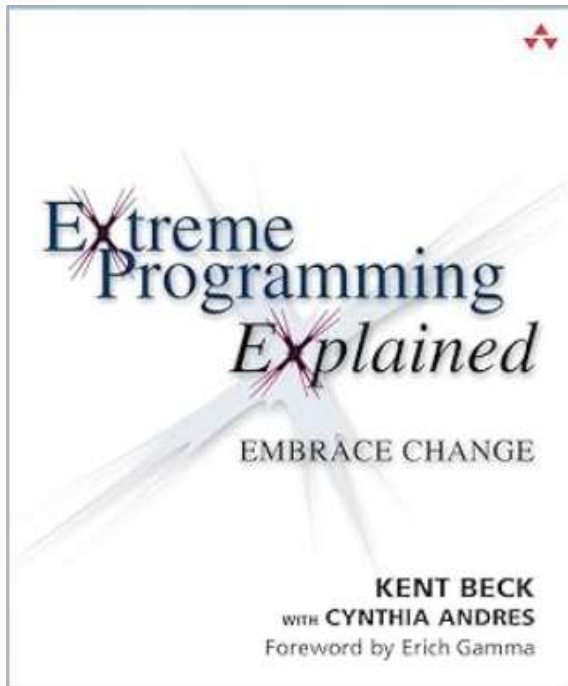
- Un triptyque...
 - Le BDD : une méthode
 - Gherkin : un langage
 - Cucumber & Co : des outils d'automatisation adaptés à Gherkin
- ... entièrement sécable
 - Le BDD peut être pratiqué sans Gherkin/Cucumber & Co

Erreurs sur les pratiques

- Faire du BDD seul dans son coin (Devs / POs / QAs)
- 3 amigos sur une spec terminée
- Scénarios Gherkin après le développement
- Automatiser les scénarios en e2e
- Mal écrire ses scénarios Gherkin
- ...

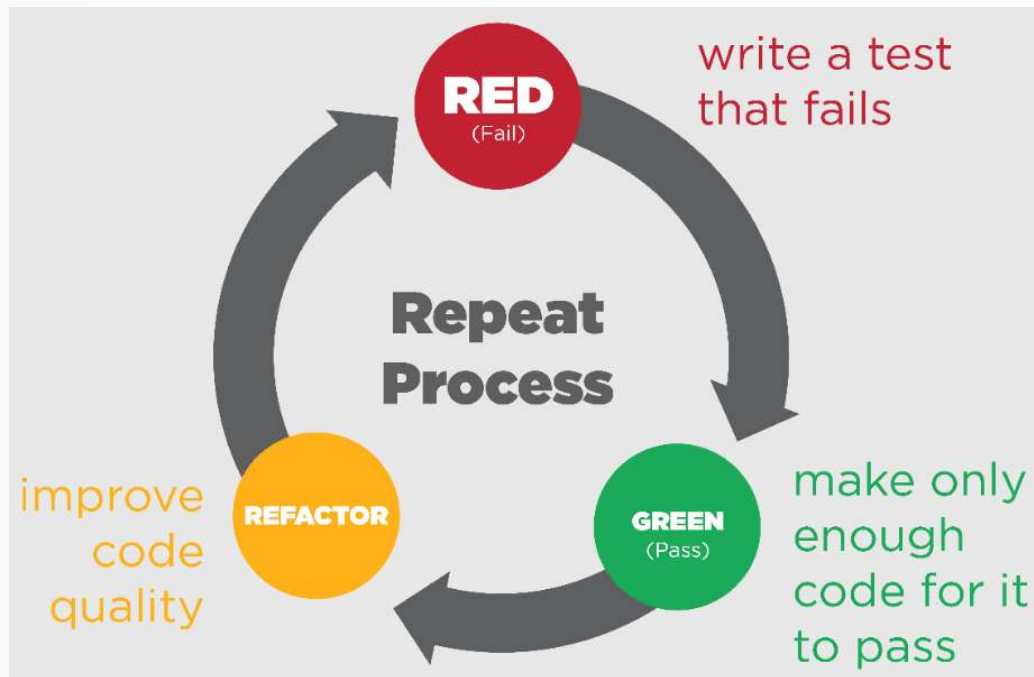
Aux origines du BDD

Le substrat de culture du BDD : l'XP



Ecrire **rapidement**
des logiciels de **haute qualité**
et savoir **s'adapter aux exigences**
changeantes des clients

Et une pratique en particulier : le TDD



- Finalité =
 - Coder propre : découvrir l'algo adapté (\neq tester)
 - Refactoriser en toute sérénité : le *détecteur de bugs* est créé en même temps que le code (*self-testing code*)
- Refactoriser = améliorer l'implémentation du code sans en altérer le *comportement*

Source : <https://www.icterra.com/tdd-is-not-about-testing-but-the-design/>

Dev



- Par quoi commencer ?
- Quoi tester / ne pas tester ?
- Combien de tests ?
- Comment nommer mes tests ?
- Comment comprendre pourquoi mon test échoue ?
- ...

Source : <https://dannorth.net/blog/introducing-bdd/>



Dan North

Focus sur ce qui apporte de la valeur au produit

Le BDD, une amélioration du TDD

« *I decided it must be possible to **present TDD in a way that gets straight to the good stuff** and avoids all the pitfalls (...): my response is BDD »*

« *If [tests] don't describe the behaviors of the system, they can cause a false sense of security. »*

Le BDD pour spécifier les comportements qui ont de la valeur

« *Remove the word 'Test' and **people suddenly think in terms of behaviors and specs** ».*

→ spécifier collectivement

→ ubiquitous langage

Un duo d'outils au service du BDD (rendre les specs exécutables)

« ***Acceptance criteria should be executable** »*

→ un template : *Gherkin*

→ un automate : *Jbehave* (et par la suite *Cucumber*, *Behat*, *Behave*, *SpecFlow/Reqnroll*, ...)

Redéfinir la qualité : a true sense of security

- D'une démarche technique....

est-ce que mon code marche ?

- ... à une démarche produit

est-ce que mon code apporte – et conserve dans le temps – la valeur métier qu'on en attend ?

L'essence du BDD

=

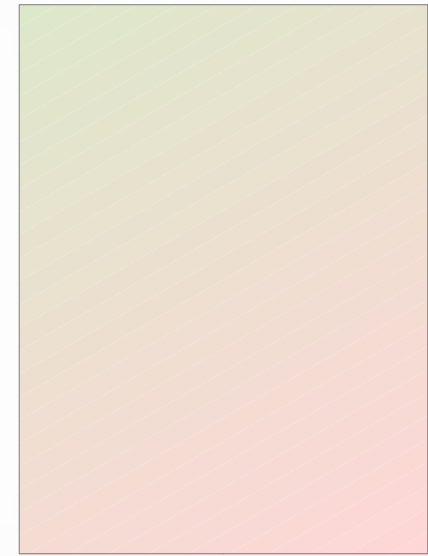
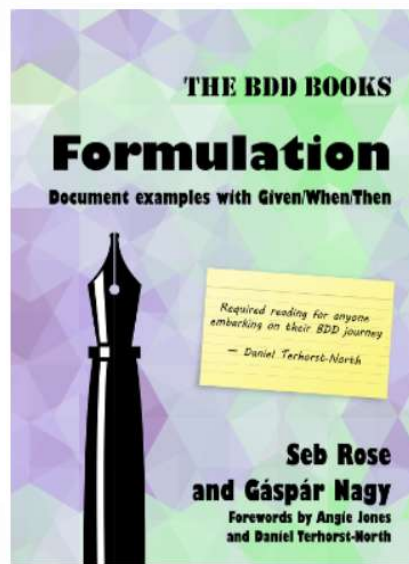
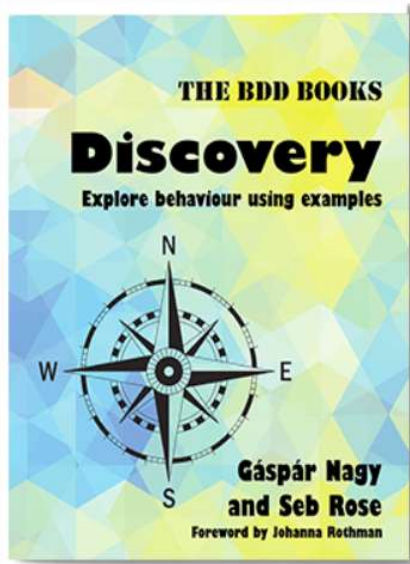
tester les comportements qui apportent de la valeur au produit
garantir

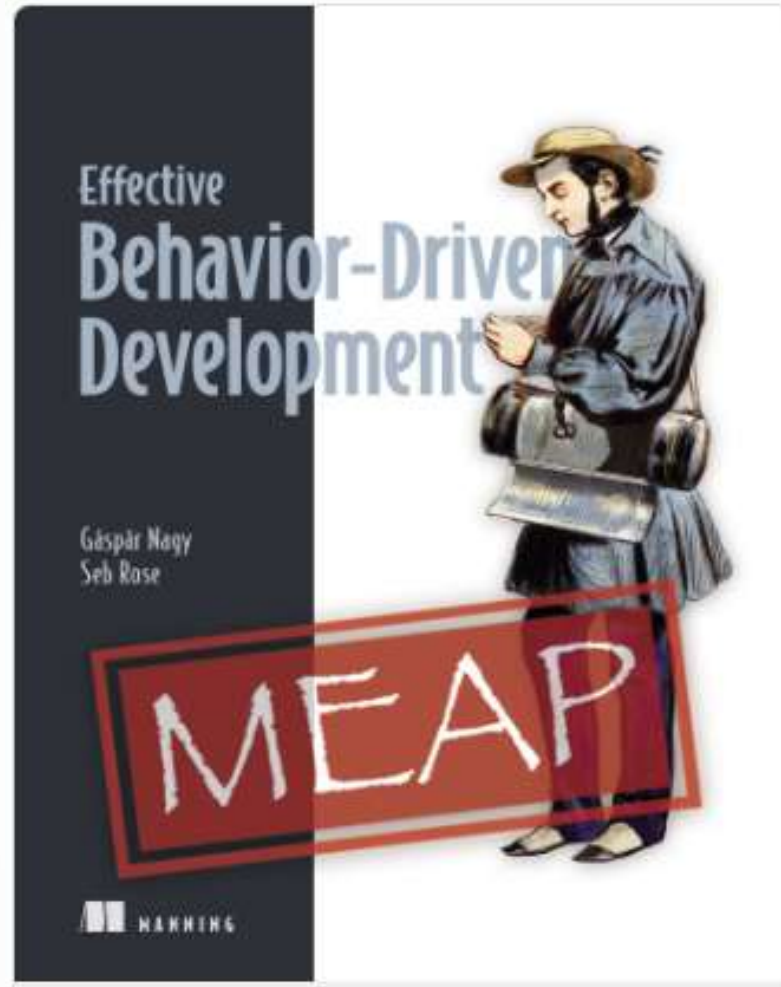
build the right product

&

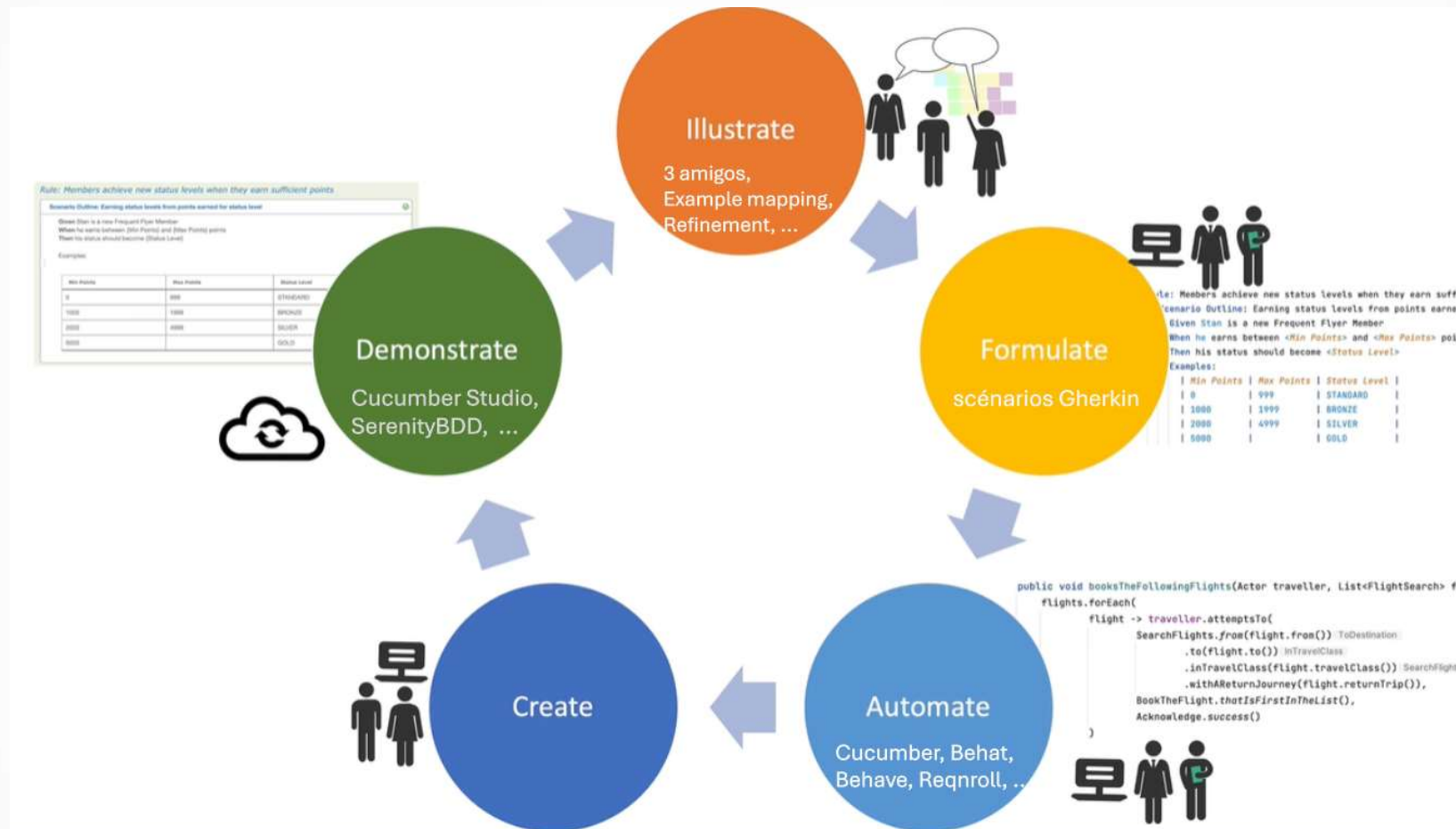
build the product right

Discover – Formulate – Automate





Le flow BDD



Source (enrichie) : BDD From Start To Finish, John Ferguson (vidéo : <https://vimeo.com/548095311>)

Discover
=
L'indispensable du BDD

Objectif : Aligner l'équipe

« It's developer (mis)understanding that's released in production, not the experts' knowledge. » (A. Brandolini)

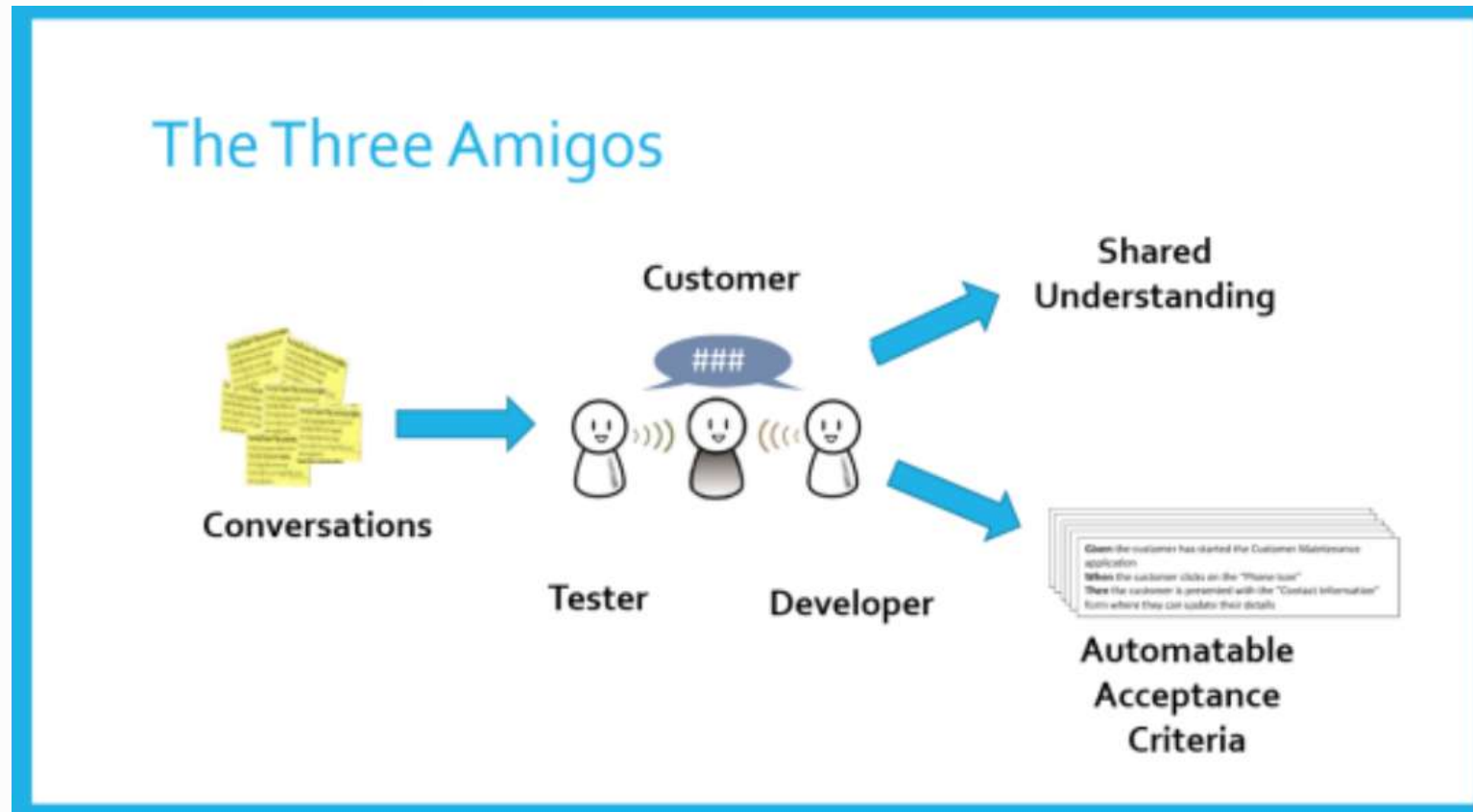
→ Créer une **compréhension commune** (*« shared understanding »*)

Side-effect : Co-créer le produit

→ intelligence collective

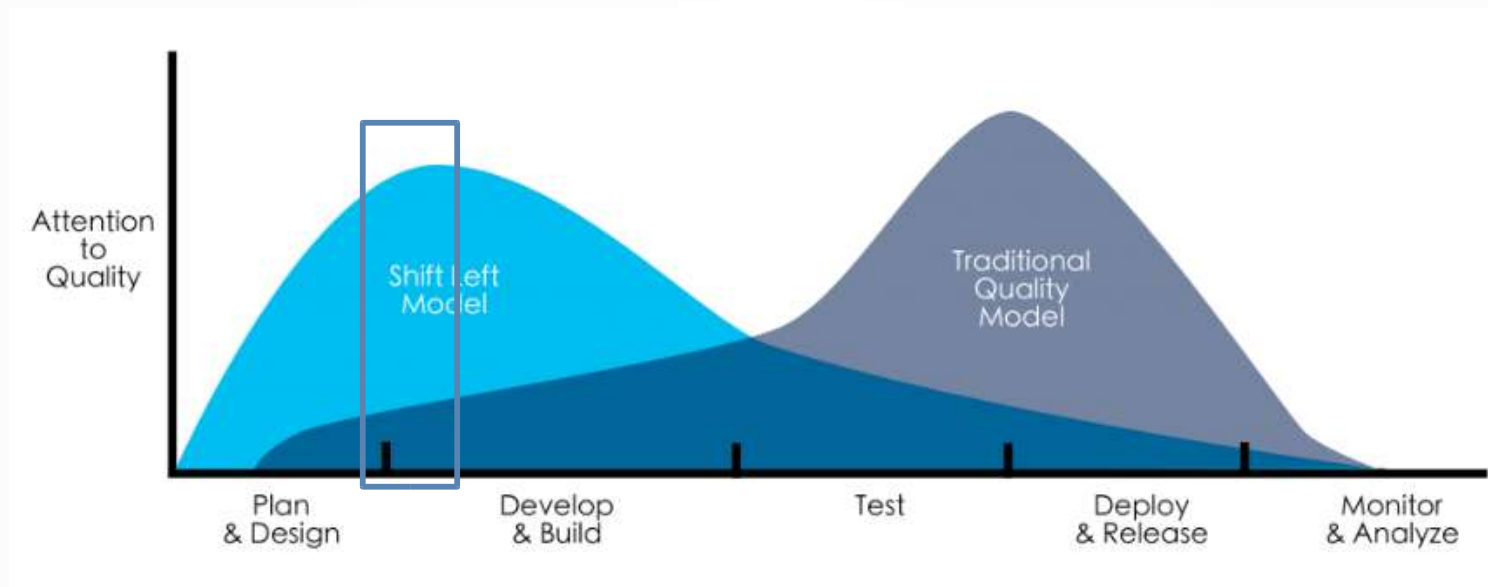
« Les devs ne sont plus de simples « traducteurs » mais des créateurs de systèmes » (D. North)

Les 3 amigos (ou +)



Source : <https://blogs.sap.com/2017/04/12/behaviour-driven-development-bdd-closing-the-loop-on-a-great-fiori-ux/>

Partie intégrante du « Shift left »



Source : <https://devopedia.org/shift-left>

Pas de BDD sans Exemples concrets



Daniel Terhorst-North

@tastapod · [Follow](#)



[#BDD](#) in a tweet: Using examples at multiple levels to create a shared understanding and surface uncertainty to deliver software that matters

10:10 PM · May 26, 2013



65

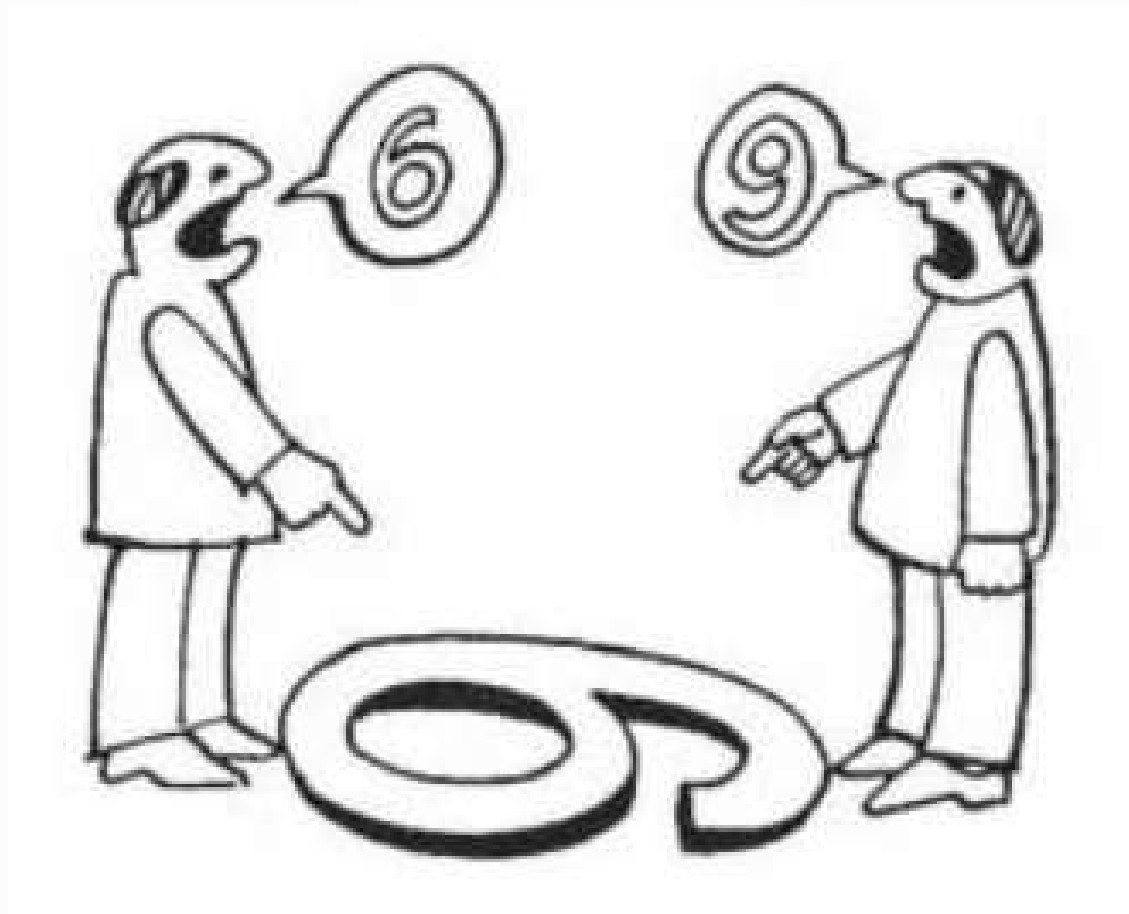


Reply



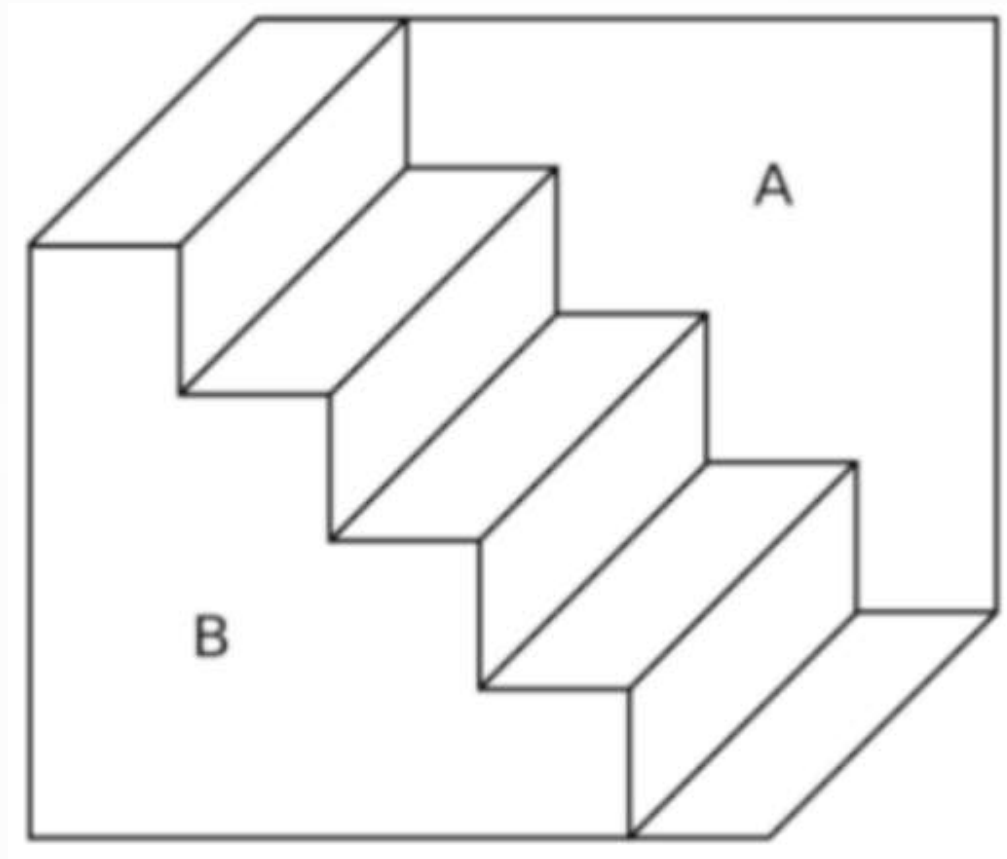
Share

[Read 2 replies](#)



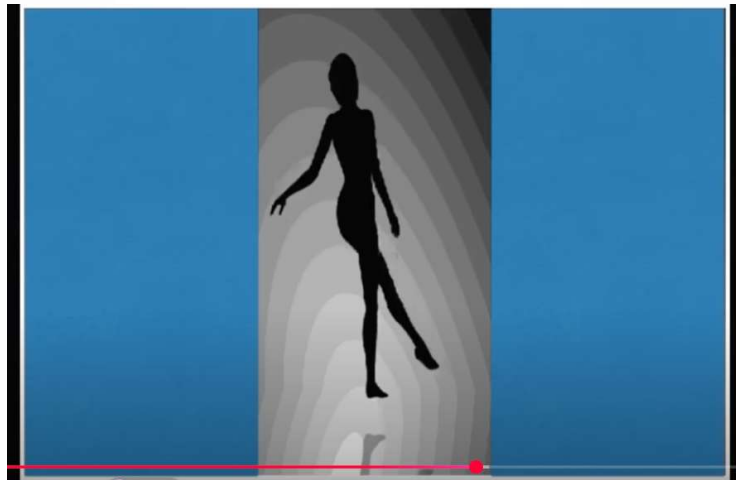
B

12
A B C
14



Albert Moukheiber : la danseuse

- TEDx « A la découverte de notre cerveau » (2017)



Source (10:47) : https://www.youtube.com/watch?v=u_soKgjGzrU

Mise en situation

- « Ajouter des frais de livraison au prix total du panier ».
- Avec les règles suivantes :
 - Les frais de livraison pour les paniers de moins de 30€ sont de 6€
 - Les frais de livraison pour les paniers de plus de 30€ sont de 3€

Des questions, des questions, toujours des questions

- Et si le panier est de 30€ pile ?
- Parle-t-on en HT ou en TTC ?
- Et si j'ai un code promo ? Est-ce que les frais de livraison sont appliqués avant ou après la réduction ?
- Attention au cas où le code promo n'est plus valide aussi !
- ...

Une illustration vaut mille mots

Imaginons,

**tu as un panier de pile 30€ TTC,
comme on applique les frais de livraison d'abord
et que pour 30€ ou moins les frais sont de 3€,
on obtient 33€,
puis on applique la réduction de 5€,
soit au final $33€ - 5€ = 28€$ à payer.**

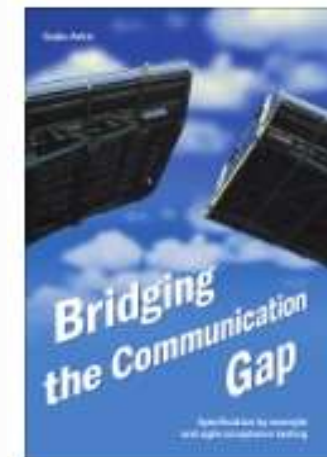
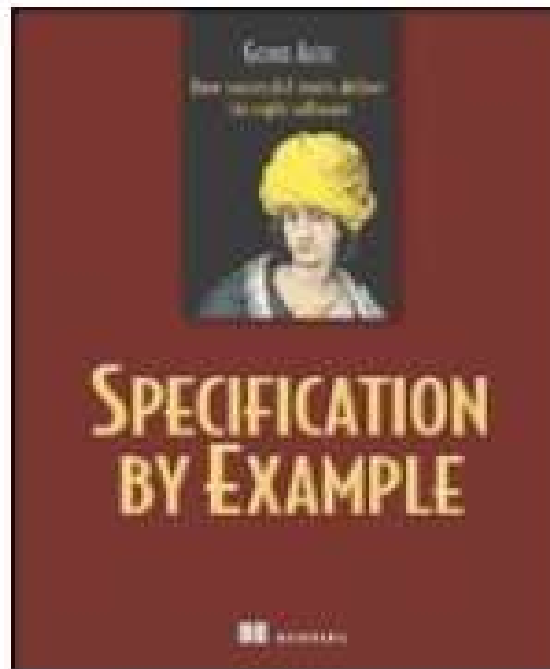
L'exemple : à la fois spec et test d'acceptation

- Illustrer et approfondir les attendus métier
- Identifier les critères d'acceptation (ce qui compte)
- Les devs savent **à l'avance** ce qu'ils doivent garantir
- Leurs développements sont guidés par ces exemples/tests d'acceptation

Je s'appelle « spec par l'exemple »

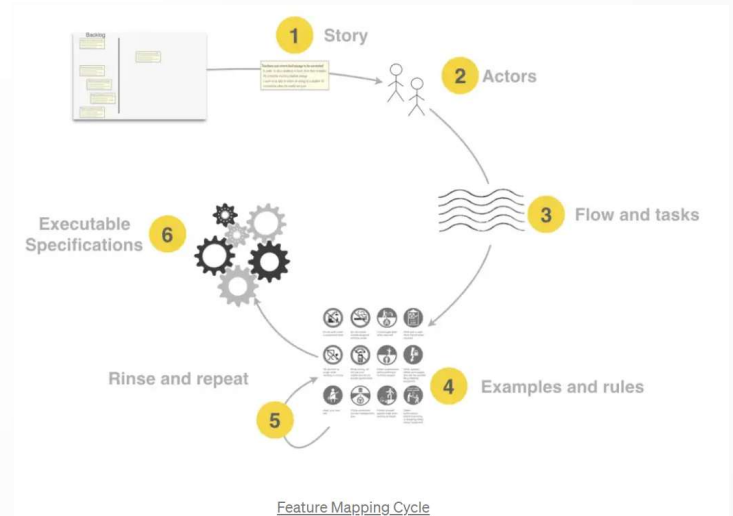
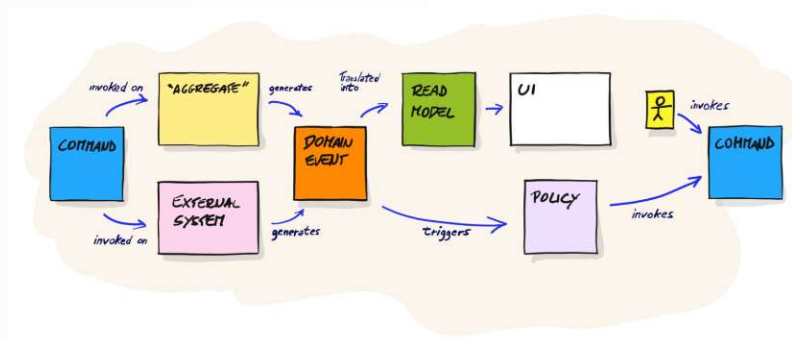
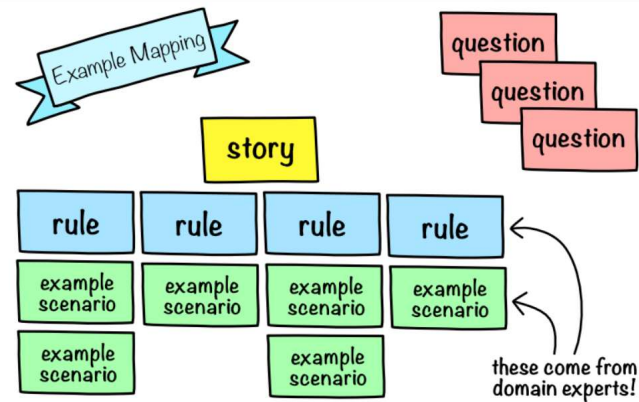
- « *BDD, or ATDD, or Spec by example, or whatever you choose to call **the thing where you write acceptance criteria, automate them and then make the application match** (...)* » (D. North, 2011)
- « *Gojko Adzic gave BDD a new and better name: **Specification by Example*** » (Aslak Hellesoy, 2014)

Sources : <https://dannorth.net/blog/whose-domain-is-it-anyway/>
<https://cucumber.io/blog/collaboration/the-worlds-most-misunderstood-collaboration-tool/>



Peu importe la forme, pourvu qu'il y ait l'alignement

- **Example Mapping,**
- Feature Mapping,
- Event Storming,
- Simple discussion...



Illustrate / En tant que QA

- NON :

- Relire des tests
- Pas de perspective utilisateur
- Ecrire déjà des scénarios Gherkin

- OUI :

- Co-crérer la spec
 - Dériskuer
 - Prioriser
- Questions
 - Situations (*Et si... ?*)
 - Techniques de conception des tests
 - ...

Formulate
=
Là où le bât blesse

Ecrire de bons scénarios Gherkin serait facile... ?!

```
Feature: Authentication
```

```
Scenario: User is login in
```

```
When the user is login in with valid credentials "hello@alexsoyes.com" and "test1234"
```

```
Then the user should have a valid uuid token
```

Non...

```
Feature: I want open the Site and verify links in Home Page
```

```
Background:
```

```
Given I navigate to the Website
```

```
Scenario: Verify content in Home Page
```

```
When I views links in Home Page of the Site
```

```
| HomePageLinks |
```

```
| Contact us |
```

```
| Sign in |
```

```
| Women |
```

```
| Popular |
```

```
| Best Seller |
```

```
| Cart |
```

Non plus...

Scenario : Add an article as favorite and add it to the shopping cart from favorites tab

Given a user has an account
And the article is not yet added to his favorites
And the user accesses his account
And the user opens the articles list

When the user adds the article "Article1" to his favorites
And the user opens his favorites tab
Then the following article is available in the favorites list:
| Article1 |

When the user adds the article "Article1" from favorites to his shopping cart
Then the article is added to shopping cart

Vraiment pas...

Scenario : Young drivers

Given a person :

Name	Birthdate	City	
Jane Doe	01/01/2004	Paris	

And a car :

Brand	Name	Registered	Engine	
VW	Beetle	2010	800cc	

And an insurance type :

Type	Collision	Usage	
Complete	Yes	Work	

When the person subscribes the insurance

Then the person should pay 495€

Toujours pas...

```
Given I sign up as "Matt"  
And my password is "password"  
And my password confirmation is "password"  
And I have deposited "$60" in my account  
And I have deposited "$40" in my account  
When I check my bank balance  
Then my bank balance is "$100"
```

Le mémo : BRIEF

B	Business language	Business terminology aids cross-discipline collaboration
R	Real data	Using actual data helps reveal assumptions and edge cases
I	Intention revealing	Describe the desired outcomes, rather than the mechanism of how they are achieved
E	Essential	Omit any information that doesn't directly illustrate behaviour
F	Focused	Each scenario should only illustrate a single rule
	Brief	Shorter scenarios are easier to read, understand and maintain

Source (Seb Rose) : [https://cucumber.io/blog/bdd/keep-your-scenarios-brief-\(1\)/](https://cucumber.io/blog/bdd/keep-your-scenarios-brief-(1)/)

Compléments

- **Origine/Inspiration : Arrange, Act, Assert**
 - Contexte initial (Given), Action (When), Attendu (Then)
- **Doit pouvoir être automatisé bas niveau**
 - focus sur la logique métier (le quoi, pas le comment)
- **Deviendra de la Documentation**
 - lisible, court, essentiel
 - est-ce qu'on apprend qqch ?

Jamais une seule solution : collaborer

Scenario : les frais de livraison devraient être appliqués avant un code promo

Etant donné un panier dont le prix total est de 30€ TTC

Et qu'on a appliqué une réduction de 5€ encore valide

Lorsqu'on calcule le prix final du panier

Alors les frais de livraison devraient être de 3€

Et le prix final du panier devrait être de 28€

Scenario : le cas où il y a un code promo à appliquer

Etant donné que mon panier est de 30€ TTC

Lorsque j'applique une réduction de 5€ encore valide

Alors les frais de livraison devraient être de 3€

Et le prix final du panier devrait être de 28€

Formulate / En tant que QA

- NON :

- Scénarios Gherkin écrits après avoir codé
- Par les QAs, pour les QAs
- Scénario Gherkin \neq Test step by step
- Focus sur le *Comment*

- OUI :

- Scénarios Gherkin écrits avant de coder
- Par l'équipe, pour les Devs et les POs/clients
- Exemples \rightarrow scénarios Gherkin
- Focus sur le *Quoi*
- Ne doivent pas changer tant que la spec ne change pas
- Documentation utile

Automate

Le test n'est pas la raison première

Automatiser : plusieurs finalités

- Avant le déploiement : guider les développements
 - **Specs** exécutables
- Après le déploiement : garantir la valeur dans le temps
 - Assurer la *non-régression*
 - Documentation *vivante*

Guider les développements

- Découvrir l'algo / Coder juste ce qu'il faut
- Feedback loop immédiate
- Refactoriser sereinement (*Self-testing code*)

retrieve-next-showings.feature

```
1 Feature: As a cinephile In order to never miss a minute of a movie I want to know what are the next showings of the movie
2
3 Background:
4   Given a cinephile
5
6 Scenario: Looking for the next showings before the start of the last one
7   Given the last showing is not started yet
8   When the user asks for the next showings
9   Then the returned showings are not started yet
10  And the returned showings are not starting in the next 15 minutes
```

RetrieveNextShowingsSteps.java

```
1 public class StepDefs implements En {
2
3     public StepDefs() {
4
5         Given("^a cinephile$", () -> me = new Cinephile("me"));
6         Given("^the last showing is not started yet$",
7             () -> lastShowing = new Showing(now().plusHours(3)));
8
9         When("^the user asks for the next showings$",
10            () -> nextShowings = searchShowings.searchNextShowingsFor(me));
11
12        Then("^the returned showings are not started yet$", () -> {
13            assertThat(allShowings.stream().filter(showing -> !showing.hasStarted())
14                .containsAnyElementsOf(nextShowings));
15        });
16
17        Then("^the returned showings are not starting in the next 15 minutes$", () -> {
18            assertThat(nextShowings)
19                .noneMatch( showing -> showing.startsBefore(now().plusMinutes(15)));
20        });
21    }
22 }
```

Garantir la valeur dans le temps

- Non-régression
- Documentation *vivante*
- Feedback loop immédiate/rapide

Automate / En tant que QA

- NON :

- Les QAs écrivent les step definitions
- Les scénarios Gherkin sont figés et ne peuvent plus être retravaillés

- OUI :

- Les QA n'ont aucun rôle d'automatisation...
- ...Ou un peu (pair / haut-niveau)
- Collaboration pour la réutilisation des steps

Perspectives

« Tests » == « Truc » ?

- Faire la différence entre « Test - spec » et « Test - critique » ?
 - Point de bascule : anticipé ou pas
 - « attendu » vs « inattendu »
- « Tests » : mot valise, trop vaste

Etes-vous QAs ou Testeur.se.s ?

- Acteurs de la Qualité :
 - Challenger les specs (*larvae hunting*)
 - Faciliter l'alignement et la communication
 - Lever les alertes
 - Confronter / Critiquer le résultat produit (bug hunting)
 - 100 % Exploratoire (!?)
- Responsabilité de chacun, à chaque étape, avec des objectifs différents (besoin utilisateur, règles métier, UI/UX, sécurité, performance, ...)

L'alignement, un sujet d'Actualité

Merci
pour votre attention !

Contact & Feedbacks



www.linkedin.com/in/rose-lutz



rose.lutz@alt-qa.com