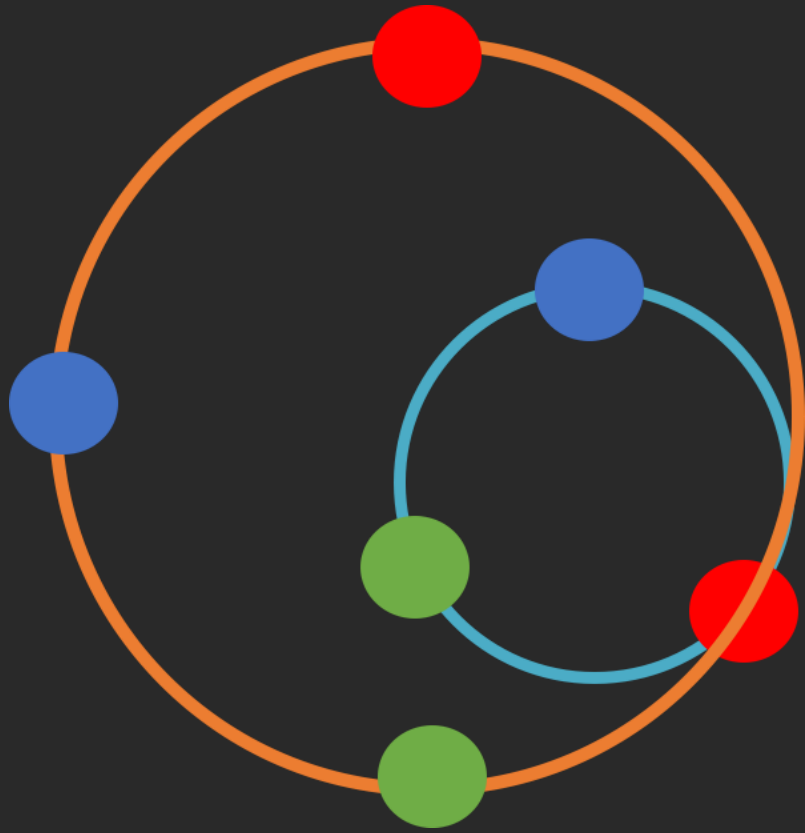
ATDD à double boucle
automatiser un test d'acceptation



FÉLIX-ANTOINE BOURBONNAIS

B.ING., M.SC, PSM

Diapositives, **références** et **vidéos**

http://conferences.elapsetech.com/atdd-double-boucle

Cette présentation s’adresse à un public **avancé** et **très technique.**

ELAPSE Technologies

Objectifs

- Comprendre le **cycle** de l'ATDD
- Comprendre comment **piloter** le développement à partir d'une ***User Story***.
- Comprendre le lien entre le **TDD** et **l'ATDD**
- Savoir comment exécuter des scénarios à **différents *niveaux***
- Avoir un aperçu de comment enchaîner (**pipeline**) les tests

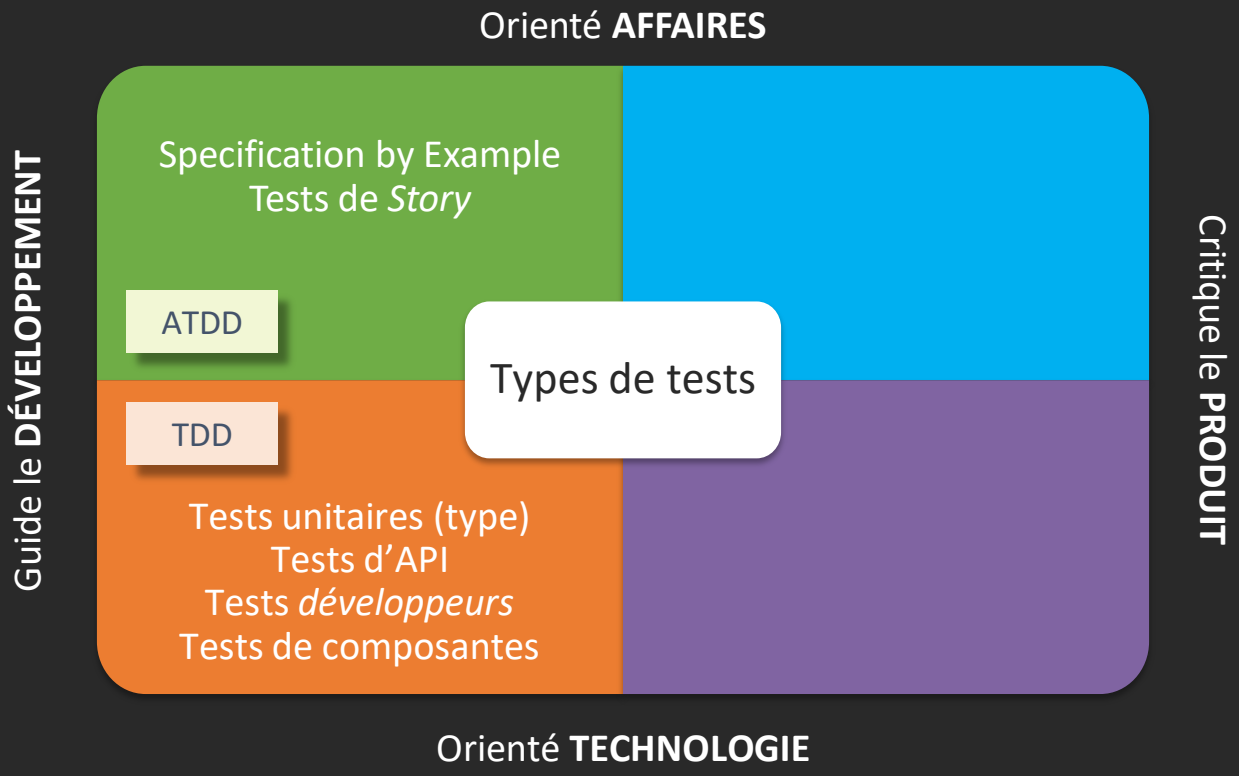
Les tests de Stories / acceptation

Qu’est-ce qu’un test d’acceptation ou une *Story Test* ?

Est-ce que je produis la bonne chose?

(Building the right thing)

Les **types** de tests…



Tiré du livre More Agile Testing

La pyramide des tests

Tests d’acceptation/*Story Tests*

!=

Tests bout-en-bout

La **portée** (scope) d'un test est la **distance** entre le point d'appel et de validation

Un **type** de test (ex.: *Feature/Story*)
n'est **pas** directement **relié** à
une **portée** (ex.: bout-en-bout).

Attention en automatisant !

Fragilité des tests

% du portfolio de tests auto. (tous les types)

Bout-en-bout
Toutes composantes

Une composante intégrée

Une ou quelques classes

~10%

~20%

~70%

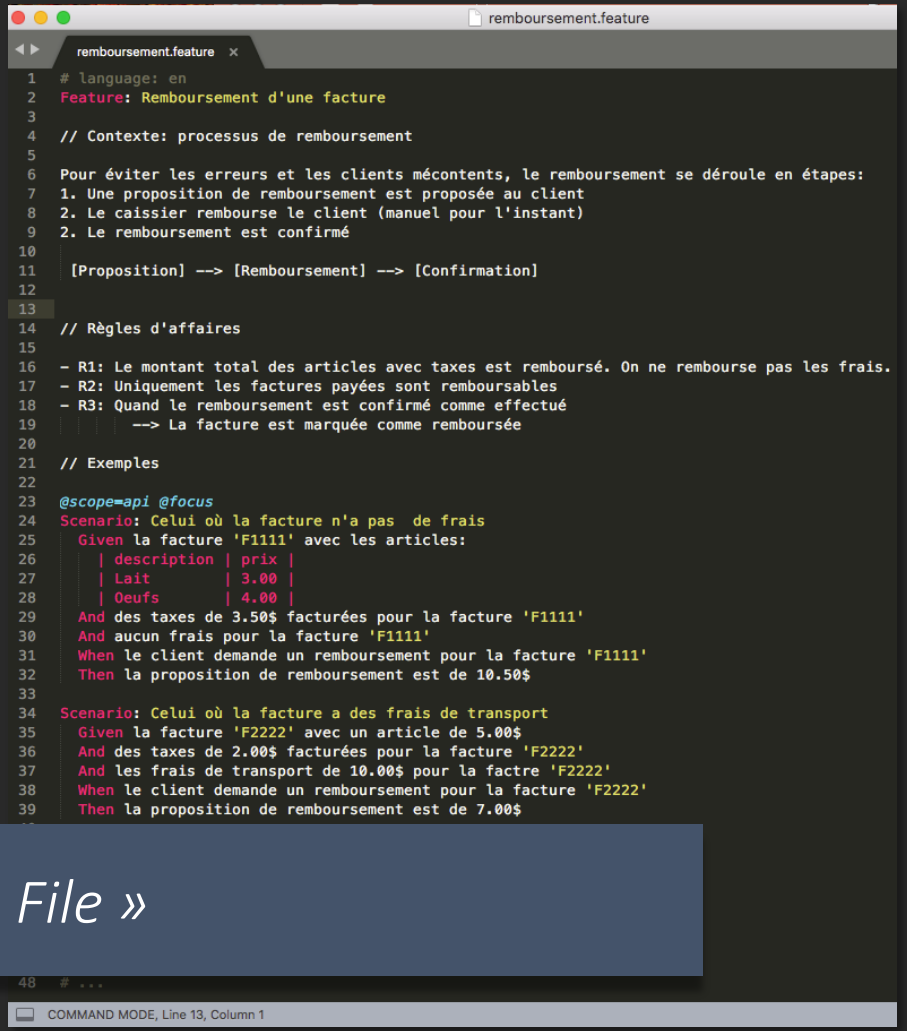
Large (L)

Moyen (M)

Petit (S)

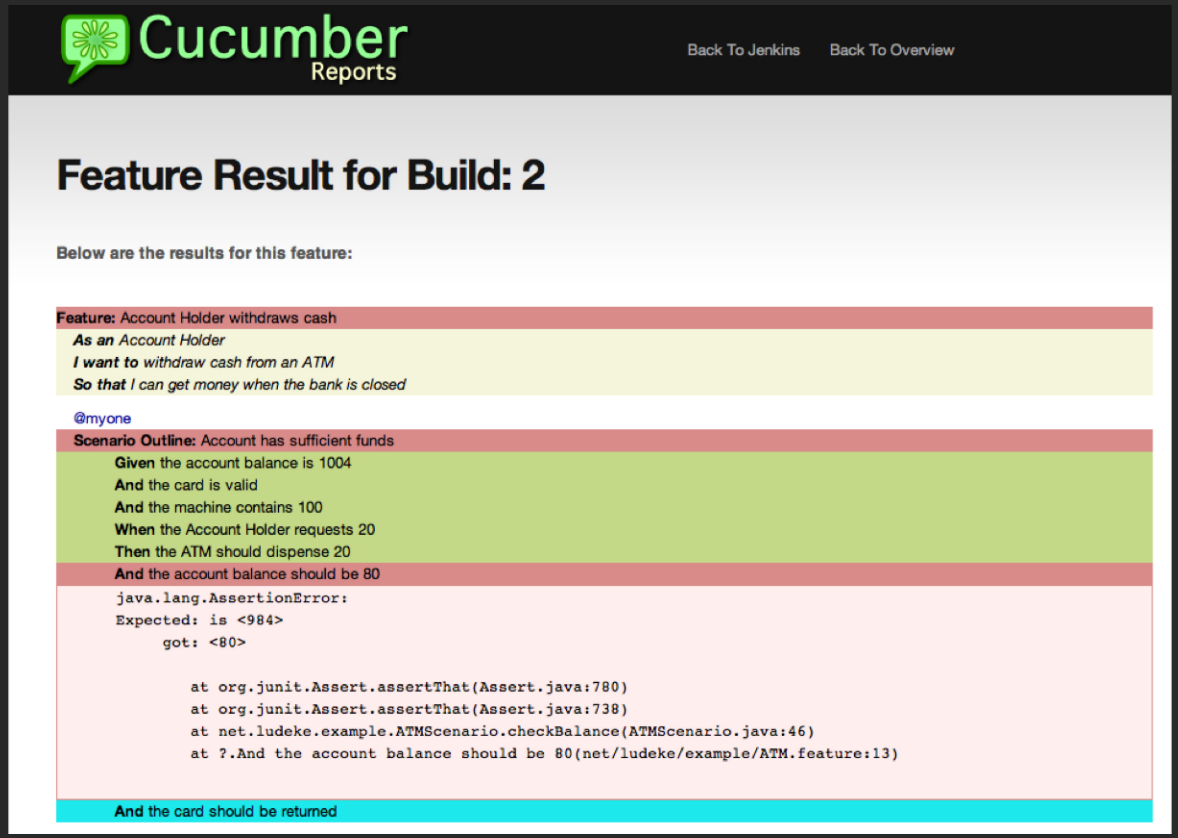
Image par Gamma-Ray Productions sur Flickr

Le Gherkin



« Feature File »

Le Gherkin


Cucumber
Reports
Back To Jenkins
Back To Overview
Feature Result for Build: 2
Below are the results for this feature:
Feature: Account Holder withdraws cash
As an Account Holder
I want to withdraw cash from an ATM
So that I can get money when the bank is closed
@myone
Scenario Outline: Account has sufficient funds
Given the account balance is 1004
And the card is valid
And the machine contains 100
When the Account Holder requests 20
Then the ATM should dispense 20
And the account balance should be 80
java.lang.AssertionError:
Expected: is <984>
got: <80>
at org.junit.Assert.assertThat(Assert.java:780)
at org.junit.Assert.assertThat(Assert.java:738)
at net.ludeke.example.ATMScenario.checkBalance(ATMScenario.java:46)
at ?.And the account balance should be 80(net/ludeke/example/ATM.feature:13)
And the card should be returned

Gherkin != BDD

Ce n'est pas notre sujet!

Le **BDD** est une méthode axée sur la **collaboration** pour **comprendre** et **spécifier** les **besoins d'affaires**

Automatisation à partir du Gherkin (rappel)

Les outils en bref

- Cucumber et sa famille
- SpecFlow

Anatomie de Cucumber

- Feature File
- Step Definitions / Glues
- Support Classes (à venir)

ATDD à simple boucle

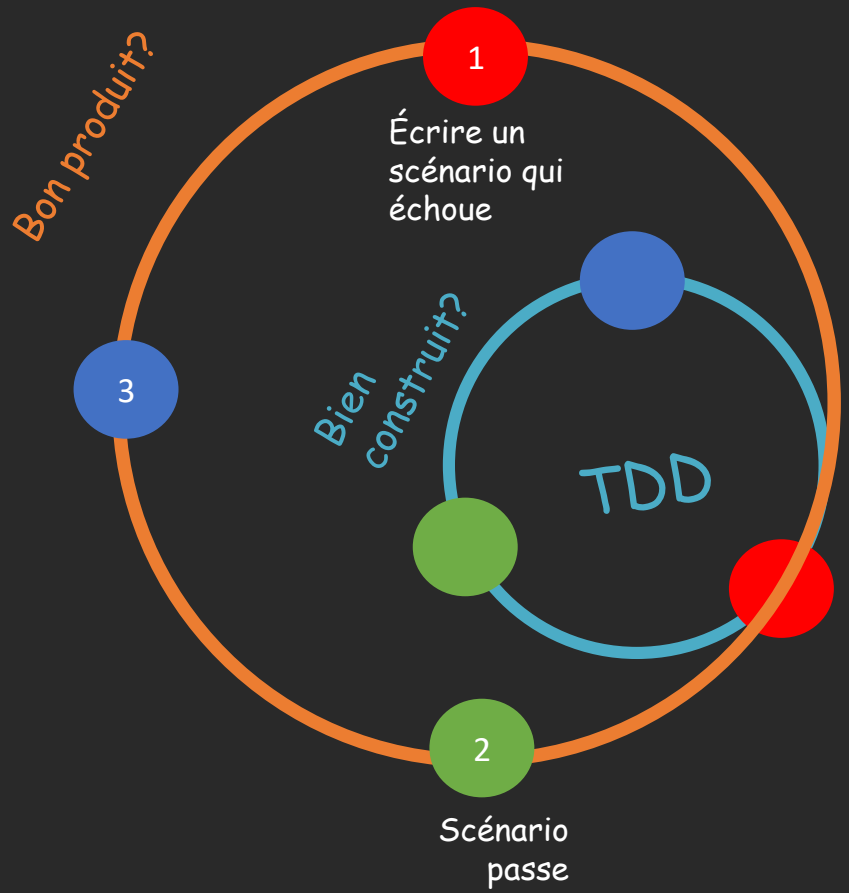
ATDD:
Acceptance* Test Driven Development

* Ne pas confondre avec les tests d'acceptation utilisateur. Acceptation ici = test des critères d'acceptation de la Story.

Le but de l’ATDD est de **piloter** le développement du système

Notre but est de nous assurer de répondre aux **critères** de la ***User Story*** et la considérer comme **terminée**.

ATDD



Outils utilisés pour la démonstration

- Java
- Cucumber-JVM
- Junit
- Eclipse (IDE)
- Maven

Vidéo de la démonstration
http://conferences.elapsetech.com/atdd-double-boucle

Démonstration <partie 1>

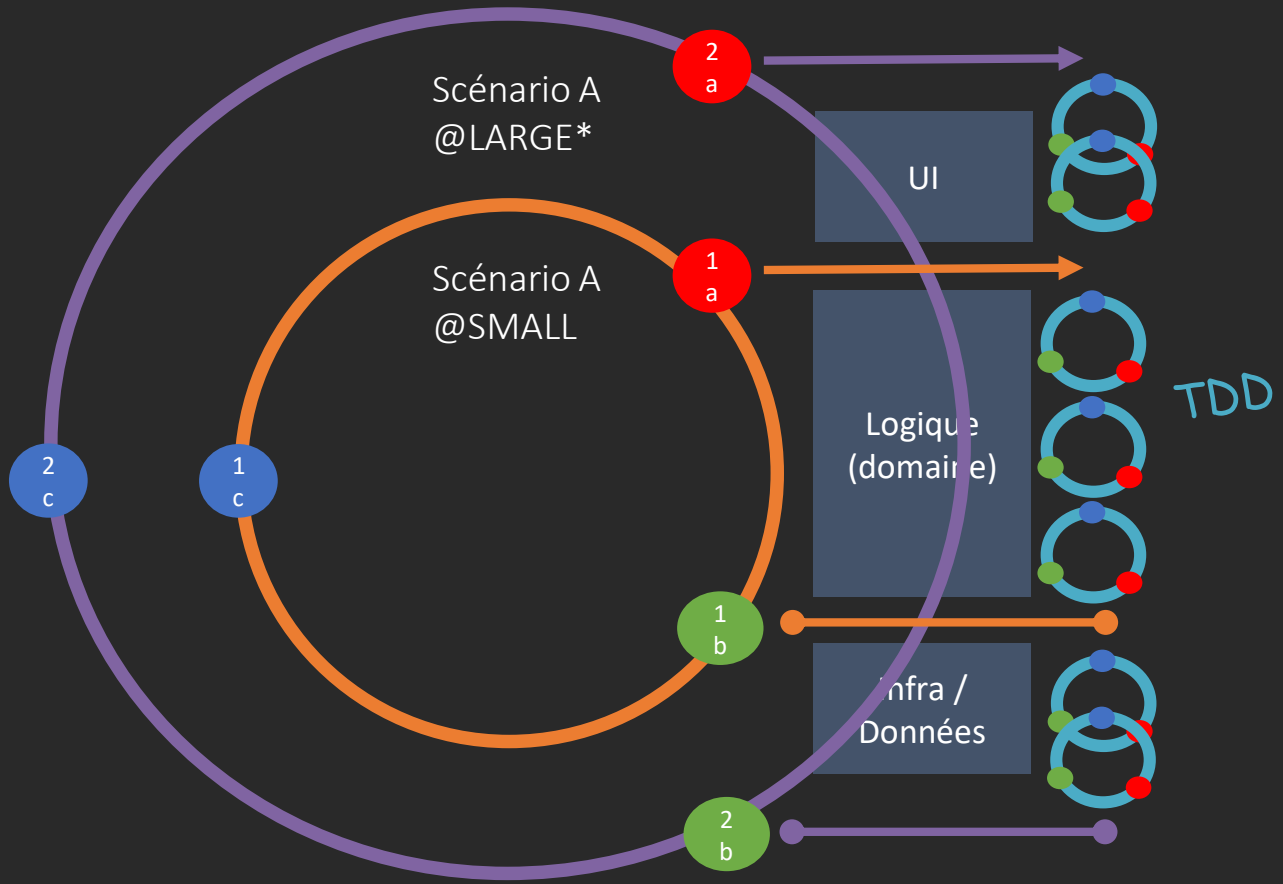
ELAPSE Technologies

ATDD à double boucle

Constat: nous n’avons pas codé le *UI* et tous les tests passent !?!

La solution n’est **pas** de choisir la portée de chaque scénario.

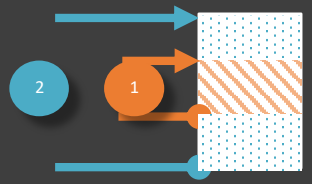
ATDD à double boucle



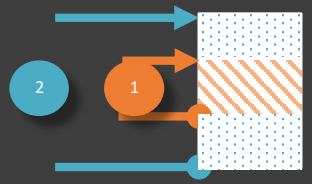
* Uniquement certains scénarios (voir Pyramide des tests). Pourrait être @MEDIUM ou autre portée…

Exemple

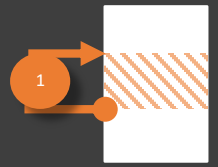
@scope=web
Scenario: Transfering money adjusts the account balances
Given an account 111 with 1000$ in it
And an account 222 with 500$ in it
When I create a transaction of 100$ from 111 to 222
Then the account 111 has 900$ in it
And the account 222 has 600$ in it



@scope=web
Scenario: Transfering money creates an accepted transaction log
Given an account 333 with 1000$ in it
And an account 444 with 500$ in it
When I transfer 100$ from 333 to 444
Then a transaction log is created for the amount of 100$



Scenario: Transfering money when the account doesn't have the funds
Given an account 555 with 99$ in it
And an account 666 with 500$ in it
When I transfer 100$ from 555 to 666
Then a transaction log shows that the transfer was refused



Vidéo de la démonstration
http://conferences.elapsetech.com/atdd-double-boucle

Démonstration <partie 2>

ELAPSE Technologies

Conclusion

Résumé

Étape 1

- Nous avons écrit 1 *Story Test* pour nous assurer du **comportement** tel que spécifié dans la ***User Story***.
- Nous avons piloté l'écriture de la **logique d'affaires** pour faire passer le premier scénario.
- Nous avons écrit des **tests unitaires en TDD** pour nous assurer de couvrir et **bien construire** le code de production.

Résumé

Étape 2

- Nous avons **branché** le **même scénario** à une **portée** plus grande afin de **piloter** le développement de l'API (UI)

Résumé

Étapes suivantes

- Implémenter les autres scénarios avec la même boucle
- Par contre, ces scénarios ne requièrent **pas** d'être pilotés à une portée plus grande que *AppService+FakeDB*.

Pourquoi ?

merci .

ELAPSE
Technologies

Toutes nos présentations

conferences.elapsetech.com

Diapositives et références

conferences.elapsetech.com
/atdd-double-boucle

Pascal Roy

Site
elapsetech.com

Twitter
@fbourbonnais

Courriel
fbourbonnais@elapsetech.com

ELAPSE Technologies