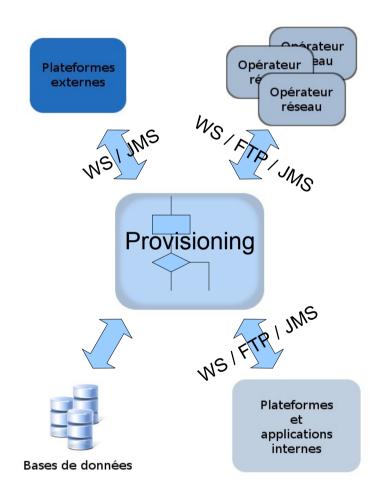
Tests d'acceptation de workflows automatisés avec Concordion

Situation initiale

- Equipe Agile
 - Binômage
 - Tests unitaires
 - Itérations courtes
- Jungle informatique
 - Moteur de workflows
 - Interfaces multiples
 - Système en constante évolution
- Tests d'acceptation absents



Un dur constat

- Défauts en intégration
 - Enchaînement des tâches laborieux
 - Paramétrage manquant
 - Nombreuses livraisons nécessaires
- Pas de démonstration en fin d'itération
- Mauvaise compréhension des attentes du client

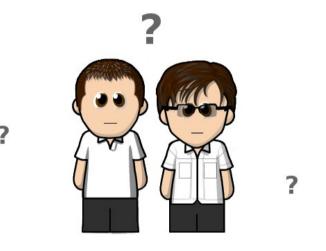


Comment s'améliorer?

- Volonté d'améliorer nos pratiques XP
- Tests fonctionnels mis en place, mais :
 - Lourds à maintenir
 - Pas de garantie de complétude
 - Rien à montrer au client
- Etude de différents frameworks de tests d'acceptation

Concordion?

FitNesse?



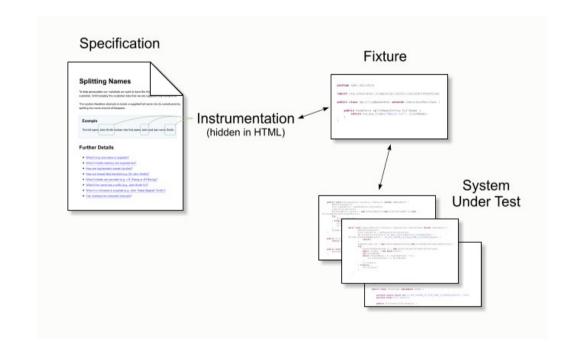
jUnit ?

Tests fonctionnels?

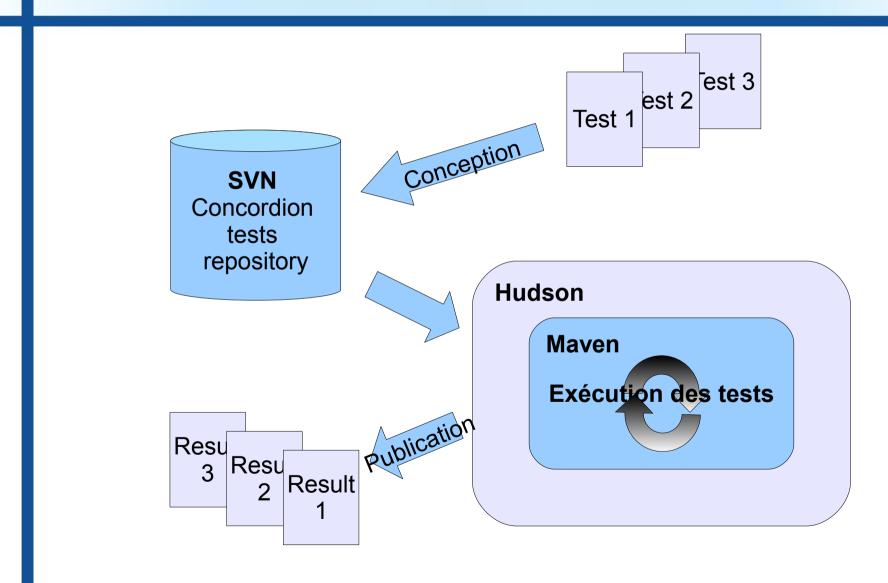
Pourquoi Concordion?

- Marketing
- Langage fonctionnel
- Documentation

Simple à intégrer dans l'environnement de développement

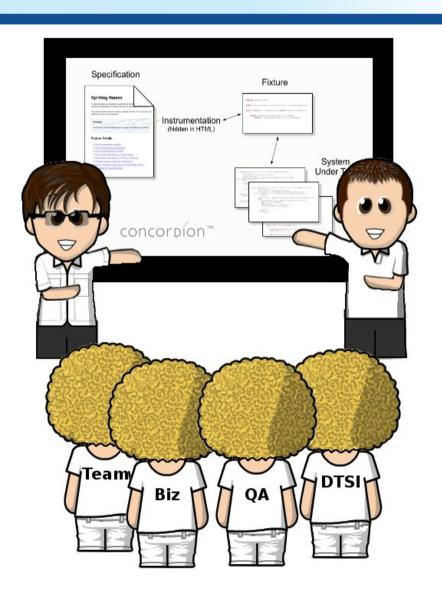


Intégration au sein de l'environnement de développement



Convaincre

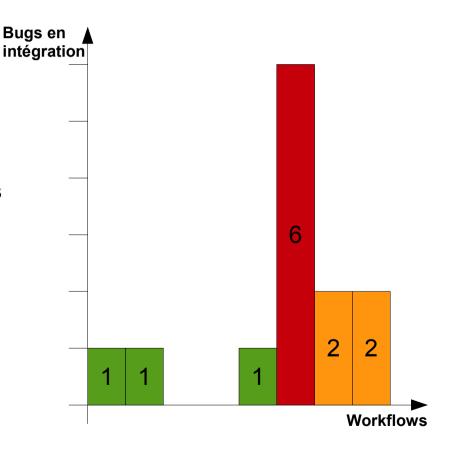
- Présentations sur l'outil Concordion
 - Intérêt pour le business et la MOA
 - Atouts en matière de qualité
 - Aspects de documentation
- Proof of concept
 - Composant « façade » : Catalogue
 - Périmètre restreint, coût de setup faible
- Au sein de l'équipe
 - Lever les doutes
 - Phase d'apprentissage



Première mise en œuvre au sein d'un projet

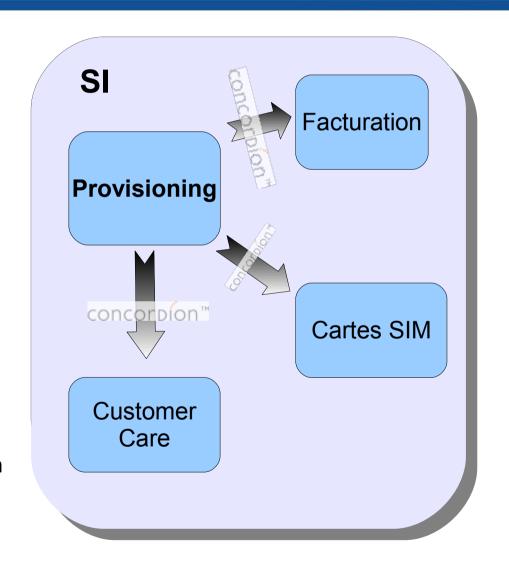
Evaluation du ROI

- Quelques workflows seulement
- Sous étroite surveillance du management
- Résultats probants
 - Déroulement des workflows en une fois (au lieu de 2 ou 3 livraisons)
 - Nombre de bugs très faible
 - Complétude fonctionnelle atteinte
- Réaction mitigée
 - QA conquise
 - Business sceptique

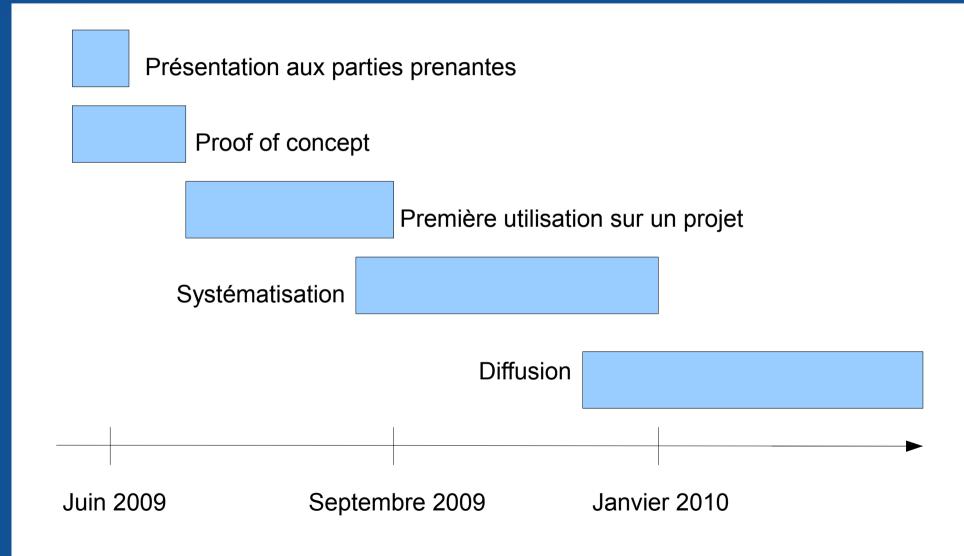


Généralisation

- Utilisation sur les nouveaux projets
- Retro-engineering sur les workflows existants
- Propagation au sein du SI
 - Facturation
 - Gestion des cartes SIM
 - Automatisation des tests d'intégration



Le Planning



Exemple – Composant simple

Checkpoint - Exécution normale

Description d'un cycle d'exécution normal de Checkpoint, en Dry Run.



Consommation et archivage d'une commande

Dans le répertoire des commandes client (sandbox/input), on dépose un fichier CSV ActivationRequest_20090101_1200.csv contenant les demandes

1;2009-01-01;NA;1234567894567;nom;prenom;;;;;;0;IR

2;2009-01-01;NA;1234567894557;nom2;prenom2;;;;;;0;IR

Dans la configuration client, les répertoires d'archivage en entrée et en sortie sont respectivement : sandbox/input/archives et sandbox/output/archives.

L'exécution de Checkpoint entraîne la consommation de la commande client dans le répertoire de dépôt du FTP. Ce dernier est désormais vide.

Par ailleurs, la demande ayant été archivée, le répertoire d'archivage des commandes contient désormais 1 fichier(s), dont le nom ActivationRequest_20090101_1200_20090201_060000.000.csv correspond au nom du fichier de commande suivi de la date de traitement.

En sortie, les ordres générés ont été archivés dans le répertoire adéquat qui contient ainsi 2 fichiers, dont le nom est composé d'un préfixe issu de la datasource suivi de la date du traitement :

Nom du fichier

ActivationRequest_20090101_1200.csv_output_request_1_20090201_100000.000 xml
ActivationRequest_20090101_1200.csv_output_request_2_20090201_120000.000 xml

Le répertoire des fichiers en erreur (sandbox/output/errors) reste vide.

Checkpoint dépose le message JMS correspondant au mail client dans le dossier sandbox/output/mails

Nom du fichier

LEOBIZ_mail_summaryOK_20090201_090000.000.xml

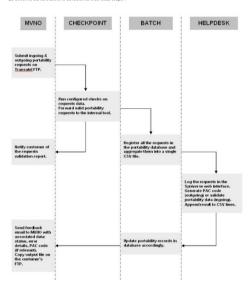
- · Comment les fichiers au nom invalide sont-ils gérés ?
- Que se passe-t'il si le fichier comporte une ligne erronée ?
- Quel est le comportement lorsqu'une ligne comporte un nombre de champs inattendu ?

Exemple Workflow (1)

Artemis - MVNA OUK - Prise en charge des ordres de portabilité

Les MVNOs anglais peuvent effectuer leurs demandes de portabilité entrante et sortante en déposant des fichiers au format csv dans un répertoire dédié nommé PortabilifyRoquests. Les ordres déposés sont pris en charge par Checkpoint puis transmis au système afin d'être enregistrés dans Interface Syntheses, dont le rôle set de centraliser les demandes de portabilité en Aggletere.

Le schéma suivant décrit le déroulement de cette étape



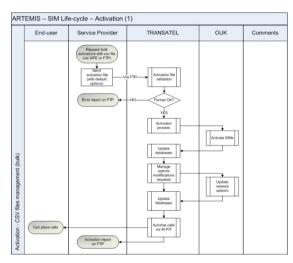


Exemple Workflow (2)

Artemis - MVNA OUK - Activation d'une ligne via FTP



Le dépôt d'une demande d'activation (NA) valide sur le FTP doit déclencher un workflow qui va réaliser les différentes opérations de provisioning nécessaires à l'activation de la ligne



Activation d'une ligne par requête sur le FTP

On dépose un fichier ActivationRequest_20090101_1200.csv dans le répertoire de dépôt MVNA-OUK sandbox input l'activation, contenant l'ordre suivant 1;2009-01-01;MA;8944133397300000196;447000000196;1234;nonzprenom;;;DOD;YdR;

L'exécution de Checkpoint entraîne le lancement d'un workflow d'activation MVNA-OUK nommé OUKActivation, comportant les données suivantes

Variable workflow	Valeur
орТуре	NA
iceld	894413339730000019
msisdn	+447000000196
firstName	prenom
lastName	nom
wholesalerName	ARTEMIS_POSTPAID
telcoProfile	UK MVNA PAYM INT1
subscriberMvnoReference	1234
OUK_DataOnDemand	on
OUK_AdultContent	on
OUK_Roaming_International	on

A l'issue de l'exécution du workflow, les différentes interfaces de provisioning ont été mises à jour.

Un subscriber a été créé dans la core DB, de même que ses informations de contact

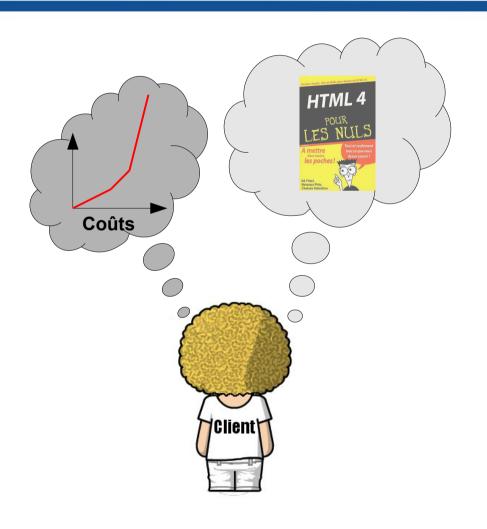
Champ subscriber	Valeur	Champ contact	Valeur
subscriberNumber	00007176.00000001	firstName	prenon
subscriberId	1	lastName	nom
statusFlag	A		
ratePlanName	MVNA_OUK_1		

Bilan

Un an plus tard...

Les peurs

- Coût en temps de mise en place
 - Grande crainte du management
 - Inquiétude des développeurs
- Surcharge de travail pour les clients
- Trop technique pour les clients
- Manque d'efficacité



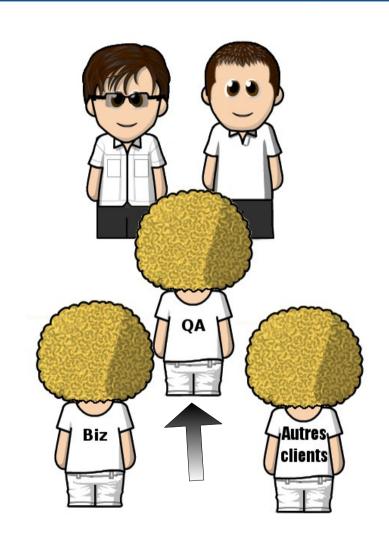
Les embûches

- Coût de maintenance
 - Plus linéaire qu'exponentiel, mais bien présent
 - Effort supplémentaire par rapport aux tests unitaires
- Pratique laborieuse en environnement non-Agile
 - Implication et disponibilité des intervenants extérieurs fluctuantes
 - Entretenir la motivation de l'équipe et des clients



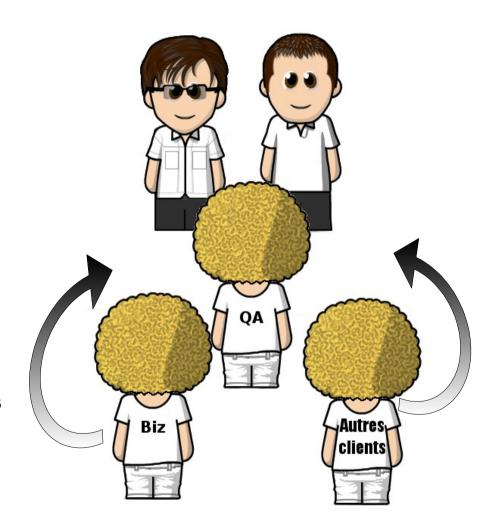
Ce que nous y avons gagné

- Le « Given When Then » adopté
- Documentation à chaud
- Emergence d'un Domain Specific Language
- Collaboration avec QA



ToDo

- Faciliter la conception des tests
 - Finaliser et publier le DSL
 - Templates HTML
 - Sandbox pour tester le comportement du système
- Centraliser et répertorier les tests
 - Repository accessible par les différents intervenants
 - Permettre aux clients et à la QA de soumettre des tests aux développeurs



Quelques extraits de code

HTML specification

Fixture

```
public String readOutputLines(String filePath) throws Exception {
    return CSVUtil.readCSVLines(filePath);
}

public String readOutputLines(String folder, String file) throws Exception {
    return CSVUtil.readCSVLines(folder + File.separator + file);
}
...
```

Résultat

Pour le MVNO mvno1, le répertoire sandbox/mvno1/export/report contient le fichier PortingReport_20090201_0000.csv comportant les données : NP;8941030835112400140;+447234567890;;;OUK123456;2009-01-10;OK; PO:8941030835112400141;+447234567891;;:TSL123456;:OK;

Références

- Concordion : http://www.concordion.org/
- WeeWorld : http://www.weeworld.com/

Merci!

Questions & réponses