

## Exemple d'Application des Méthodes Agiles au Développement d'un Produit Software

**CETIC, 21 avril 2004**

*Jean-Marc Bodart, Océ Software Laboratories Namur SA*



April 21, 2004

### Plan

- Les sociétés Océ et Océ Software Laboratories Namur SA
- Le projet PRISMA satellite Framework
- Héritage méthodologique
- Méthodes agiles
- Application des méthodes agiles au projet
- Conclusions préliminaires

## Groupe Océ

Océ est l'un des principaux fournisseurs de matériels, de services et de solutions complètes destinés aux entreprises et aux professionnels pour l'impression et la gestion de leurs documents, notamment dans les environnements traitant d'importants volumes.

- Siège social à Venlo (NL)
- 125 ans
- 80 pays
- 22 204 personnes



- ✓ C A 2003 : 2,8 milliards €
- ✓ Bénéfice net 2003 : 61,5 millions €

## Groupe Océ: Recherche et développement

7,7 % du CA sont consacrés à la R&D.

1900 employés travaillent dans le secteur de la R&D.

500 de ceux-ci sont impliqués dans le développement de software

7 centres de R&D software à travers le monde:

- Cleveland
- Créteil
- Konstanz
- *Namur*
- Phoenix
- Salt Lake City
- Tokyo



## Océ Belgium

- Filiale pour la Belgique et le Luxembourg
- 366 employés
- Chiffre d'affaires : 89.5 millions d'€



## Océ Software Laboratories Namur: Historique

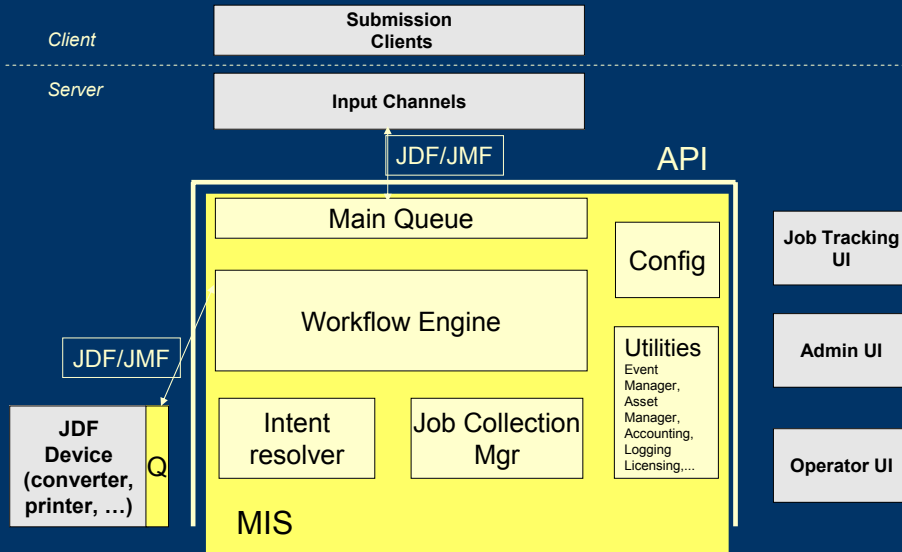


- Au départ: BS2000 pour la maison mère de Siemens à Munich.
- BS2000 → systèmes 'ouverts': logiciels d'application sous UNIX.
- Services de consultation et de gestion des projets; centre de compétence dans le stockage massif de données.
- Ouverture aux technologies Internet.
- OSL: Définition, développement et assurance de qualité, support pré- et post-sales et services professionnels liés aux softwares de gestion d'impression, d'archivage et restauration de données, gestion de flux de documents et traitement de données en provenance de systèmes ERP (Enterprise Resources Planning).

## Le projet: PRISMA satellite Framework

- Ensemble de composants réutilisables pour la construction d' "Output Management Systems"
  - Serveur impression traditionnelle office/DPC/EDP
  - Serveur impression large format
- Moteur de workflow articulé autour d'un standard de job ticket: JDF (XML – publié par CIP4)
- Nouvelles technologies pour OSL en 2003:
  - Microsoft .NET
  - Interfaces Web services
- Fortes contraintes au niveau délais
- Développement de la partie applicative en parallèle avec le Framework, avec un léger décalage

## Le projet: PRISMA satellite Framework (2)



## Héritage méthodologique

- Développement produit vs développement projet
  - Pas de contact direct avec le client
  - Longue durée de vie (parfois > 10 ans), versions successives
  - Assurance qualité primordiale pour couvrir les cas d'utilisation très diversifiés
- Héritage Siemens
  - "Methode Handbuch": méthodologie en modèle "waterfall"
  - Milestones intermédiaires dans la vie du projet: requirement study, solution study, interface specification, global design (+review), detailed design (+review), coding (+ review), component test, functional tests, system tests.
  - Méthode formalisée de review inter-département (R&D, Mkg, Svc – Development Document Control)
  - Long cycle de développement
  - QA succède au développement
  - Beaucoup de documentation avant le coding

## Méthodes agiles

- Plan de projet itératif, incréments tous les 1 - 3 mois
- Client décide du contenu fonctionnel des itérations et des priorités
- Représentant du client disponible sur le site de développement
- Intégration permanente, assemblage et test quotidiens
- Simplicité du code (au prix de refontes si nécessaire)
- Mise en production des itérations
- Unit tests et tests d'acceptance automatisés
- Pair programming (review permanent à la création)
- Délais sacrés, mais pas d'heures sup'

## Application des méthodes agiles au projet

- Choix pour méthodes agiles justifié par
  - Perspective de productivité plus élevée
  - Minimisation des risques technologiques
- Principes appliqués:
  - Embryon fonctionnel très tôt, dérivé d'un prototype
  - Planification d'incréments mensuels de fonctionnalité croissante (fonctions critiques ou à haut risque d'abord), mais sans véritable discussion avec le "client"
  - Travail en sous-groupes (mais pas de pair programming)
  - Génération quotidienne et unit tests intégrés (Nunit)
  - Génération de la documentation intégrée au build quotidien (Together Software, NDoc)
  - Intervention de la QA dès les premiers incréments

## Conclusions préliminaires

- Pas encore de réflexion ou de conclusion formelle (développement encore en cours)
- Impressions "à chaud"
  - Un bon travail préparatoire de spécification reste nécessaire
  - Adéquat pour de petites équipes (ca. 10 personnes)
  - Tests automatiques rapidement limités par l'environnement nécessaire
  - Environnement stimulant pour les développeurs: défis à court terme
  - Evolution tangible entre les incréments, rassurante (et démontrable !)
  - Développement de produit
    - Responsable QA peut jouer le rôle du client
    - Beaucoup de synergie entre dvp et QA
  - Au début du développement, peu de clarté sur le contenu final et les coûts: problématique dans les rapports avec mkg produit
- Interrogation: coût comparé aux méthodes traditionnelles, à qualité finale égale ?