

User Story :

L'histoire utilisateur au cœur

du développement logiciel agile

(partie 2 : Raconter une belle histoire)

(D)écrire l'histoire

**Raconter une belle histoire et
bien comprendre le récit à l'aide
de conditions d'acceptation
et d'exemples**

Jeu



**Et si on jouait un peu
avant de (d)écrire la story ?**



Illustration du jeu :
Des spécifications
ouvertes avant tout ...

Négociable ... et négociée



« ne formuler dans un premier temps que l'essentiel »

« Une bonne story capture l'essence, non les détails. » ⇒ description courte
« les détails seront plutôt co-créés » ⇒ détails négociés lors des discussions

« la carte pourra se voir ajouter des *notes, des idées de tests, ...*
mais nous n'avons pas besoin de ceci pour prioriser ou planifier les stories. »

**L'important est d'amener
de la valeur métier
au travers de spécifications
ouvertes ...**

Bonne Pratique : Formalisme recommandé pour la *description* d'une story (Card)



So that (Afin de)
(bénéfice apporté à la partie prenante)
peut être optionnel,
mais reste une bonne aide
→ pour la décomposition
→ pour la priorisation



Template, appelé aussi matrice rôle-fonctionnalité, proposée par Rachel Davies et Tim McKinnon en 2001

Exemples de description avec le pattern « En tant que ... Je veux ... Afin de ... »

BigMoneyJobs

En tant que recruteur

Je veux effectuer le paiement
d'une offre d'emploi
avec une carte de crédit

Afin de gagner du temps.



Peetic

En tant que membre de Peetic

Je veux demander un conseil
à un éleveur

Afin de savoir quoi faire quand
mon chien aboie tout le temps

En tant que recruteur

Je veux qualifier la liste
des candidats ayant répondu
à une offre

Afin de réduire et de trouver
facilement les profils
correspondants à différents critères

En tant que maître de chien

Je veux pouvoir inscrire mon
chien à une réunion
de confirmation

Afin de soigner son pedigree

Zoom sur la confirmation canine

Pratique

- > En savoir plus sur ...
- > Formulaire d'examen de confirmation
- > L'âge minimum pour la confirmation
- > Ce que dit la loi
- > Contacter le service confirmation



L'EXAMEN DE CONFIRMATION

Au cours de l'examen, le juge vérifie que votre chien ne possède pas de défauts éliminatoires (morphologie, couleur, caractère ...) ne permettant pas la confirmation.

Pour donner à votre compagnon les meilleures chances d'être confirmé , il est nécessaire de le préparer à se présenter devant le juge, à l'arrêt (en statique) puis en mouvement (aux allures).

Le juge doit vérifier l'identification de votre chien ; en cas de tatouage, pensez à vérifier que celui-ci est parfaitement lisible.

La confirmation peut être ajournée si le juge considère que le chien peut évoluer dans un sens favorable, par exemple si le juge n'a pu se déterminer sur un défaut susceptible d'évoluer.

Si la confirmation est ajournée, elle est assortie d'un délai pour représenter le chien ; attendez ce délai pour présenter à nouveau votre chien.

Extrait : <http://www.scc.asso.fr/En-savoir-plus-sur,184>

**Remarque** : les organisateurs de l'événement de la confirmation disposent d'une salle qui ne peut accueillir qu'un certain nombre de chiens.
Ils ont donc souhaité mettre en place sous Peetic une **inscription qui va permettre de limiter le nombre d'inscrits**. Pas de liste d'attente.

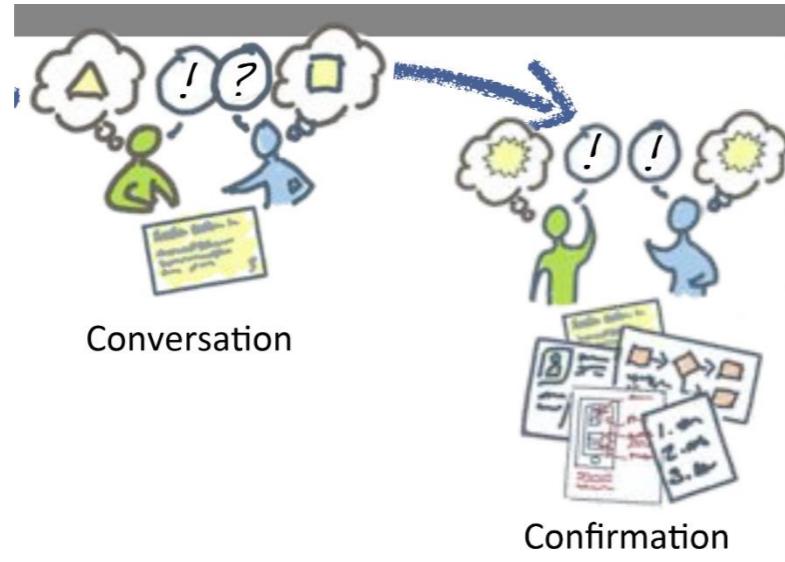
Pour bien **INVESTir**, il faut savoir ...

Rappel



Décomposer une histoire
en petites histoires
(**splitting**)

Ajouter des conditions
de satisfaction
(critères et tests d'acceptances)



Ajouter des conditions d'acceptation

*pour garder une trace du récit de l'histoire,
notamment des détails qui permettront d'« accepter »
la story en fin de développement*

Identifier le(s) condition(s) d'acceptation

BigJobMoneys

Etant que recruteur

Je veux effectuer le paiement
d'une offre d'emploi
avec une carte de crédit

Afin de gagner du temps

Note : carte Visa uniquement pour
cette story

Ce que l'on va montrer lors de
la démonstration de la story

Comment tester la story ?

- ✓ Essayer avec une carte Visa (pass)
- ✓ Essayer avec une carte vitale (fail)
- ✓ Essayer avec un bon numéro de carte,
un mauvais et un numéro manquant
- ✓ Essayer avec des cartes expirées
- ✓ Essayer avec des différents montants
(y compris un montant qui dépasse le
montant limite permis par la carte)

Identifier le(s) condition(s) d'acceptation



Peetic

Etant que maître de chien

**Je veux pouvoir inscrire mon chien
à une réunion de confirmation**

Afin de soigner son pedigree

*appelées aussi
Critères d'acceptation*

Conditions d'acceptation :

✓ ***Inscription acceptée :***

l'inscription d'un chien à une confirmation est validée (*succès*)

✓ ***Inscription refusée :***

l'inscription d'un chien à une confirmation est refusée,
car la salle a atteint sa limite (*échec*)

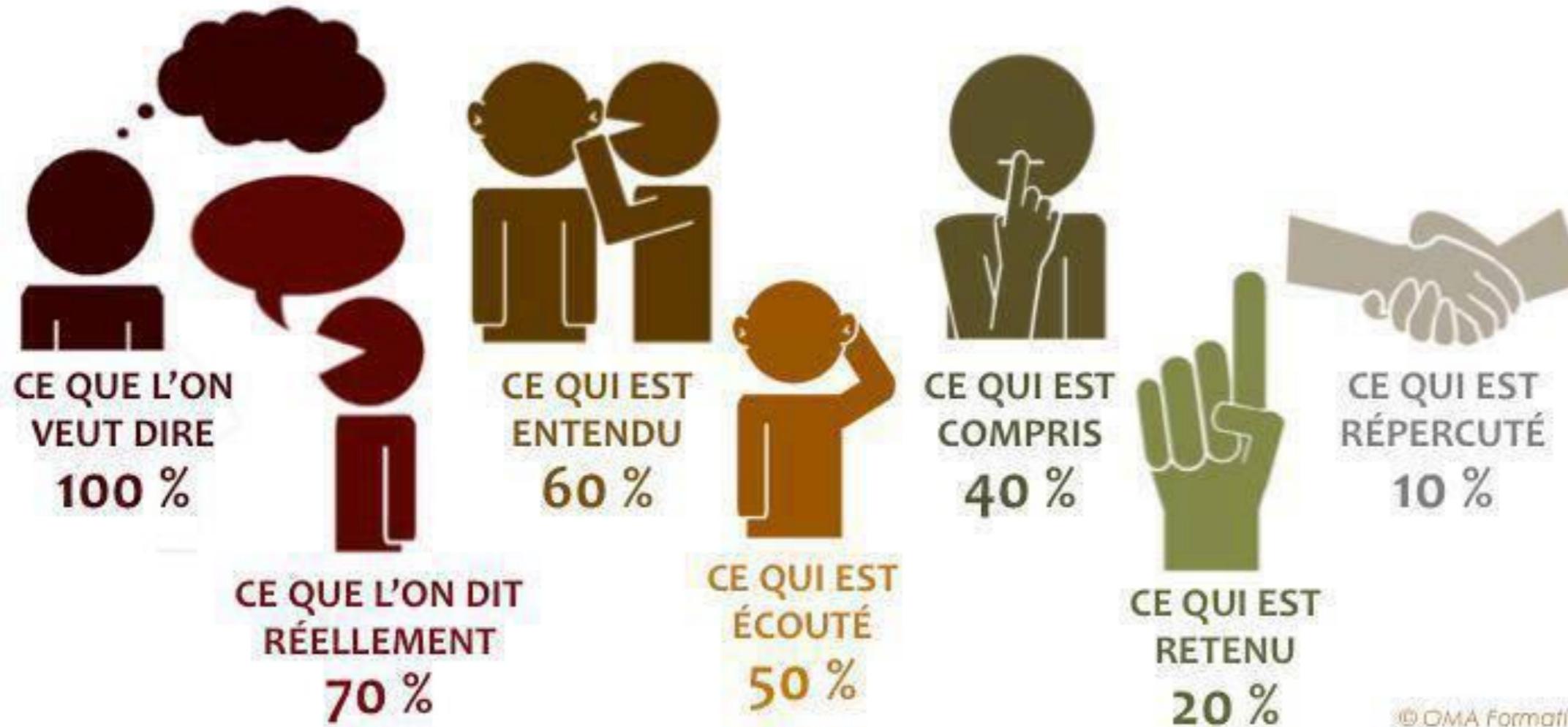
Remarque : Trop de conditions d'acceptation peut parfois amener à s'interroger sur la nécessité d'un nouveau découpage (INVEST)

Jeu

Une âme d'artiste ? ...



Et parce qu'on sait bien que la communication avec le client et la compréhension des besoins du client n'est pas toujours parfaite ...

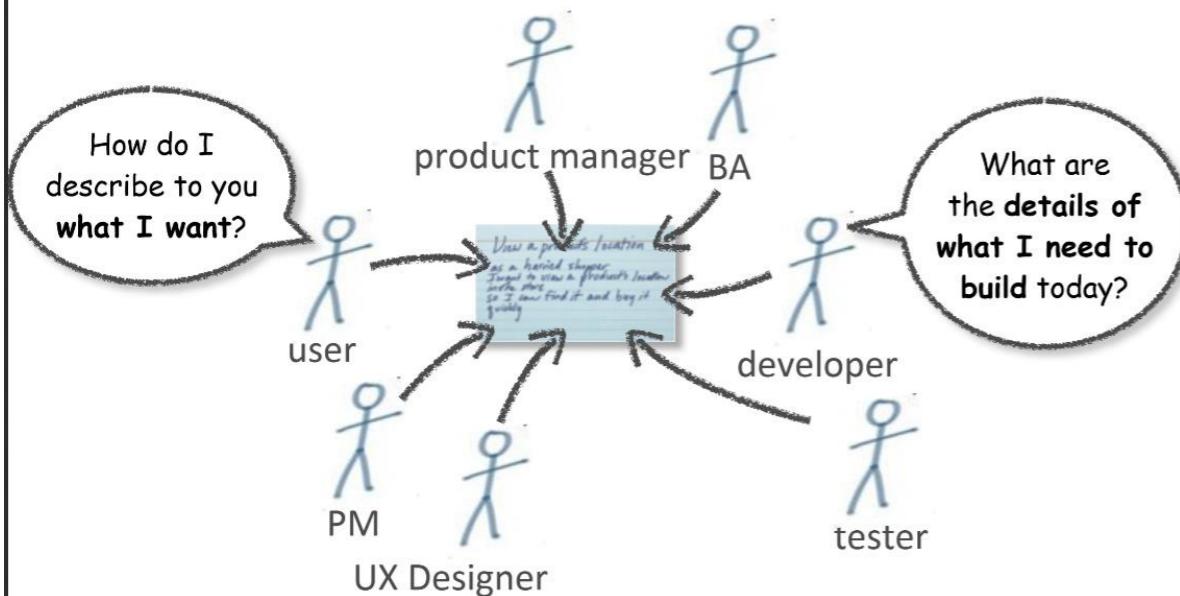


... la rédaction des stories se doit d'être un travail collaboratif ...

« Les individus et leurs interactions »

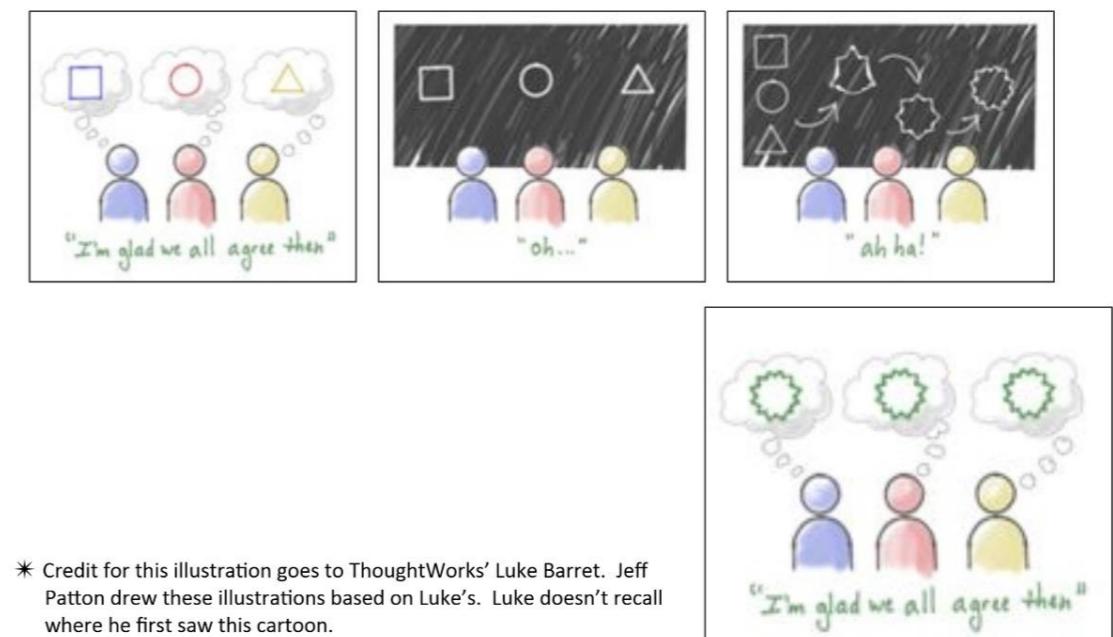
La rédaction d'une story est une responsabilité collective ...

Stories need to support lots of conversations across lots of project roles



... où tout le monde doit se comprendre et s'exprimer.

Shared understanding and alignment are the objectives of collaborative work



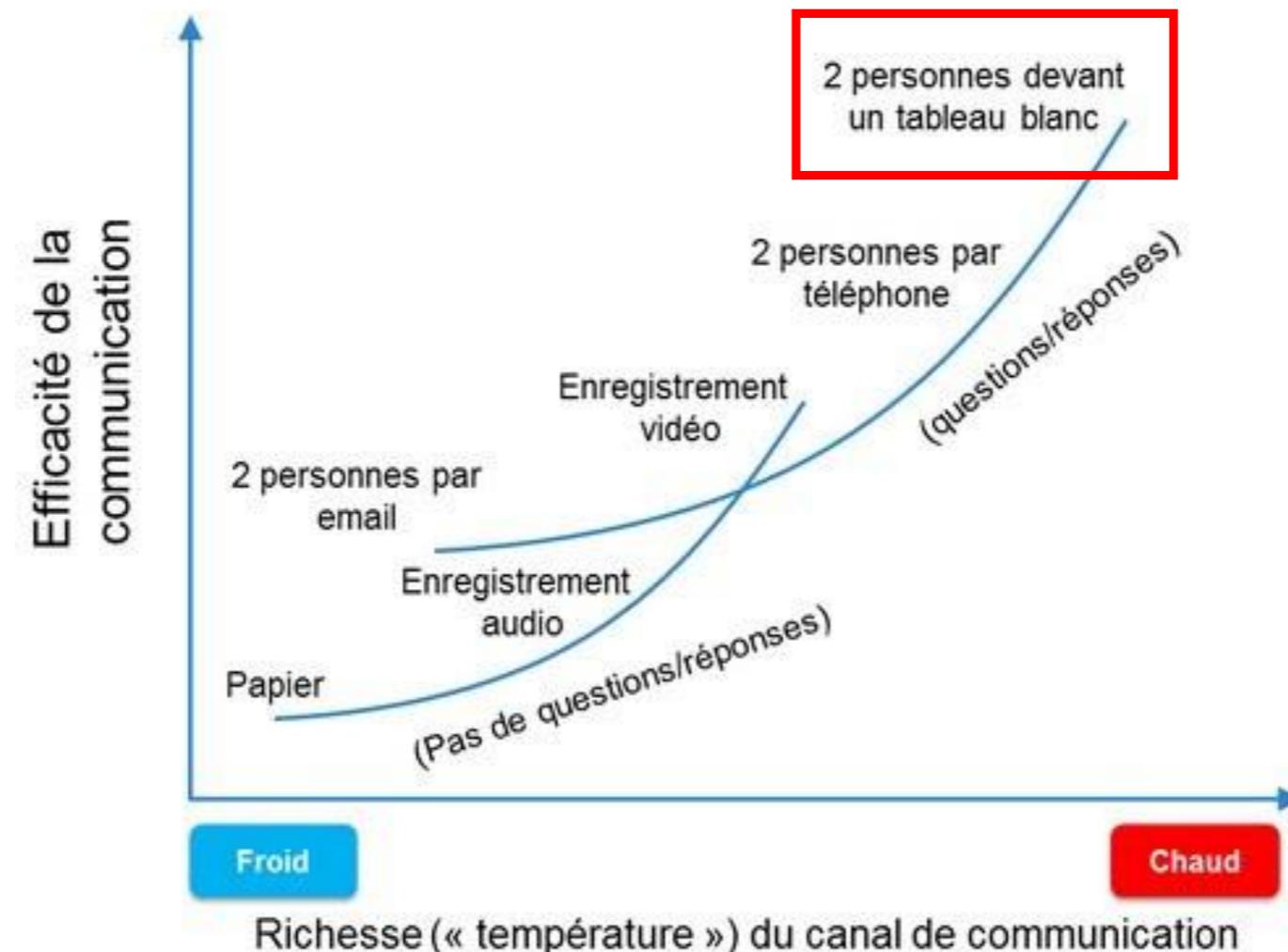
* Credit for this illustration goes to ThoughtWorks' Luke Barret. Jeff Patton drew these illustrations based on Luke's. Luke doesn't recall where he first saw this cartoon.

comakers www.comakewith.us :: youshould@comakewith.us

(cc) BY-NC-SA

... qui doit favoriser l'interaction orale en mode présentiel

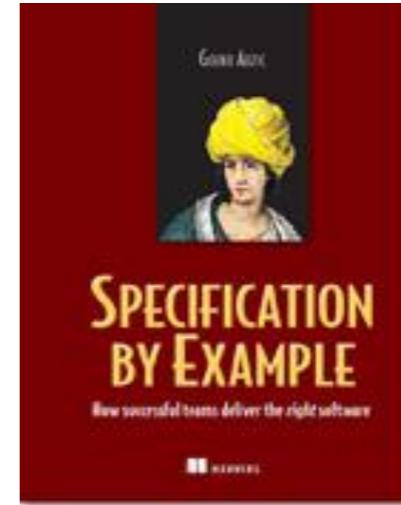
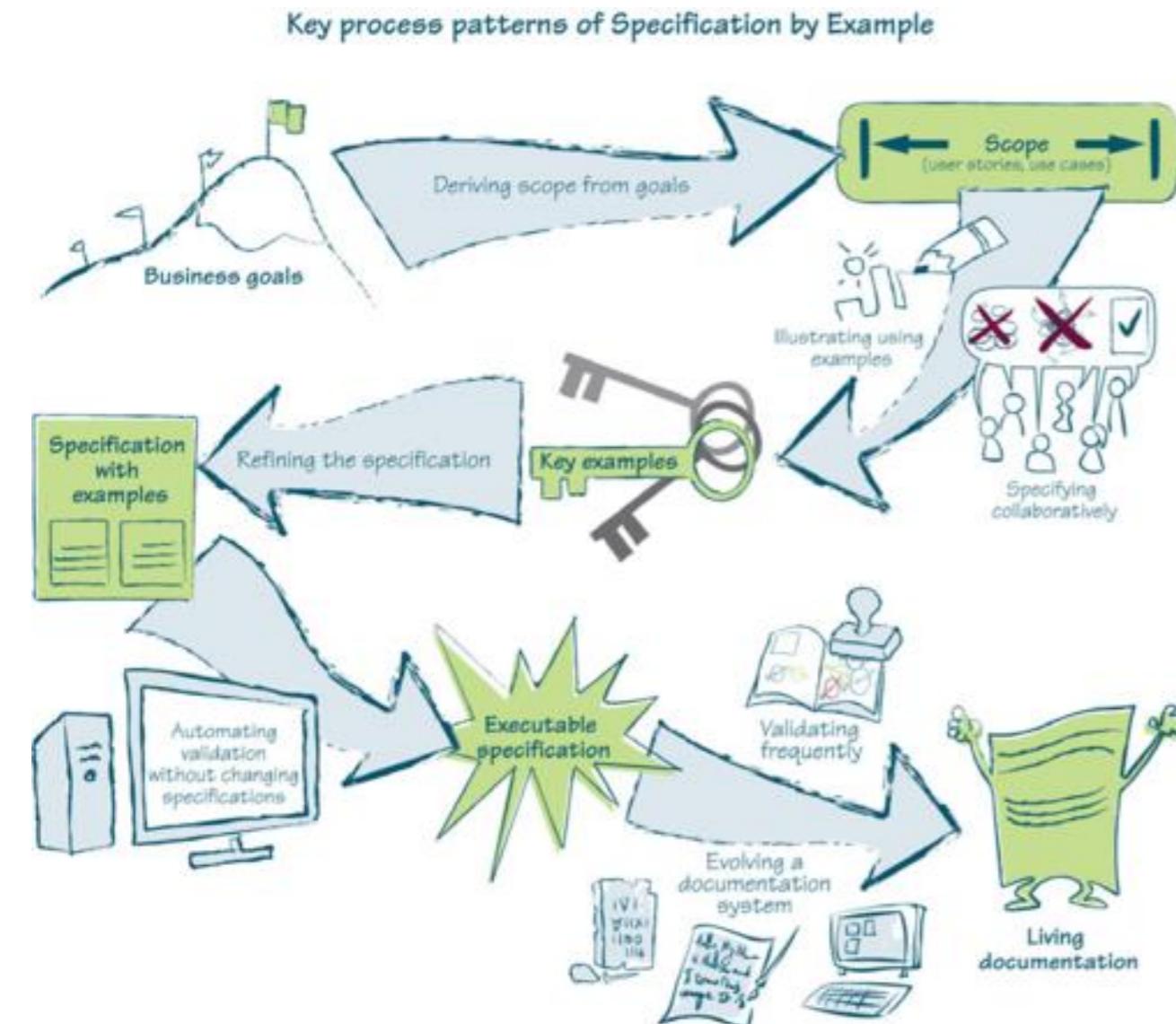
« Les individus et leurs interactions plus que les processus et les outils. »



*Comparaison de l'efficacité des moyens de communication
(Source : Alistair Cockburn)*

Isabelle BLASQUEZ - 2016

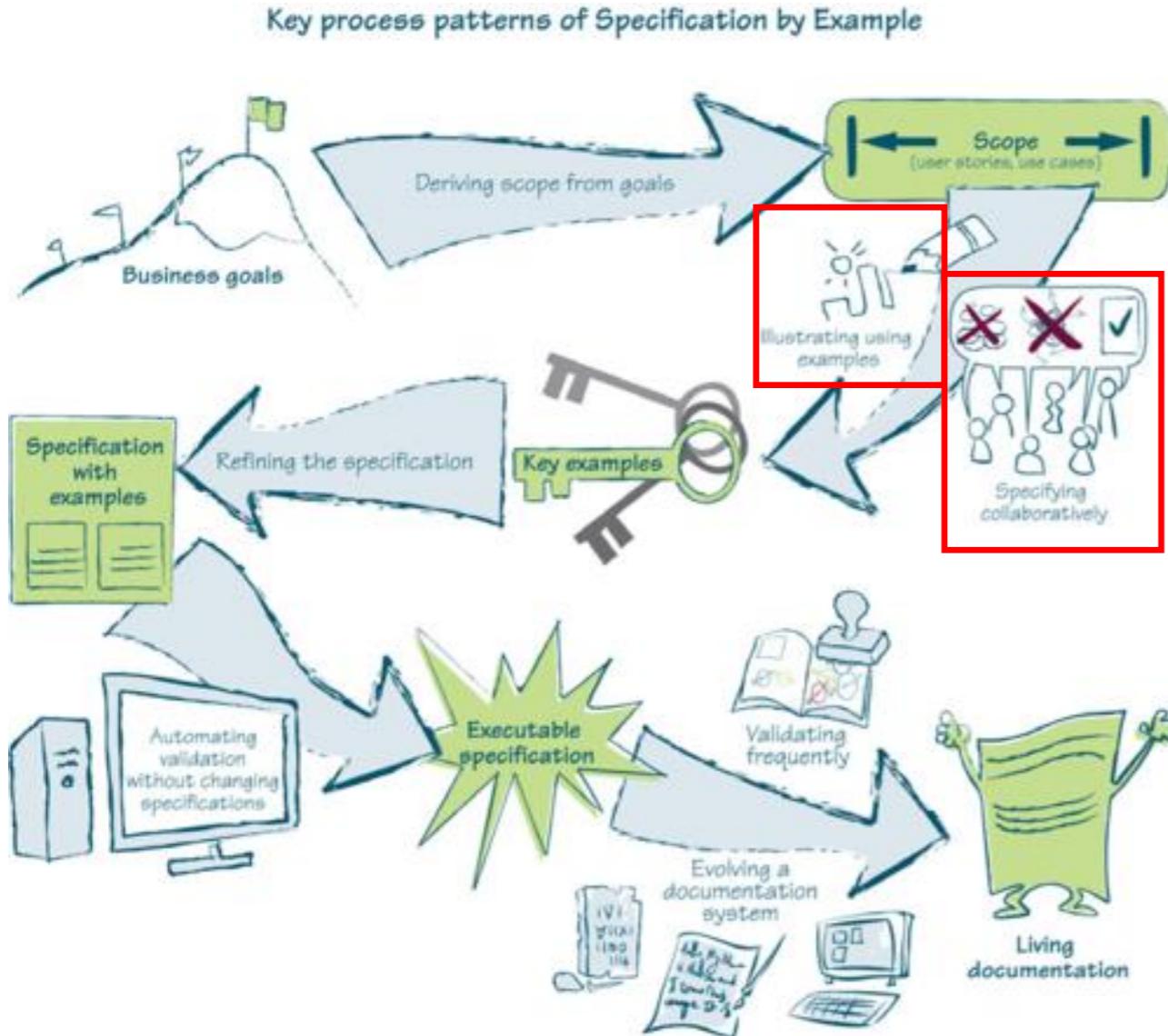
Situons l'évolution de la story dans les patterns de la Spécification par l'exemple



Cheminement au travers d'un ensemble de 7 patterns, qui permet de s'assurer que le « bon » produit (right product) sera effectivement livré.

Specification by example (SBE) is a collaborative approach to defining **requirements** and **business-oriented functional tests** for software products based on capturing and illustrating requirements using **realistic examples** instead of abstract statements.

Nous venons de montrer comment décrire une story ... ce qui correspond dans une approche de spécification par l'exemple aux patterns suivants ...



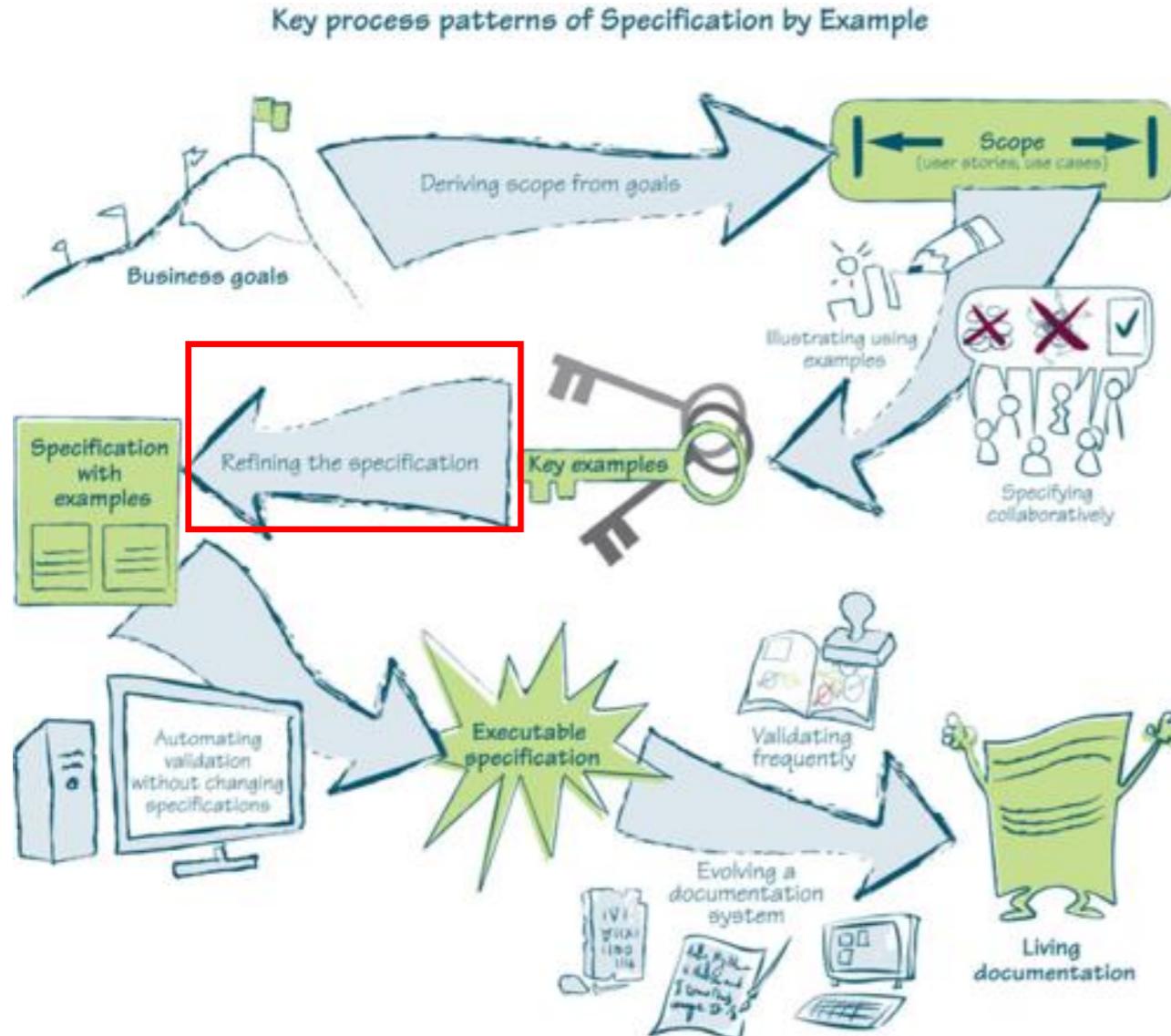
Specifying collaboratively

Compréhension commune. La Collaboration permet aux équipes de produire des spécifications qui sont faciles à comprendre.

Illustrating requirements using example

Exemples précis, complets, réalistes

Affinons maintenant la story ...



Refining specifications

Création d'un contexte concret et précis pour le développement et le test

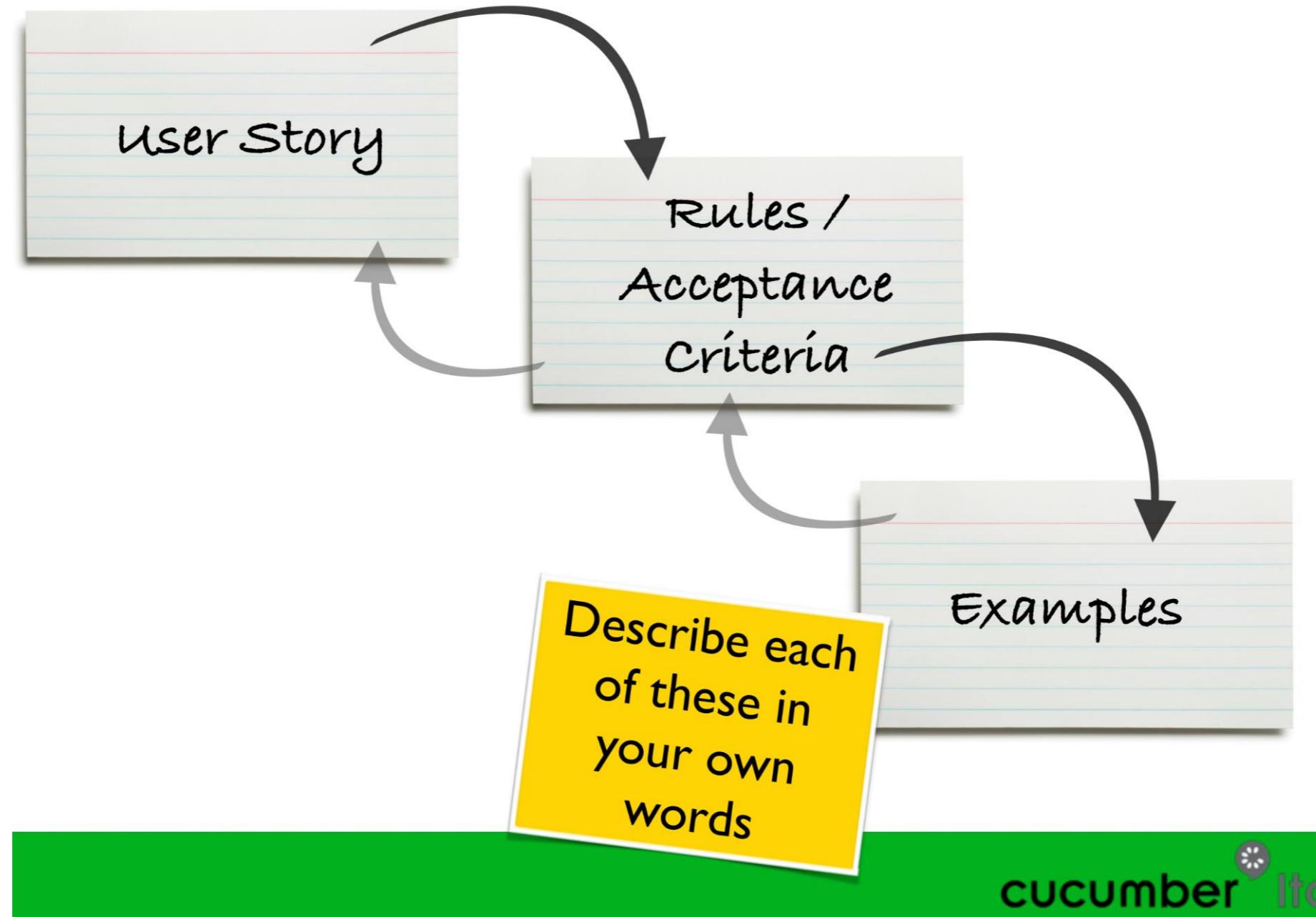
Jeu

Pourquoi affiner une story ?



Nous avons besoin
de règles ET d'exemples !!!!

Utiliser des exemples
Permet d'illustrer les règles



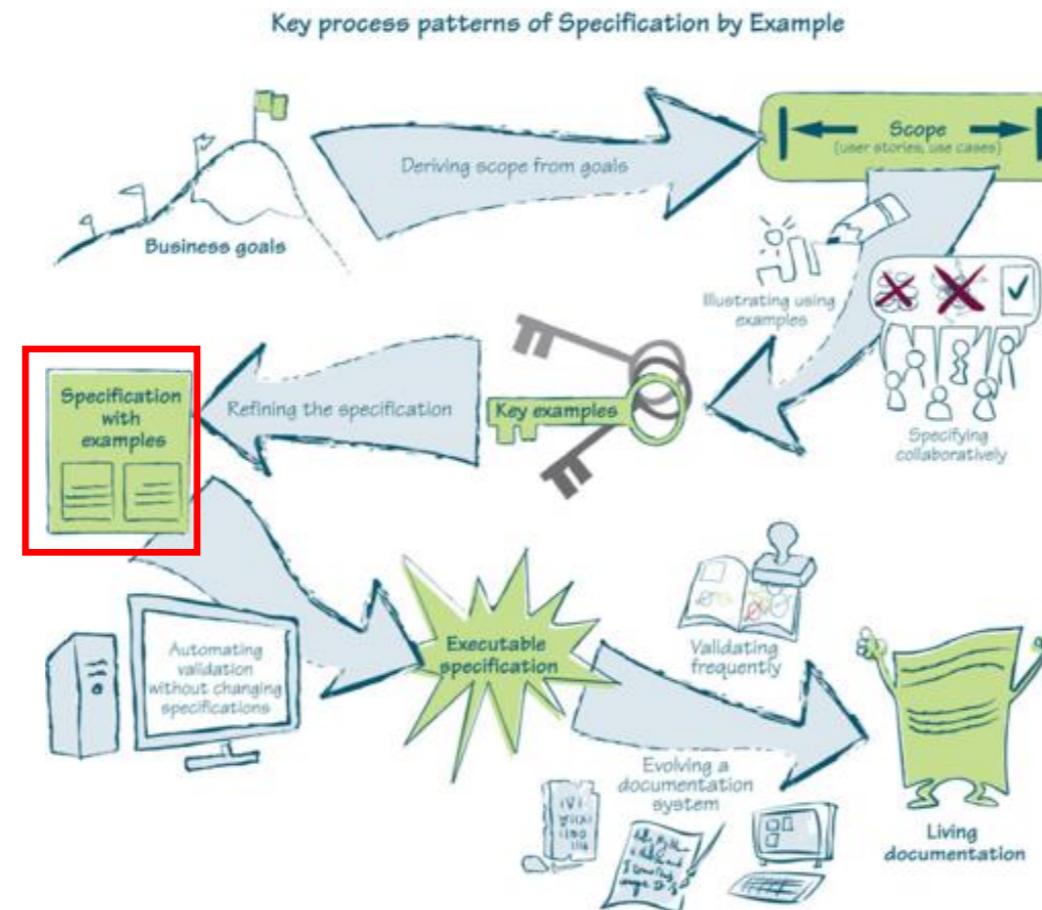
Conditions (critères) d'Acceptation

Et

Tests d'Acceptation

Vers une spécification à base d'exemples ...

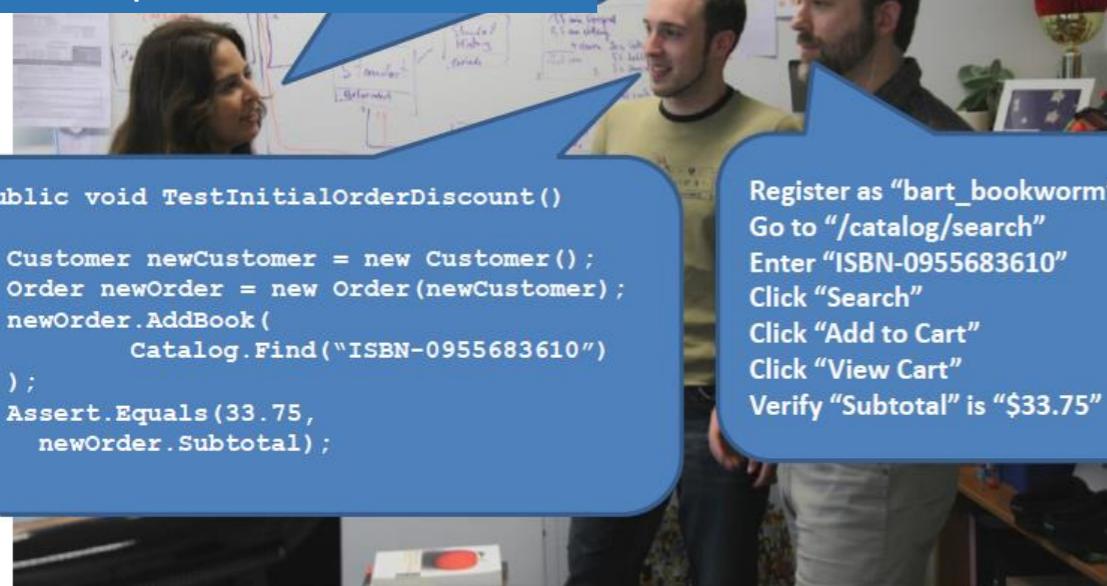
Avec de vraies valeurs, les scénarios abstraits décrivant le comportement de la condition d'acceptation deviennent des **exemples** concrets.



Ces **exemples**, décrits par des scénarios de tests,
ne sont autre que de **tests d'acceptation**.

En image : De la condition d'acceptation au test d'acceptation

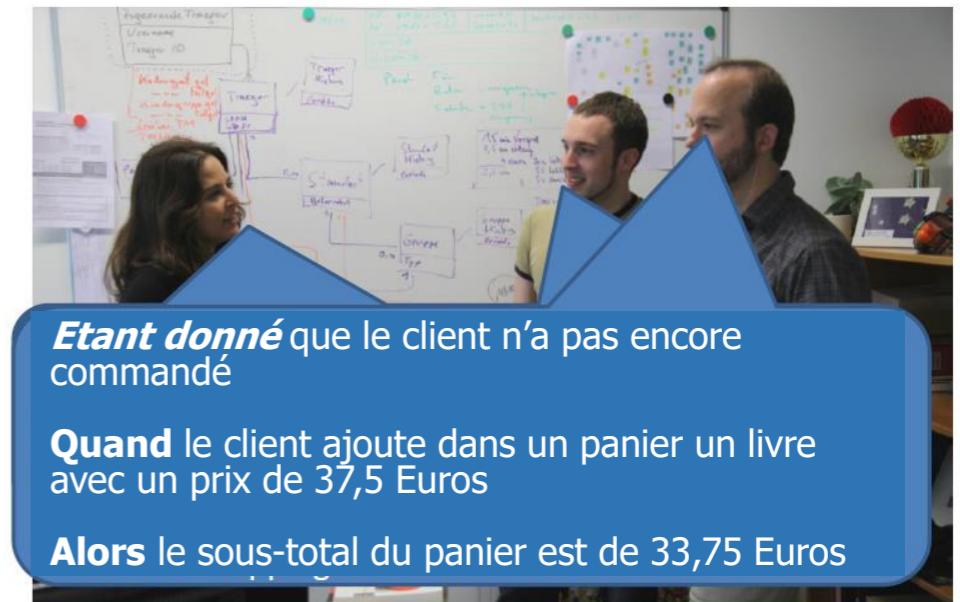
On désire encourager les nouveaux clients à réaliser un achat en leur offrant 10% sur leur premier achat



L'expression des besoins se fait en « **langage naturel** » (*Ubiquitous Language*)

La grammaire **Etant donné/Quand/Alors** (*Gherkin*) peut permettre de structurer le scénario

L'écriture des scénarios se fait en présentiel de **manière collaborative** (atelier des 3 Amigos)



Affiner les détails de la story en structurant son comportement

Dan North propose au travers du **BDD** (Behavior Driven Development) de **décrire le comportement d'une fonctionnalité** en le **structurant** avec une machine à

Etat initial avant exécution
(précondition ou contexte)

Evénement qui déclenche l'exécution

Etat après l'exécution
(postcondition ou résultat attendu)

Il propose alors le formalisme ***Given-When-Then***, appelé aussi « **gherkin** » pour exprimer ce comportement dans un langage de spécification «naturel» compréhensible de tous

Etant donné le contexte *et* la suite du contexte

Quand un événement survient

Alors on obtient un résultat *et* éventuellement un autre

On parle alors de **scénario** (de test).

Détailler une condition d'acceptation par son **comportement** dans un « *langage naturel* »



En tant que maître de chien

Je veux pouvoir inscrire mon chien
à une réunion de confirmation

Afin de soigner son pedigree

Inscription acceptée

Etant donné un maître de chien de race **et** un événement de confirmation prévu pour cette race

Condition
d'acceptation
au format BDD

Quand le maître de chien inscrit un chien d'un âge autorisé à une confirmation

Alors l'inscription est acceptée **et** le maître de chien est informé de l'inscription **et** le nombre d'inscrits **est** incrémenté de 1

Contexte

Exécution
de la story

Résultats
attendus

Passer au **test d'acceptation** en illustrant une condition d'acceptation par un exemple **concret**

Pattern AAA

Arrange

Act

Assert

Inscription acceptée :

Etant donné Corinne propriétaire de Corsaire
et une confirmation pour la race d'Epagneul annoncée pour le 15 Décembre avec 23 inscrits



Corinne et "corsaire" son épagneul



Peetic

Quand Corinne inscrit son épagneul Corsaire de 2 ans à la confirmation du 15 décembre

Alors l'inscription de Corsaire est acceptée
et le message « *Vous êtes bien inscrit à la confirmation Epagneul du 15 Décembre* » est envoyé à Corinne
et le nombre d'inscrits passe à 24.



On parle alors de **test d'acceptation** (au format BDD)

On peut parfois être amené à écrire plusieurs tests d'acceptation (concrets) pour illustrer une condition d'acceptation (abstrait)

Autres formalismes possibles pour exprimer des exemples (scénarios de tests)

Gherkin

```
Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers
```

```
Scenario: Add two numbers
  Given I have entered 50 into the calculator
  And I have entered 70 into the calculator
  When I press add
  Then the result should be 120 on the screen
```

Tables

eg.Division

numerator	denominator	quotient?
10	2	5.0
12.6	3	4.2
22	7	~=3.14
9	3	<5
11	2	4<_<6
100	4	33

|eg.Division|

numerator	denominator	quotient?
10	2	5
12.6	3	4.2
100	4	33

Mots Clés

```
*** Settings ***
Test Template    Calculate
Library          CalculatorLibrary

*** Test Cases ***
Additions        Expression    Expected
Additions        12 + 2 + 2    16
                  2 + -3           -1

Subtractions     Expression    Expected
Subtractions     12 - 2 - 2    8
                  2 - -3           5

Multiplication   Expression    Expected
Multiplication   12 * 2 * 2    48
                  2 * -3           -6

Division         Expression    Expected
Division         12 / 2 / 2    3
                  2 / -3           -1

Calculation error [Template] Calculation should fail
kekkonen        Invalid button 'k'.
${EMPTY}         Invalid expression.
1 / 0           Division by zero.

*** Keywords ***
Calculate
[Arguments]    ${expression}  ${expected}
Push buttons   C${expression}=
Result should be ${expected}

Calculation should fail
[Arguments]    ${expression}  ${expected}
${error} =      Should fail  C${expression}=
Should be equal ${expected}  ${error}
```

En résumé : Un exemple de rédaction de story



En tant que Corinne,
Je veux pouvoir **m'enregistrer sur le site**
Afin de pouvoir y naviguer et l'utiliser



Une maquette rapide à main levée pourrait être ajoutée à la story et servir de support de discussion

Conditions d'acceptation :

Etant donné la page d'accueil du site avec un bouton « enregistrez vous »
Quand je clique sur le bouton « enregistrez vous »
Alors j'arrive sur le formulaire d'enregistrement

Etant donné le formulaire d'enregistrement

Quand je m'enregistre
Alors je reçois un email de demande de confirmation
Et cet email contient un lien de confirmation

Etant donné un lien dynamique sur une page de confirmation d'enregistrement

Quand j'accède à la page mon enregistrement est confirmé
Alors je reçois un email de confirmation d'enregistrement
Et je peux désormais me connecter au site

*Une rubrique **champs des formulaires** avec leurs contraintes permet d'être plus concis c-a-d moins verbeux dans la description des détails :
La spécification doit rester « légère »*

Précisions :

Champs du formulaire d'enregistrement :

Nom, prénom, adresse email, adresse (pour localisation de l'animal) et mot de passe sont obligatoires.
Mot de passe de 8 caractères minimum (pas d'autres contraintes sinon trop complexe pour Denise).

En résumé : Un autre exemple de rédaction de story



Peetic

En tant que Corinne,
Je veux pouvoir **me connecter au site**
Afin de l'utiliser (enregistrer un animal,
chercher des amis pour celui-ci, ...)



Conditions d'acceptation :

Etant donné que je suis sur la page d'accueil **et** que je ne suis pas connecté
Quand je remplis mon adresse mail et mon mot de passe
Alors le bandeau d'entête indique que je suis connecté

Etant donné que je suis sur la page d'accueil **et** que je ne suis pas connecté
Quand je remplis mon adresse mail et mon mot de passe
Et que mon mot de passe est faux
Alors je suis redirigé vers la page de login dédié

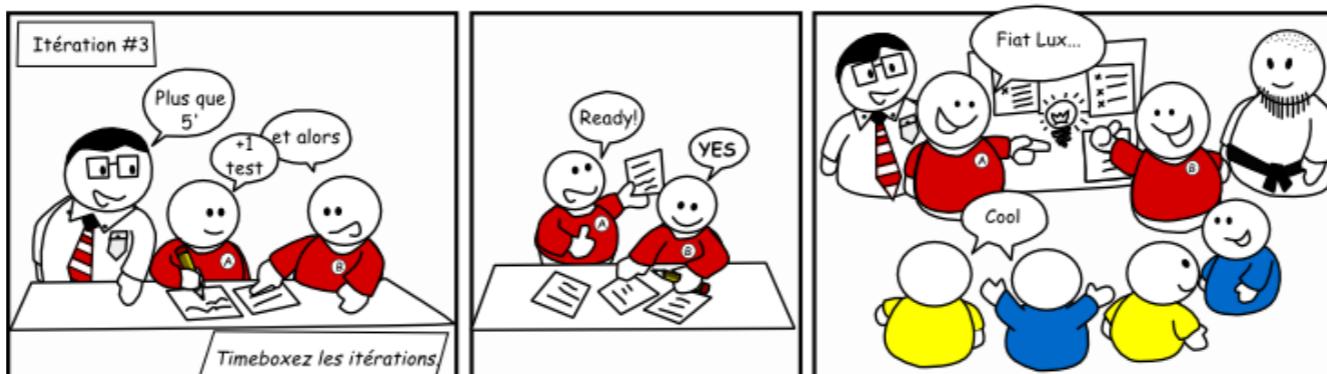
Etant donné que je suis sur la page d'accueil **et** que je ne suis pas connecté
Quand je remplis mon adresse mail et mon mot de passe
Et que mon adresse mail n'est pas connu
Alors je suis redirigé vers la page de login dédié



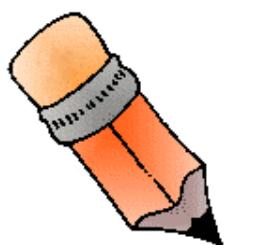
PODOJO :

Ecriture de votre première

story avec l'atelier USTA



Extrait : <https://twitter.com/ramuncho/status/453988188290179072/photo/1>



Des spécifications plus ou moins détaillées ...

suivant les besoins des développeurs pour satisfaire la *bonne* valeur métier et permettre de *bien* tester la story (INVEST) en fin de développement



Peetic



En tant que maître de chien

Je veux pouvoir inscrire
mon chien
à une réunion de confirmation

Afin de soigner son pedigree



Critères d'acceptance :

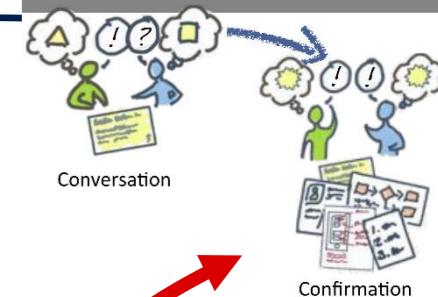
✓ **Inscription acceptée :**
l'inscription d'un chien à une confirmation est validée (succès)

✓ **Inscription refusée :**
l'inscription d'un chien à une confirmation est refusée, car la salle a atteint sa limite (échec)



Précisions :

Possibilité de lister les champs d'un formulaire, d'indiquer les champs obligatoires, les valeurs par défaut



Si nécessaire,
tout autre pour clarifier le besoin (diagrammes...)

Scénario : Inscription refusée : plus de place ...

Scénario : Inscription acceptée et limite atteinte

Scénario : Inscription acceptée

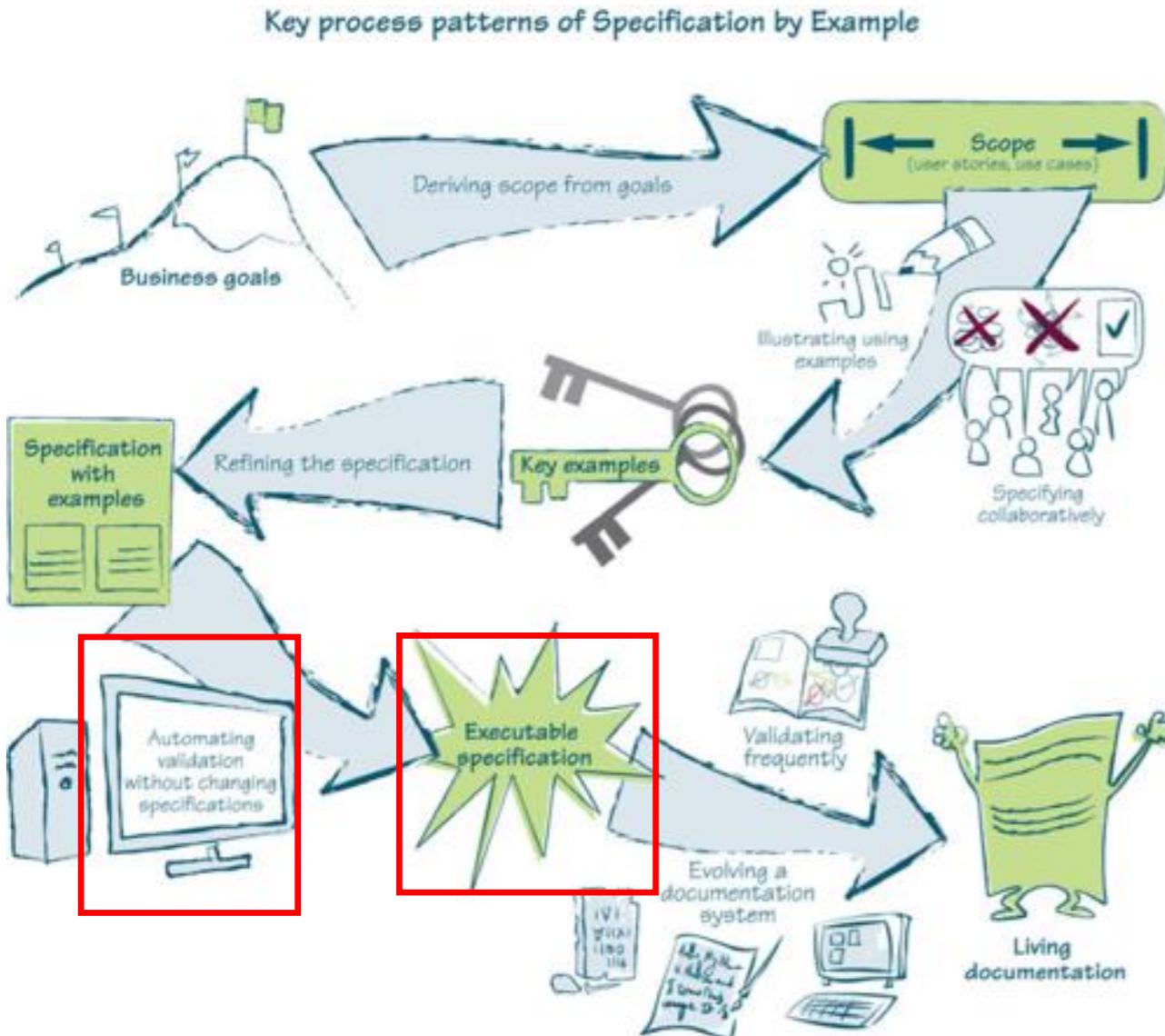
Etant donné Corinne propriétaire de Corsaire **et** une confirmation pour la race d'Epagneul annoncée pour le 15 Décembre avec 23 inscrits

Tests d'acceptance sous forme d'exemples ...

Quand Corinne inscrit son épagneul Corsaire de 2 ans à la confirmation du 15 décembre

Quand Corinne inscrit son épagneul Corsaire de 2 ans à la confirmation du 15 décembre

Avec de vraies valeurs, il est ensuite alors possible d'automatiser les exemples (tests) si on le souhaite afin d'obtenir une spécification exécutable !



Automating validation
without changing specification

Une **Spécification avec des exemples** automatisés (tests) qui est compréhensible et accessible par tous les membres de l'équipe devient une **spécification exécutable**

Les différentes étapes de l'exemple à la spécification exécutable avec *jbehave*

1. Write story

Exemple

Plain text

Scenario: A trader is alerted of status

```
Given a stock and a threshold of 15.0
When stock is traded at 5.0
Then the alert status should be OFF
When stock is traded at 16.0
Then the alert status should be ON
```

Configuration
Indispensable pour mettre en place l'automatisation

3. Configure Stories

Only once

```
public class TraderStories extends JUnitStories {
    public Configuration configuration() {
        return new MostUsefulConfiguration()
            .useStoryLoader(new LoadFromClasspath(this.getClass()))
            .useStoryReporterBuilder(new StoryReporterBuilder()
                .withCodeLocation(codeLocationFromClass(this.getClass())))
            .withFormats(CONSOLE, TXT, HTML, XML);
    }

    public List<CandidateSteps> candidateSteps() {
        return new InstanceStepsFactory(configuration(),
            new TraderSteps(new TradingService())).createCandidateSteps();
    }

    protected List<String> storyPaths() {
        return new StoryFinder().findPaths(codeLocationFromClass(this.getClass()),
            "**/*.story");
    }
}
```



Ecriture du code métier de production permettant d'implémenter le périmètre fonctionnel du test

2. Map steps to Java

POJO

```
public class TraderSteps {
    private TradingService service; // Injected
    private Stock stock; // Created

    @Given("a stock and a threshold of $threshold")
    public void aStock(double threshold) {
        stock = service.newStock("STK", threshold);
    }

    @When("the stock is traded at price $price")
    public void theStockIsTraded(double price) {
        stock.tradeAt(price);
    }

    @Then("the alert status is $status")
    public void theAlertStatusIs(String status) {
        assertThat(stock.getStatus().name(), equalTo(status));
    }
}
```

Automatisation : steps du scénario transformées en code pour pouvoir être exécutées

Assertion ⇒

c'est bien un test d'acceptation !

4. Run Stories



With any of

Exécution de l'automatisation : lancement du test

5. View Reports

HTML

Scenario: A trader is alerted of status

```
Given a stock and a threshold of 15.0
When stock is traded at 5.0
Then the alert status is OFF
When stock is traded at 16.0
Then the alert status is ON
```

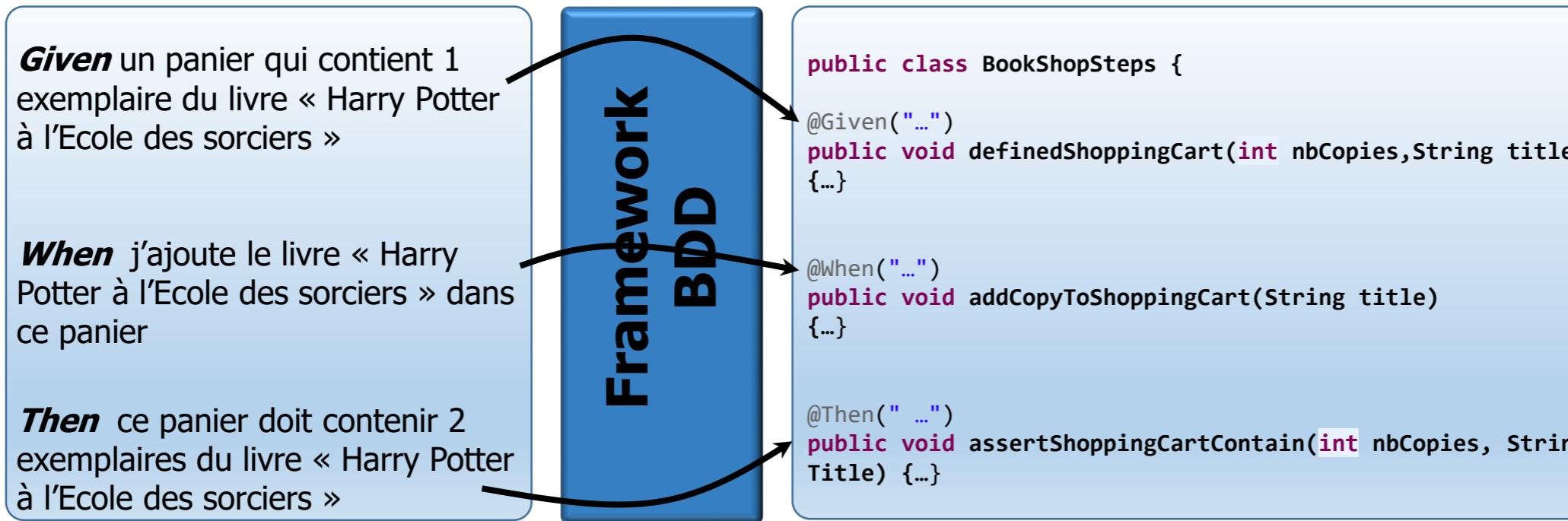
Spécification exécutable

Principe de fonctionnement des frameworks BDD pour l'automatisation des tests d'acceptation



Cucumber

... et bien d'autres



- ✓ Les outils de BDD permettent de traduire un scénario en langage naturel en appels de méthodes.
- ✓ La grammaire **Given/When/Then** (appelé langage **Gherkin**) permet de réaliser le mapping entre les « étapes » du scénario et les « steps » du code.

Un exemple de spécification exécutable obtenue avec FitNesse

Au delà d'une simple démarche d'automatisations des tests, il faut percevoir les spécifications exécutables comme une véritable opportunité de rapprocher les populations techniques et fonctionnelles autour d'une vision partagée et non ambiguë du produit logiciel.

Final expression of our example

do with bank

Our first business rule says that a new account should have a balance of 0.00 dollars.

open	checking	account	12345-67890	under the name of	Spongebob	?	Squarepants
check	that balance of account	12345-67890	is	\$0.00			

Our next rule says that the bank should not take any fees when we deposit money in our account.

deposit	\$100.00	in account	12345-67890	
check	that balance of account	12345-67890	is	\$100.00

The following rule says that a customer should be able to withdraw funds if the balance of his account is sufficient.

withdraw	\$50.00	from account	12345-67890	
check	that balance of account	12345-67890	is	\$50.00
reject	withdraw	\$75.00	from account	12345-67890
check	that balance of account	12345-67890	is	\$50.00
accept	withdraw	\$25.00	from account	12345-67890

Un exemple de spécifications exécutables venant d'être exécutées et en succès.

Panorama des différents outils pour l'automatisation des tests d'acceptation

Tables & Wiki

Scénarios (BDD) (Given-When-Then)



1. Describe behaviour in plain text

```
Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers

  Scenario: Add two numbers
    Given I have entered 50 into the calculator
    And I have entered 70 into the calculator
    When I press add
    Then the result should be 120 on the screen
```

Extrait : <https://cucumber.io/>



eg.Division		
numerator	denominator	quotient?
10	2	5.0
12.6	3	4.2
22	7	~=3.14
9	3	<5
11	2	4<_<6
100	4	33

The wiki markup for our table above :

```
|eg.Division|
|numerator|denominator|quotient?|
|10|2|5|
|12.6|3|4.2|
|100|4|33|
```

Extrait : <http://www.fitnesse.org/>

Mots Clés

(Data-driven test development)

ROBOT FRAMEWORK

```
*** Settings ***
Test Template Calculate
Library CalculatorLibrary

*** Test Cases ***
Additions           Expression   Expected
Additions           12 + 2 + 2   16
                           2 + -3      -1

Subtractions        Expression   Expected
Subtractions        12 - 2 - 2   8
                           2 - -3      5

Multiplication      Expression   Expected
Multiplication      12 * 2 * 2   48
                           2 * -3      -6

Division            Expression   Expected
Division            12 / 2 / 2   3
                           2 / -3      -1

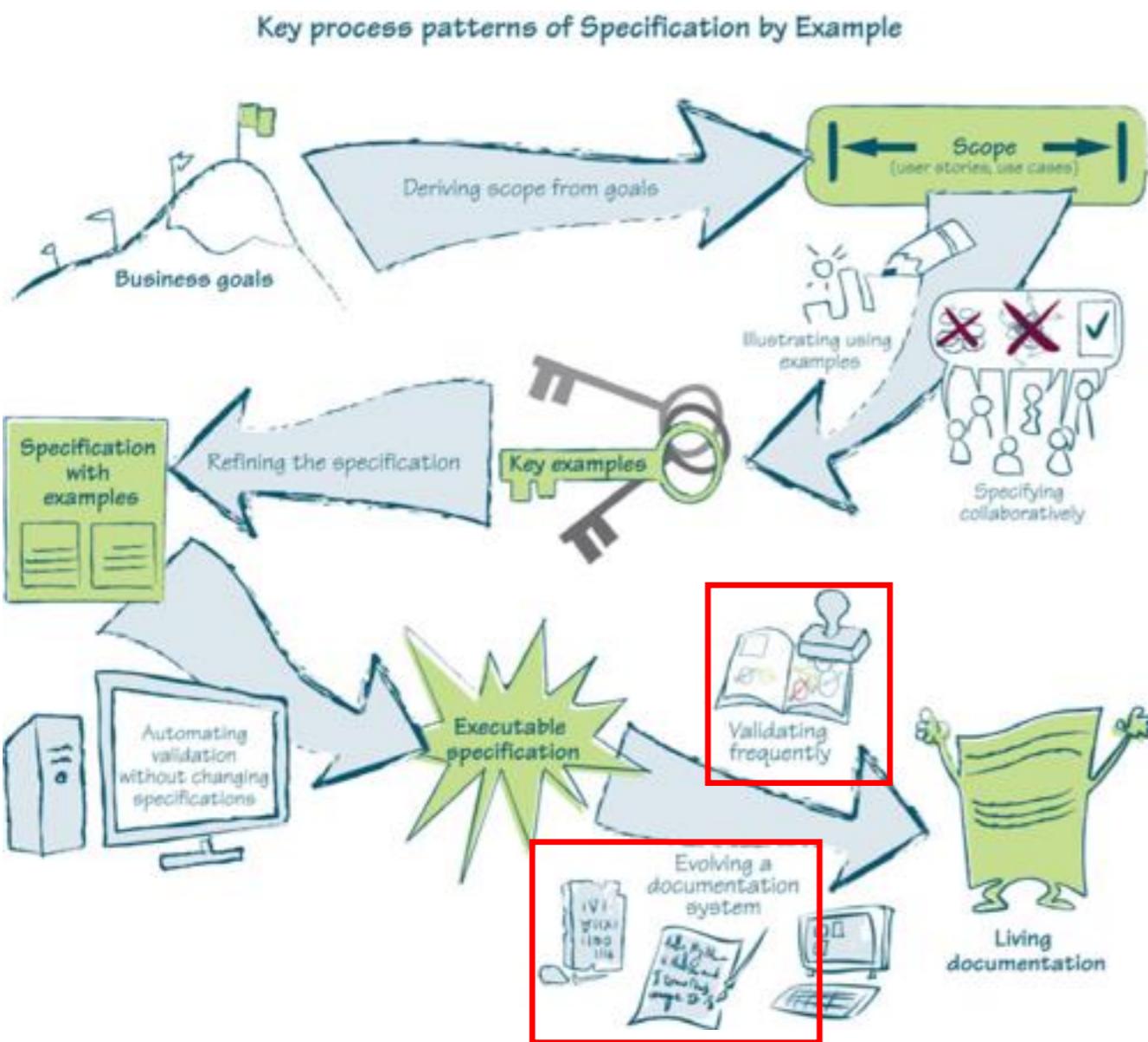
Calculation error   [Template]  Calculation should fail
Calculation error   kekkonen    Invalid button 'k'.
Calculation error   ${EMPTY}     Invalid expression.
Calculation error   1 / 0       Division by zero.

*** Keywords ***
Calculate
[Arguments] ${expression} ${expected}
Push buttons  C${expression}=_
Result should be ${expected}

Calculation should fail
[Arguments] ${expression} ${expected}
${error} = Should fail C${expression}=_
Should be equal  ${expected} ${error}
```

Extrait : <http://robotframework.org/>

Et pour finir le cheminement de la Spécification par l'exemple



Validating frequently

Evolving a documentation system

Proposer une **documentation vivante** : facile à comprendre, cohérente et organisée

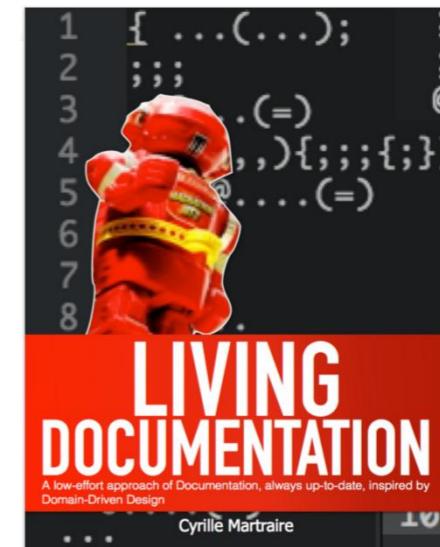
En savoir un peu plus sur la documentation vivante...



Living Documentation : vous allez aimer la documentation ! (Cyrille Martraire)

A visualiser sur :

<https://www.youtube.com/watch?v=Tw-wcps7WqU>



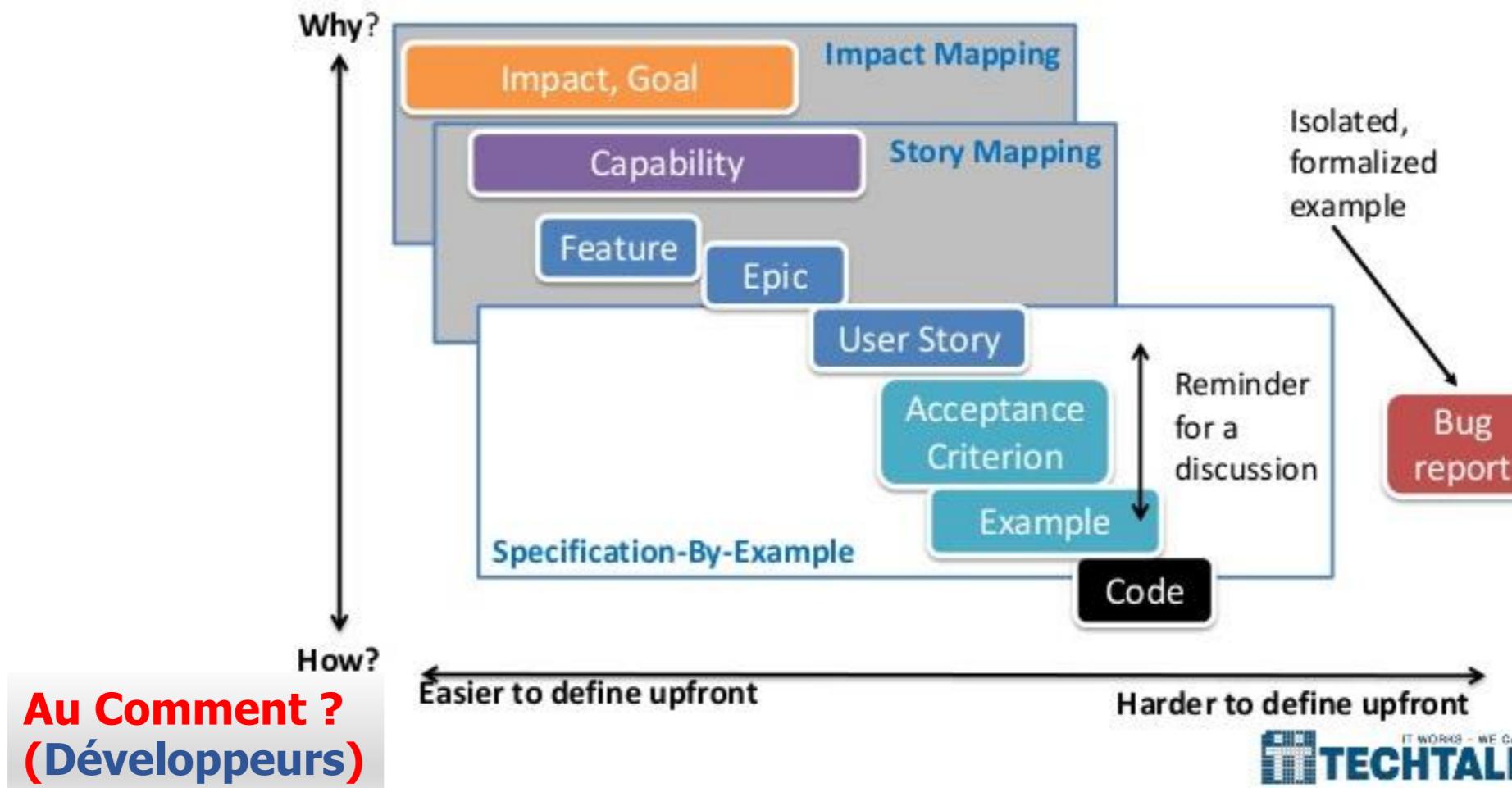
Livre disponible sur :

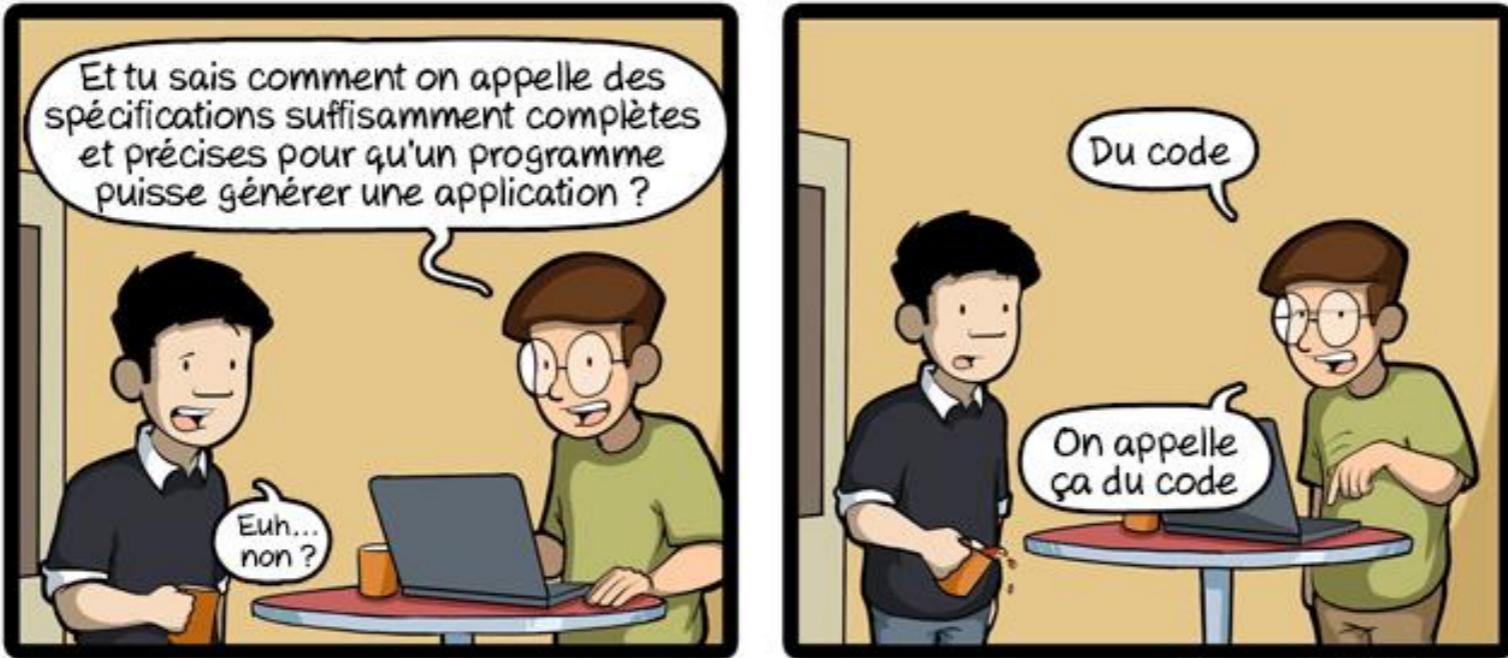
<https://leanpub.com/livingdocumentation>

La story du POurQuoi au Comment

Establishing a shared understanding

Du POurQuoi ?





CommitStrip.com