



XP Days Paris 2006

Convaincre pour la construction Incrémentale



Agenda

- **Présenter**

- Les enjeux de la construction incrémentale
- Comparaison par rapport aux approche classiques

- **Pratiquer**

- Études de cas concrets

Présenter

Convaincre pour la construction incrémentale

■ Convaincre pour quoi?

- Vous voulez installer une méthode agile dans votre organisation
- Les pratiques en place ne sont pas agiles, et vous devez obtenir l'aval de l'organisation avant même d'avoir le droit de faire un premier essai

■ Qui convaincre?

- Les managers MOE
- Les sponsors, utilisateurs et MOA

■ Comment convaincre?

- Éviter de long discours
- La pratique seule peut ouvrir les yeux

Maîtriser les risques techniques et fonctionnels

- **Un projet informatique est plein d'inconnues**
 - Des risques fonctionnels
 - Difficulté d'avoir des spécifications fonctionnelles détaillées couvrant tout le périmètre en phase de préparation
 - Des risques techniques
 - Capacité à absorber l'accroissement des volumes traités
 - Difficulté de mise en œuvre de la solution retenue
- **La solution d'une démarche itérative**
 - Une première itération courte
 - Un périmètre fonctionnel complet
 - On adresse déjà le cœur des problèmes techniques
 - Puis on avance de manière itérative
- **Il faudra ajuster le tir**
 - Accepter de modifier le résultat de l'itération précédente
 - La 1ere itération n'a pas pu tout voir, certains problèmes seront découverts en cours de route
 - Il sera sûrement nécessaire d'ajuster en cours de route
 - **Mais attention! Il faudra éviter l'empilement de fonctions et de systèmes**

**Le dialogue étroit entre tous les acteurs
est indispensable pour adresser tous les risques**

Du rêve à la réalité

2000 J/H



1000 J/H

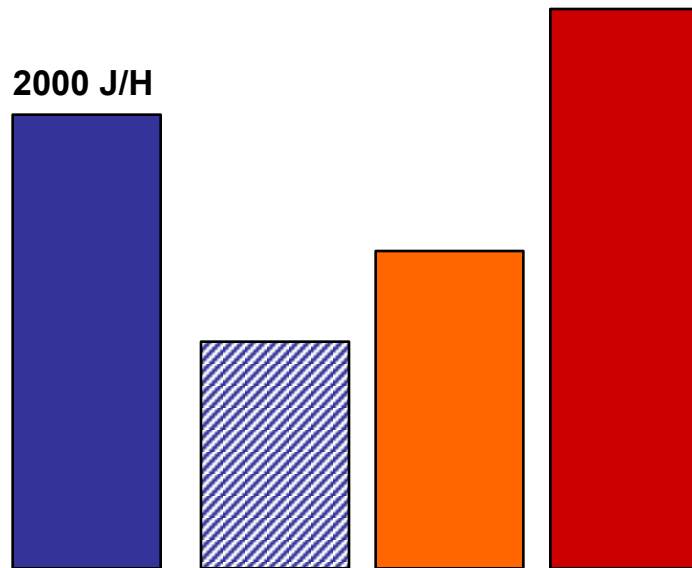


- **Le besoin initial, sans contraintes**
 - Supposons un projet sur lequel tous les besoins seraient cristallisés
 - Estimons que ce projet coûterait 2000J/H
- **« A budget illimité, et sans contraintes, tout est possible. »**

- **Dans la pratique, le budget est de 1000 J/H**
 - On n'a jamais assez de moyen pour réaliser ses rêves
 - Il est donc sain, et souhaitable, de se fixer des limites réalistes
 - **Le plus important est de connaître ses limites, et de ne pas les dépasser.**

Comment faire rentrer 2000 dans 1000?

Deux réponses à la dérive des budgets

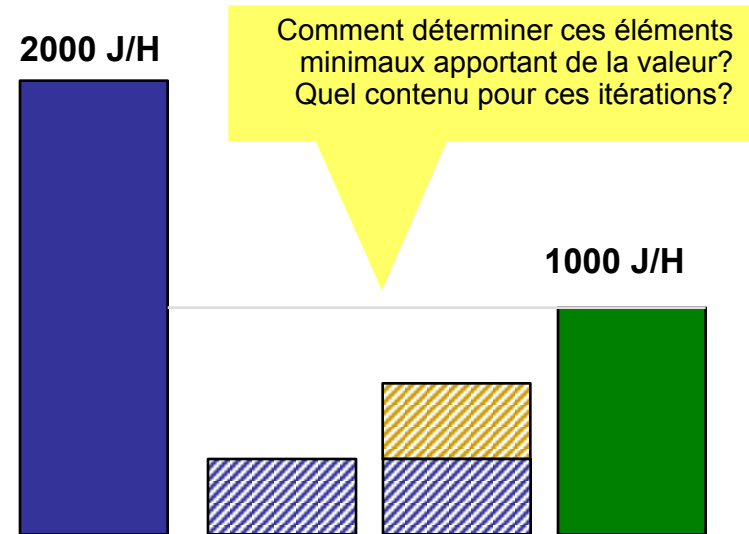


■ L'approche « classique »

- Conception détaillée de toute l'application
- Evaluation des coûts de chaque fonction
- Arbitrage pour en prendre pour 1000
- Et on lance les développements!

■ Dans la pratique...

- On se trompe parfois (souvent?) sur les évaluations
- Et les développements sont parfois (souvent?) plus compliqués que ce qu'on prévoyait



■ L'approche agile

- Délivrer de la valeur pour un budget très limité
- Ajouter itérativement des fonctionnalités, s'assurant à chaque fois que cela apporte de la valeur utilisateur
- S'interdire de dépasser le budget
 - « Je vous en fait pour le budget que vous me donnez »
- Le projet s'arrête lorsque le budget est atteint

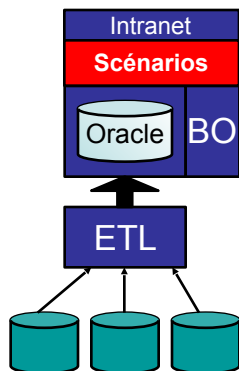
Conception incrémentale

- **Concevoir une architecture simple et incrémentale**
 - Nécessite de très bien connaître le métier
 - On pourra s'inspirer des bonnes idées des applications déjà réalisées
 - Adapter la solution technique aux exigences réelles de performance et de volumétrie
 - **Le dialogue étroit entre informaticiens et MOA/Utilisateurs est INDISPENSABLE**
 - Une connaissance approfondie et commune du fonctionnel et de la technologie est nécessaire
- => Faire un projet en mode agile est plus difficile qu'en mode classique**

Exemple d'architecture incrémentale

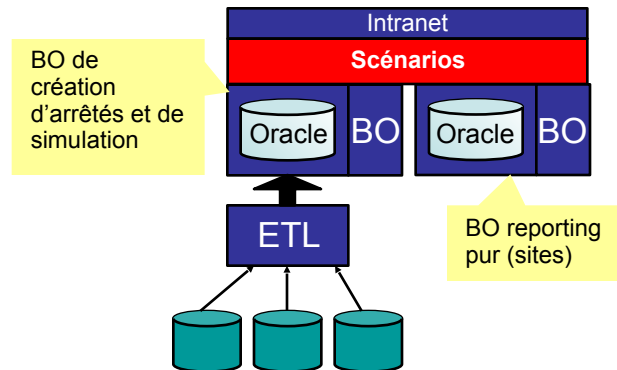
- **Métaphore d'une application décisionnelle: Atelier de création d'arrêtés Bale II**
 - L'application est conçue pour être capable de gérer des scénarios de création d'arrêtés
 - Un scénario = liste des fichiers sélectionnés+ajustements manuels+liste des états Bâle II produits pour la Commission Bancaire (XBRL)
 - **L'essentiel de la valeur (ie les développements spécifiques) se fera sur la mécanique de gestion des scénarios**
 - L'architecture technique est adaptée à ce besoin en fonction des contraintes de performance et de volumétrie

Archi initiale



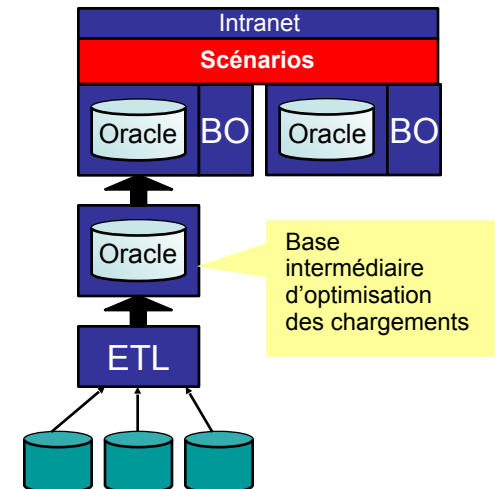
- **Un ETL pour transformer et charger les fichiers**
- **Une base Oracle unique**
 - Toutes les données chargées et les corrections sont stockées dans une base unique
- **Business Objects pour faire des requêtes**

Lot 2



- **Ajout d'une brique reporting pure**
 - Seulement si les consultations deviennent critiques
 - Par exemple, si besoin de connexion de nombreux utilisateurs simultanés

Lot 3



- **Découpage de la chaîne de chargement**
 - Seulement si les consultations deviennent critiques
 - Si les volumes en entrée deviennent très importants

Pratiquer



Bernard Notarianni
<http://www.octo.com>