

Le modèle logique en interface

18-juin-2012

Contexte

- **April WAF est une filiale jeune du groupe April**
 - 3 ans, ~10 collaborateurs
- **Développe une plateforme pour l'assurance**
 - Tournée vers le e-commerce
 - Cible l'ensemble de la gestion d'assurance
- **Remporte en 2009 le projet de refonte du système informatique métier d'une filiale du groupe**
 - Remplacement d'un système vieux de 20 ans
 - De la vente du contrat à la **gestion du sinistre**

Nécessité de faire évoluer nos méthodes
et de trouver des partenaires

- **Le partenaire : maîtriser la solution**

- Proximité
- Habitudes de travail communes
- Confiance



Clever Age,
réceptif à la
méthode

- **La méthode : répondre aux attentes**

- Périmètre complet
- Orientée SOA
- Rigoureuse
- Progressive



Praxeme

- **Le coach**

- Méthode & modélisateur



Fabien Villard

SFD

- **Clients**
- **April WAF (rôle de MOA)**
- Assistance par notre partenaire (CP + maquettiste)

Modèle
sémantique

- **Modélisateur April WAF**
- Relecture partenaire

STD & Dèvs

- **Partenaire**
- Validation April WAF

- **Produire le modèle sémantique**
 - D'après l'analyse du système existant et des SFD
- **Trouver les compromis pour appliquer progressivement Praxeme**
 - En répondant aux contraintes de planning
 - En intégrant les retours de notre partenaire
- **Valider fonctionnellement les développements**
 - Revue de code
 - Couverture de tests
 - Recette

Le modèle sémantique : un socle commun

Etablissement du modèle sémantique

- **SFD utilisées en « matière brute »**
 - Attentes précises du client en terme d'IHMs
 - Pas de remise en cause des IHMs à partir du modèle sémantique
- **Utilisation des OBC et exemples des documents Praxeme**
 - Domaine réalité : reprise importante des OBC
 - Domaine sinistre : inspiration pour initier le modèle
- **Travail et validation avec le responsable STD & Dèvs**
 - Challenge le modèle
 - Vérifie qu'il dispose de tous les éléments pour les développements

Niveau d'aboutissement du modèle sémantique

- **Focus sur les éléments les plus structurants**
 - **Liste exhaustive des règles métier / organisation**
 - Généralement issues des SFD
 - Parfois identifiées lors de la modélisation
 - **Modèle complet de classes**
 - Attributs, associations, opérations
 - **Machines à états**
 - Distinguant les états sémantiques des états du “processus”
- **Raccourcis accélérateurs**
 - Pas purement sémantique
 - Opérations peu détaillées

Diagramme de classes

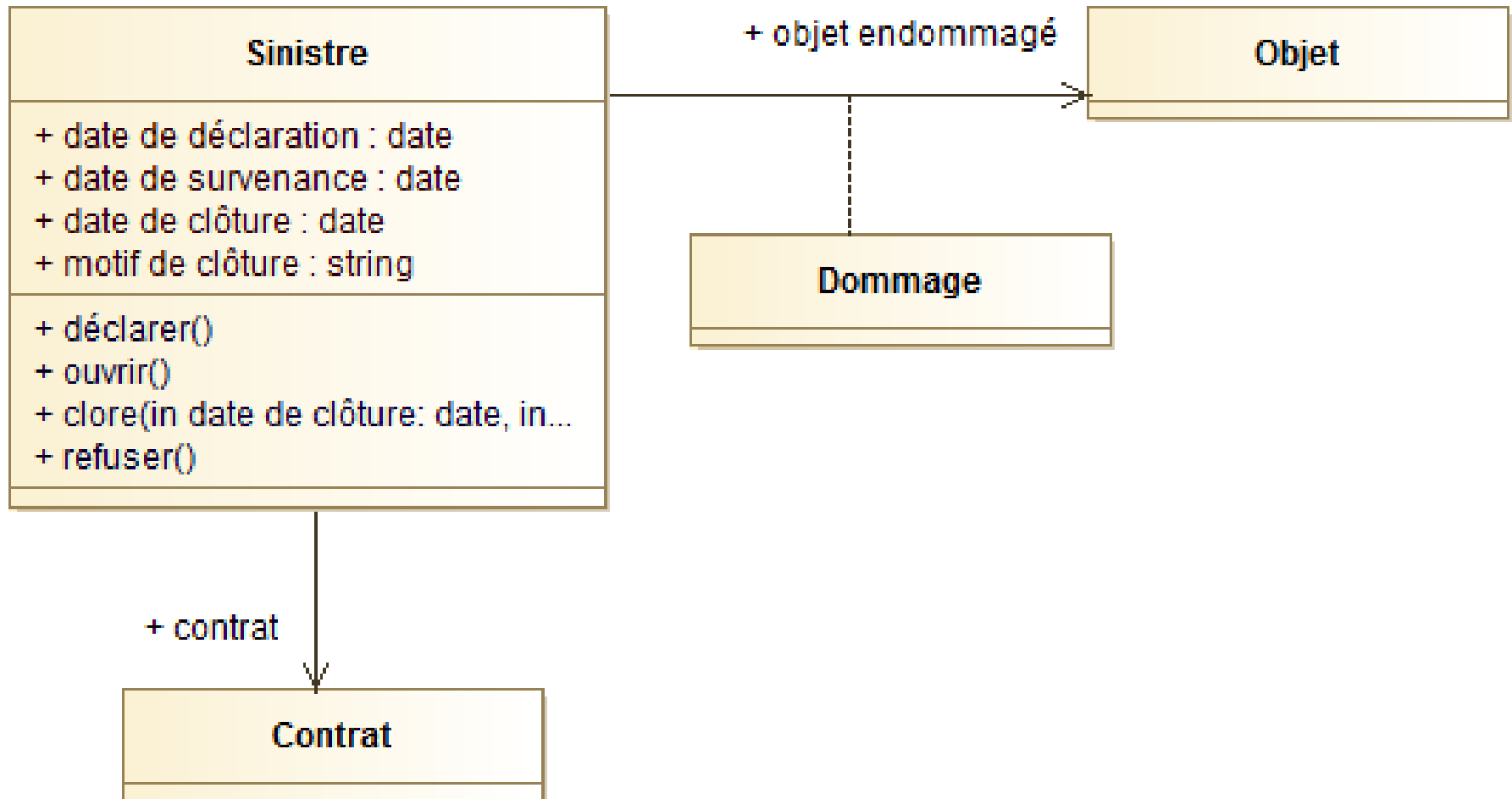
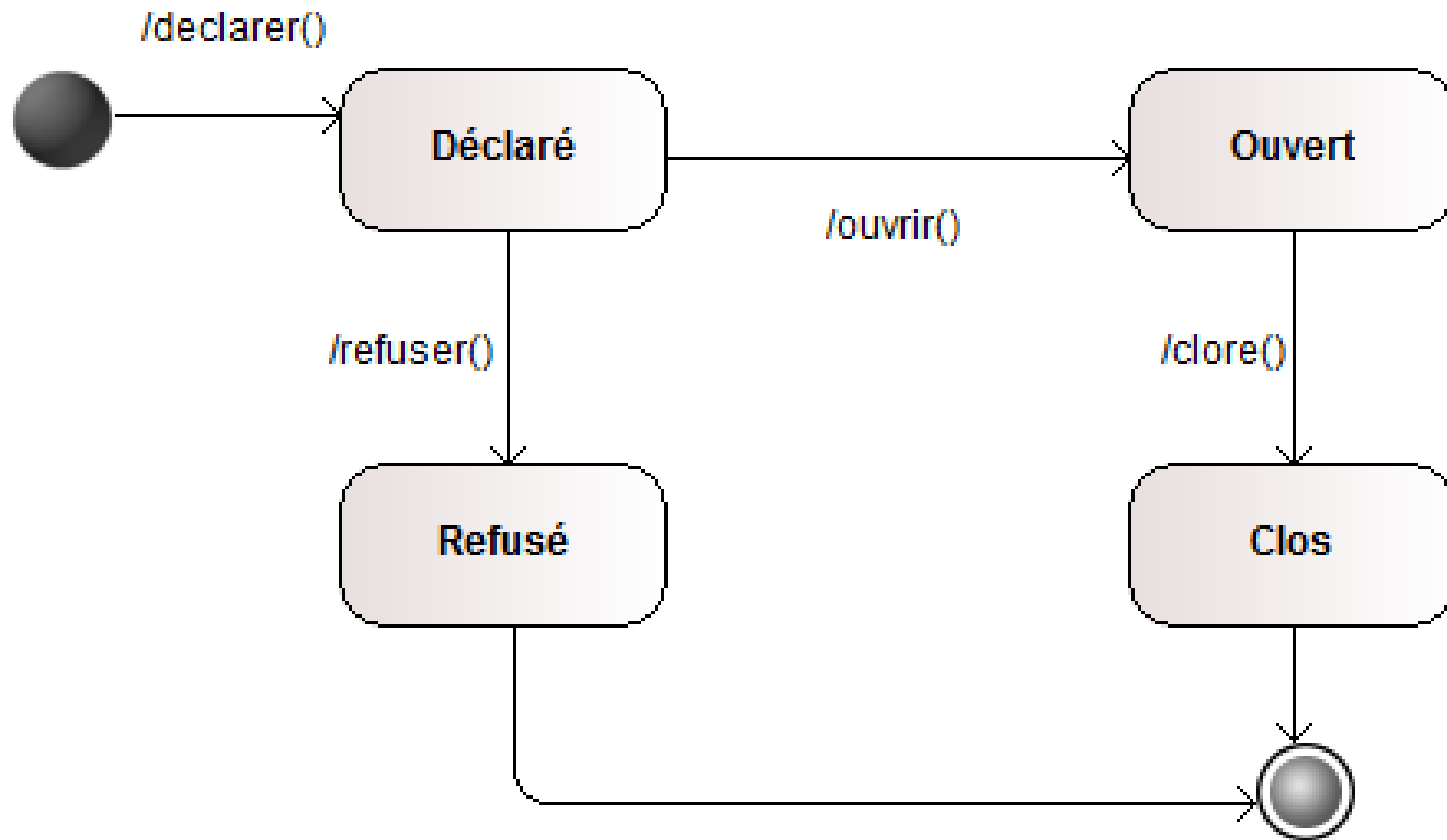


Diagramme d'états

- Lien entre le cycle de vie et les opérations de la classe métier
- Enrichissement : communication entre les objets par évènements



Apports & difficultés du modèle sémantique

- **Compréhension du métier**

- **Analyses d'impact et de transformation**
 - Suivant l'ampleur des évolutions demandées
 - « Hors du code »

- **... mais des difficultés de communication**
 - Équipes peu familières avec ce type de formalisation
 - **Ces difficultés s'atténuent avec la pratique**
 - Utilisation de diagrammes d'objets
 - Explications textuelles dans un vocabulaire compris par l'interlocuteur

- **Pas de modélisation pragmatique rigoureuse**
 - Cas d'utilisation et processus définis avec les utilisateurs
 - Pas de remise en cause des écrans ainsi définis
 - Choix que l'on retrouvera dans le modèle logique

- **Progressivité dans l'application de la méthode**
 - **Non sans impacts ...**

Le modèle logique : interface pour l'externalisation



- **Partenaire**
- Validation April WAF

- Qui est responsable du modèle logique ?
- Comment utiliser le modèle logique comme STD ?
- Comment utiliser le modèle logique en validation des livrables ?

- SFD et modèle sémantique communs avec le partenaire
 - Responsabilités claires
 - Modèle complet et bien assimilé des deux côtés
- **STD sous la responsabilité du partenaire mais avec contrainte de validation par April WAF**
 - Notre demande : obtenir une architecture Praxeme

- **Compromis – responsabilité partagée du modèle logique**
 - Définition d'un modèle logique "macro" par April WAF
 - Approfondissement en commun

Architecture logique en strates



Premier pas - Ébauche de modèle logique

- **Concentrée sur la strate métier**
 - Plus structurante
 - Mieux maîtrisée
 - Choix lié aux raccourcis sur la modélisation amont (pragmatique)
- **Strate organisation liée aux écrans**
 - Interlocuteurs davantage techniques
 - Equipe restreinte

- **Raccourcis accélérateurs**
 - Diagrammes d'ensemble
 - Liste des machines logiques et des services, sans spécification détaillée

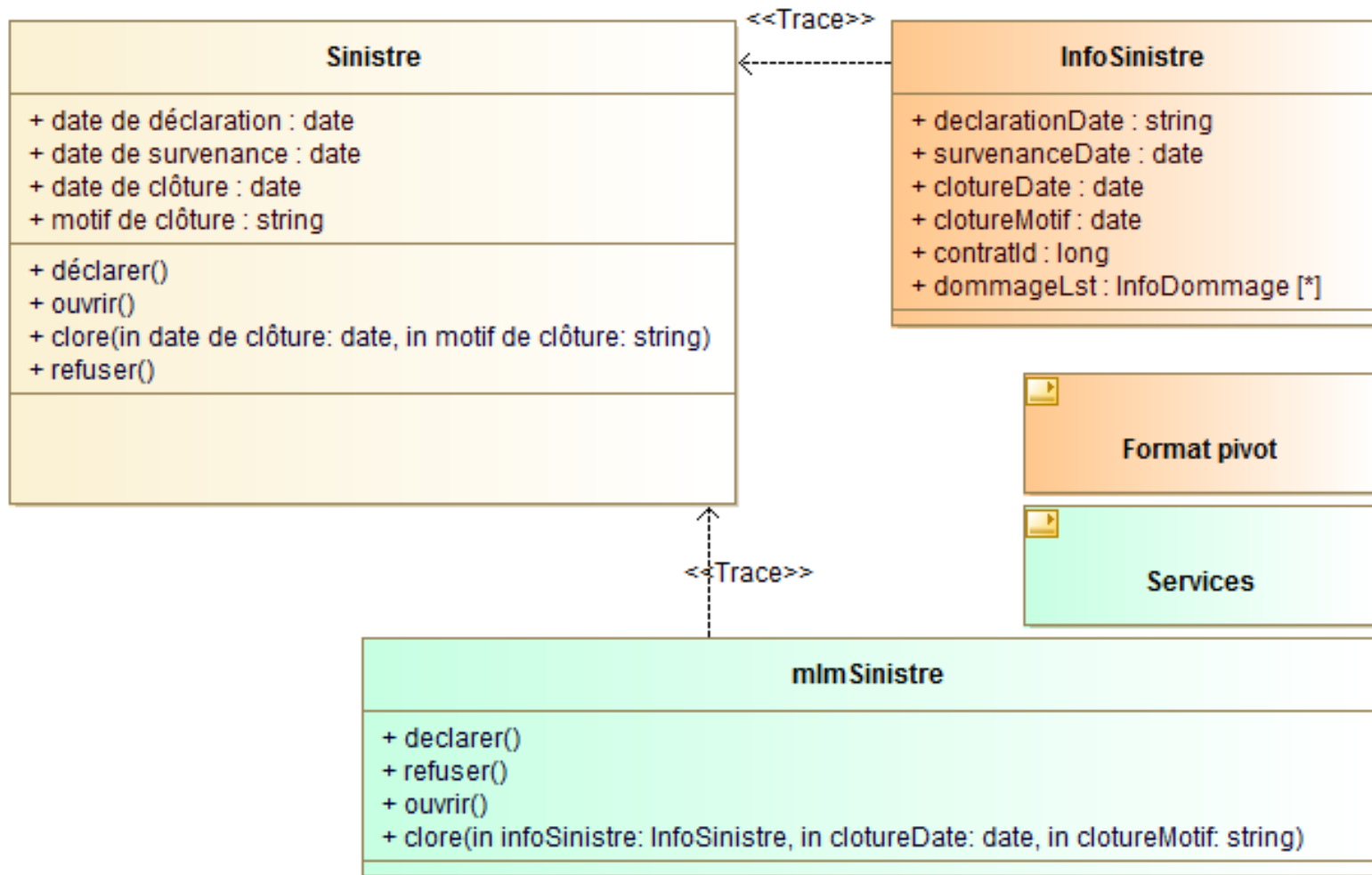
Extrait FML04 – Architecture logique



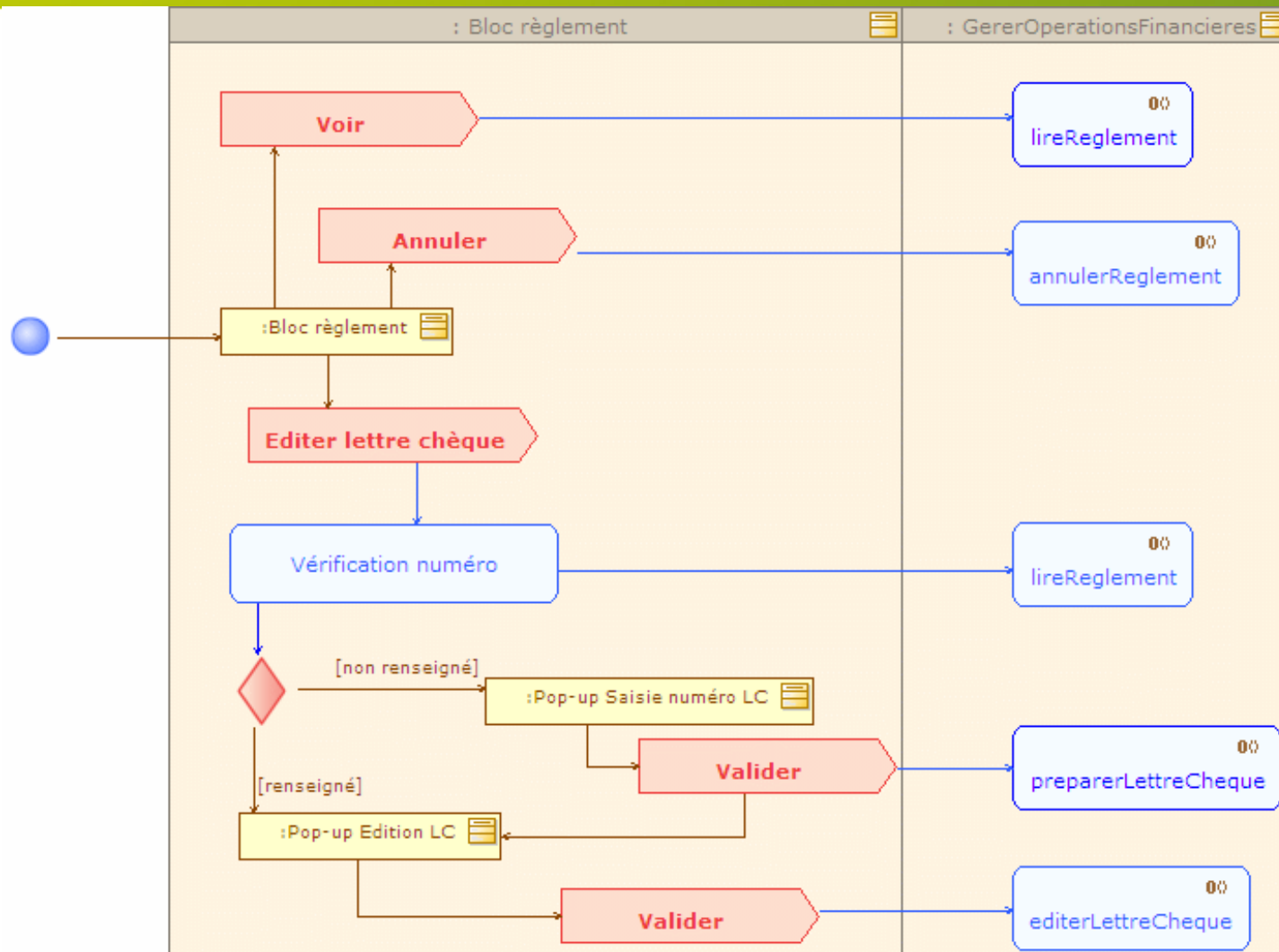
- **Utilisation du modèle d'ensemble en macro chiffrage**
- **Pour chaque sprint*, carottage sur les classes**
 - Définition du périmètre fonctionnel par la MOA
 - Spécification précise des services à développer
 - Jusqu'aux diagrammes d'activité si nécessaire
- **Diagrammes d'activité pour les écrans complexe**
 - Inspirés du livre SITA
 - Pour fluidifier l'intégration avec l'IHM

*Développements en mode agile

Exemple de dérivation sémantique => logique



Exemple de diagramme d'activités "Ecran"



Apports du modèle logique & des règles de dérivation

- **Structuration des développements de la couche métier**
 - Bonne cohésion d'ensemble
 - Tout en intégrant les évolutions des demandes
 - Bonne maintenabilité

- **Simplification du suivi d'avancement**
 - Dérivation d'une "check-list" depuis le modèle
 - Correspondance des services avec les IHMs des SFD

- **Des apports indéniables**
 - Avoir une interface rationalisée avec le sous-traitant
 - Avoir et maintenir une bonne vision du système
 - Avoir une validation objective des livrables
- **Des difficultés**
 - Vocabulaires à partager
 - Intégration avec les habitudes de travail
- **Un investissement différent**
 - Travail en amont plus important
 - Travail en aval simplifié
 - Améliore les niveaux de garantie

Utilisation du modèle logique en interne

Réécriture d'un module existant

- **Des contraintes qui semblent moins fortes**
 - Proximité immédiate => facilitateur
 - Absence de contractualisation => aspect informel
- **“On s’arrangera au fil de l’eau”**
- **Résultats**
 - Travail en commun amont moins important avec les développeurs
 - Difficultés à faire passer la méthode
 - Zones de flou sur les interfaces entre modules

La proximité ne contrebalance pas les défauts d'un manque de méthode

Évolutions et perspectives

Un projet vivant, une pratique qui évolue

- Construction d'une plateforme métier multi-sociétés
- Partenariat avec une SSII apportant ressources et expérience complémentaires
- Appropriation graduelle de la méthode en interne
- Travaux de rapprochements avec la « cartographie fonctionnelle » : construction d'un lien avec l'architecture métier

Une nécessité de s'adapter et se rapprocher de
l'architecture d'entreprise

