

## **Deux années de Continuous Delivery au pays des Traders**

Speakers : **Alexandre Victoor** et **Patrice Séqué** – Société Générale

Track : Agilité & Devops

Devvix France 2014 - Vendredi 17/04/2014 – 11h30 - 12h30

Projet Spark : application front office utilisée par les vendeurs et traders de la banque. Permet de créer des produits complexes. Workflow avec différents métiers. Appli mondiale.

### **Au départ**

Cycle en V il y'a 4 ans à la SG : étude de l'existant, architectes techniques et fonctionnels, experts => travail pendant un an pour définir une spéc. Ne convient pas au trader qui n'a pas le temps de valider une spéc de 200 pages. Cycle en V pour le métier : beau tunnel sans fin.

Au niveau de l'archi : pas de SOA, ESB, serveurs d'app, BPMN ... => coute trop cher, trop long même si belle Cathédrale à la fin.

Inadéquat pour le métier.

### **Contexte à la SG**

Installé dans 160 pays. Manipulation de flux financiers. La finance bouge très vite. Les traders savent coder et vont vite (macro excel). Très exigeants.

IT répartie dans plus de 13 pays. La grosse partie du SI est hébergée en France.

Applications utilisées 22h sur 24.

Beaucoup de data centers. 70 TO historisés chaque jour. Plusieurs centaines voire milliers d'applications.

### **Cap sur l'agilité**

Ecueil initial : Pas d'agile. Trop ambitieux.

Image : au lieu de créer une Cathédrale, ils vont créer une Maison.

Mise en place de l'agilité : coaching, story mapping, priorisation par la valeur métier, serious game, product box, buy a feature, prune the product tree.

Laurent Sarazin est intervenu pour convaincre le métier de passer en Agile.

JL de Morlon comme coach Agile.

Consultants formés venant de Xebia.

Quelques ateliers d'Innovation Games ont permis de définir un produit à sortir : backlog priorisé en un mois.

Ils ont challengé ce que demande le métier.

### **Méthodologie**

Sprint de 2 semaines avec une démo au bout.

Colocalisé à côté du métier. Product Owner très présent, au moins au début.

Equipe : Manager + archi technique + 1 MOA + 4 développeur

Livraison toutes les 2 semaines.

### **Architecture**

Petite appli web déployée sur Tomcat, techno bien connue par les Ops à la SG.

Pas de haute dispo, de cluster ... pas de réponses à des exigences non fonctionnelles qui n'existent pas.

Pas de SLA de 99.999 % = 2 minutes d'interruption de service sur l'année.  
Coupure possible pendant 2h sans perte financière.

### **Mise en œuvre**

2 semaines passent très vite. Pas de temps à perdre avec le déploiement.

Jenkins est utilisé à la SG.

Très simple au début : Jenkins, SVN, scripts SSH, binaires récupérés depuis le repo maven Nexus, redémarrage Tomcat, Hibernate en mode update. Pas prod ready.

### **Bilan intermédiaire**

#### Avantages :

Proximités métier

#### A améliorer :

Il y'a encore des développeurs spécialisés : IHM, back-end, workflow. Pas génial lorsque la personne n'est pas là.

Le fonctionnel compliqué mal partagé.

Pas mature pour le BDD.

#### Problèmes :

Déploiement encore risqué.

Dette technique : code pas toujours laissé très propre pour aller vite.

### **Tests**

Dès le départ : beaucoup de test en TDD + BDD.

Méthode XP

Extreme Programming Explained de Kent Beck

L'intégration continue est un prérequis pour le Continuous Delivery.

Collective code ownership : un développeur doit pouvoir intervenir partout dans le projet.

Pair programming : regards bizarres. Les arguments initiaux n'ont pas fait mouches auprès de l'ancien manager.

Le nouveau manager y est très favorable.

Pilote + Navigateur.

Le clavier change régulièrement de mains.

Objectifs : amener l'équipe de dev à partager la connaissance fonctionnelle, technique. Ils doivent se mettre d'accord => pas de code à l'arrache.

La valeur produite est celle que les développeurs n'ont pas en commun.

Erreurs fonctionnelles et techniques divisés par 10 si pas de recouvrement :

Hypothèses : erreurs que le pilote ferait seul : 10%

erreurs que le Navigateur ferait seul : 10%

KPI ?

La vélocité est meilleure : Before pair programming : 0.54 / After : 0.73

## **La qualité au rendez-vous**

Plugin Sqale sur SonarQube  
Unit Tests Coverage de 70%

## **Le déploiement**

Plusieurs types d'ops : admin système, équipe de support fonctionnel, QA, sécurité, DBA

Besoins différents : click bouton, scripts shell, Jenkins

Outil de déploiement idéal : très riches.

Les outils mis en place au début à base de shells n'étaient pas utilisés par tout le monde.

Utilisation de l'outil Deployit de Xebia.

Artefact (binaire + conf) préparé par Jenkins et livré par Deployit.

Deployit pousse la base, le tomcat et la conf.

## **Bilan**

### Gains :

Répétabilité, Gain de temps, Déploiements moins risqués

### Points négatifs :

Maven => version SNAPSHOT déployée sur l'environnement de dev

Process manuel : job Jenkins de déploiement exécuté manuellement. Lorsque le déploiement échoué, ils ne s'en rendent pas compte tout de suite

Prod vs Dev : perte de temps

Manque d'environnement pour tester les user story.

## **Changements**

Utilisation de Git à la place de SVN

Maven est conservé : numéro de version généré par le SHA-1 du commit pushé dans Git.

Vrai environnement pour tester le déploiement avec un build failed si le déploiement échoue.

## **Effets bénéfiques**

Time to Market

Amélioration de la qualité

Non-Event release

Autonomie des maitrises d'ouvrage pour mettre en place leur environnement

Kill bureaucratie : exigences demandées par les régulateurs. La chaîne automatisée fournit un audit complet et a donc allégé la charge de travail liée à cette tâche.

## **Conclusion**

Aucun zéro incident majeur de production

Le DSI s'est intéressé à cette initiative et va l'étendre sur le reste du SI.

Le mur entre la prod et les dév commence à céder. Encore beaucoup d'équipes. Aimerais tendre vers des équipes pluri-disciplinaires. Des pizza teams => équipe réduite de moins de 10 personnes.

### **Tests**

Tests sur des environnements ISO-PROD.

A la SG il y'a bien de tels environnements mais qui ne sont pas encore tout à fait similaires.

### **Côté Geek à la SG**

BBL

Session Dojo / Kata

Les décideurs parlent de Software Craftmanship, de pair programming

Partenaires Devvxx

Gros chantiers HTML 5 avec du Angular