Sortir des obsolescences

CRÉDIT AGRICOLE S.A.

Garder un coût de MCO acceptable

- **Les applications qui rendent un bon service, mais dont on a perdu la connaissance MOE et MOA, nous coûtent peu tant que rien n'évolue.**
  - **Peu d' évolution fonctionnelle des applications règlementaires, car souvent les nouveaux règlements nécessitent de nouvelles applications.**
  - **Optimiser les investissements informatiques, garder le plus longtemps possible les applications , mais également les matériels.**
  - **Réduire dans les coûts la part de la Maintenance en Condition Opérationnelle, pour investir au maximum dans les nouveaux développements (plus value Métier).**

Facteurs de vieillissement des applications

Les facteurs de vieillissement du parc applicatif :

- **Les applications métiers :**
  - pas d'évolution fonctionnelle sur dix ans, voire plus (exemple : FICP), dégagement des ressources vers les nouveaux projets
- **Les applications d'infrastructure**
  - Evolution rapide des équipements et des structures des réseaux (intranet internet, cloud)
- **Les équipements (Hard & Soft)**
  - Evolution des OS évolution des matériels de stockage (exemple K7 zOS)
- **Les outils informatiques**
  - Evolution des technologies de développement et des méthodes (Objet, UML, client léger))
- **L évolution de 'état de l'art**
  - Faire les nouvelles applications avec les nouvelles technologies (Garder un ROI acceptable)
- **L'évolution sociétale**
  - L'informatisation de notre vie publique (décalage entre l'interne et l'externe)

Pilotage des obsolescences :Cohérence des calendriers difficile à réaliser

- Plusieurs vues :
  - Plan et budget des applications métiers
  - Plan des produits informatiques, roadmap des éditeurs
  - Inspection des directions , métier et informatique
  - Veille technologique
  - Formations et accompagnement des hommes

le contexte Crédit Agricole SA

- Pour les applications « Poste de travail », 500 applications sont homologuées, toute ou partie s' exécutant sur le poste de travail.
- 3500 postes de travail, 25 masters gérés en permanence gestion de l'obsolescence des nouveaux matériels par revue semestrielle
- Migration Windows XP vers Windows 8 en 2010 – 2011 dans la cadre du projet Evergreen (site de Montrouge) pour un coût d' environ 1000 jours (partie applicative)
- Les coûts prohibitifs de la migration des solutions postes de travail ,à chaque migration Windows, ont orienté le standard vers le client léger depuis plus de dix ans.

Zoom sur la partie NSDK

- Depuis 2001, arrêt des développements NSDK en raison ;
  - Des craintes sur la pérennité du fournisseur
  - De la cible d'abandon des clients lourds
  - Du début de pénurie de ressources (internes et externes) maîtrisant NSDK
- La cible est le développement JAVA en client léger (pas de java sur le poste) , mais les équipes peinent à monter en compétence, le Visual Basic est utilisé par défaut
- Crédit Agricole SA a une dizaine d'applications en NSDK , Natstar et VB représentant 170 000 lignes de code
- Demandes récurrentes des équipes pour trouver une solution outillée pour aller vers la cible JAVA

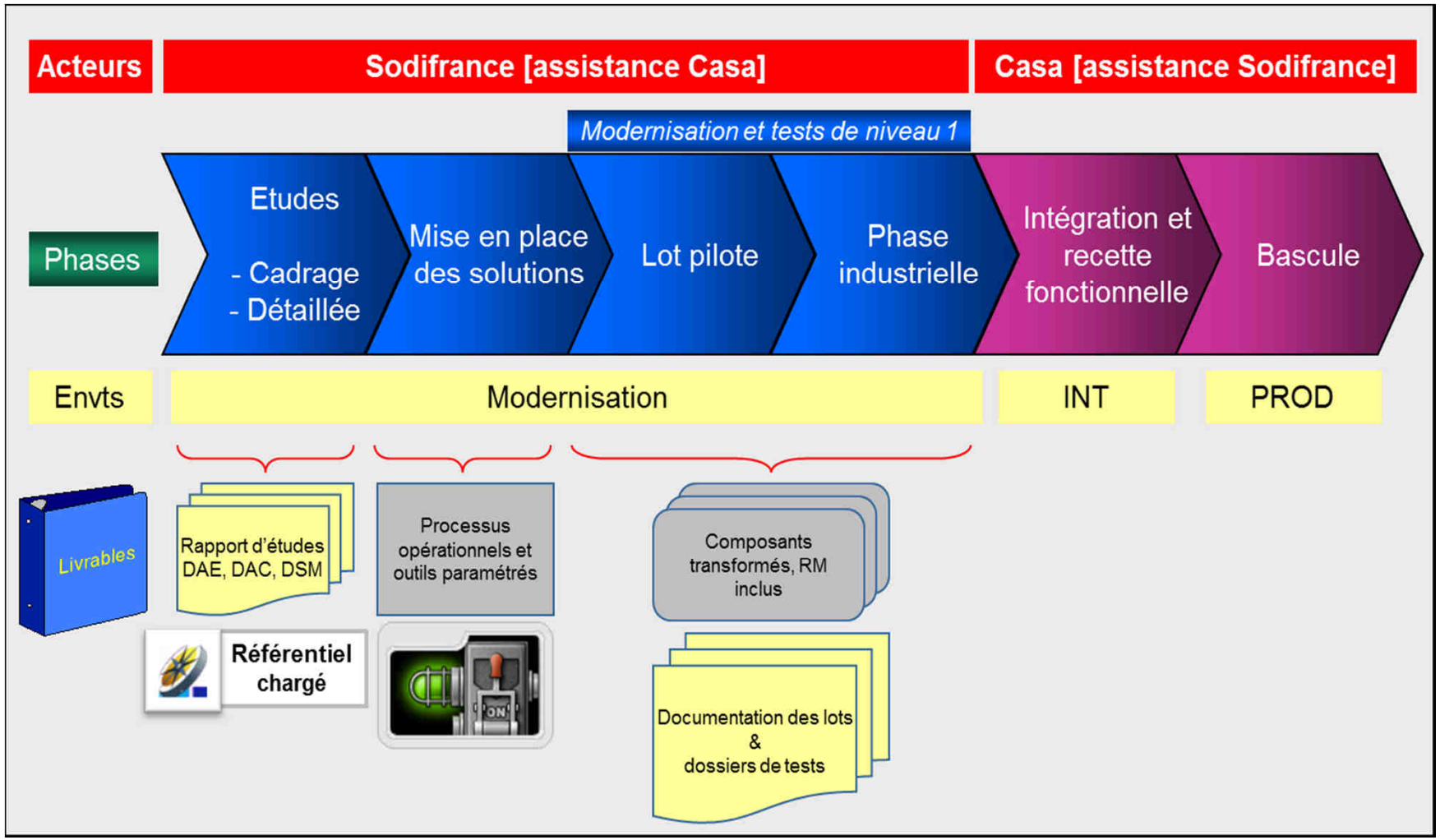
Arguments pour faire une migration industrielle

- Les applications n'évoluent pas ou très peu du point de vue métier (prêts bonifiés, rétrocessions des bonifications …)
- Les équipes ne connaissent plus le point de départ et pas encore la cible JAVA
- Les ressources coté Métier ont presque disparues
  - Spécifications obsolètes ou manquantes
  - Plus aucun référent sur le sujet
- Le parc à migrer est suffisant pour compenser le coût d'une migration industrielle (Coût de la première Phase)
- La migration industrielle permettra de reconstituer des jeux de tests fonctionnels, et les spécifications (en partie)
- L'obsolescence est un risque qui s'aggrave
- Le coût est moins élevé qu'un nouveau développement

Arguments pour faire une migration industrielle par les modèles

- La migration de code à code n'est pas adaptée :
  - La structure des applications NSDK et VB transposée en JAVA, donnerait un code peu maintenable, il faut changer de modèle pour avoir des applications répondant aux normes de développement JAVA
- L'utilisation des modèles va permettre d'atteindre la cible définie de manière industrielle (fiabilité, garantie)
- Des tests fonctionnels de référence manquants pourront être obtenus ainsi que la documentation UML associée aux nouvelles applications ainsi créées
- Les points « durs » (bloquants) de la migration pourront être mis en évidence avant le début de migration et leur résolution identifiée en début du projet
- Obtenir des applications à iso-fonctionnalité et iso- IHM

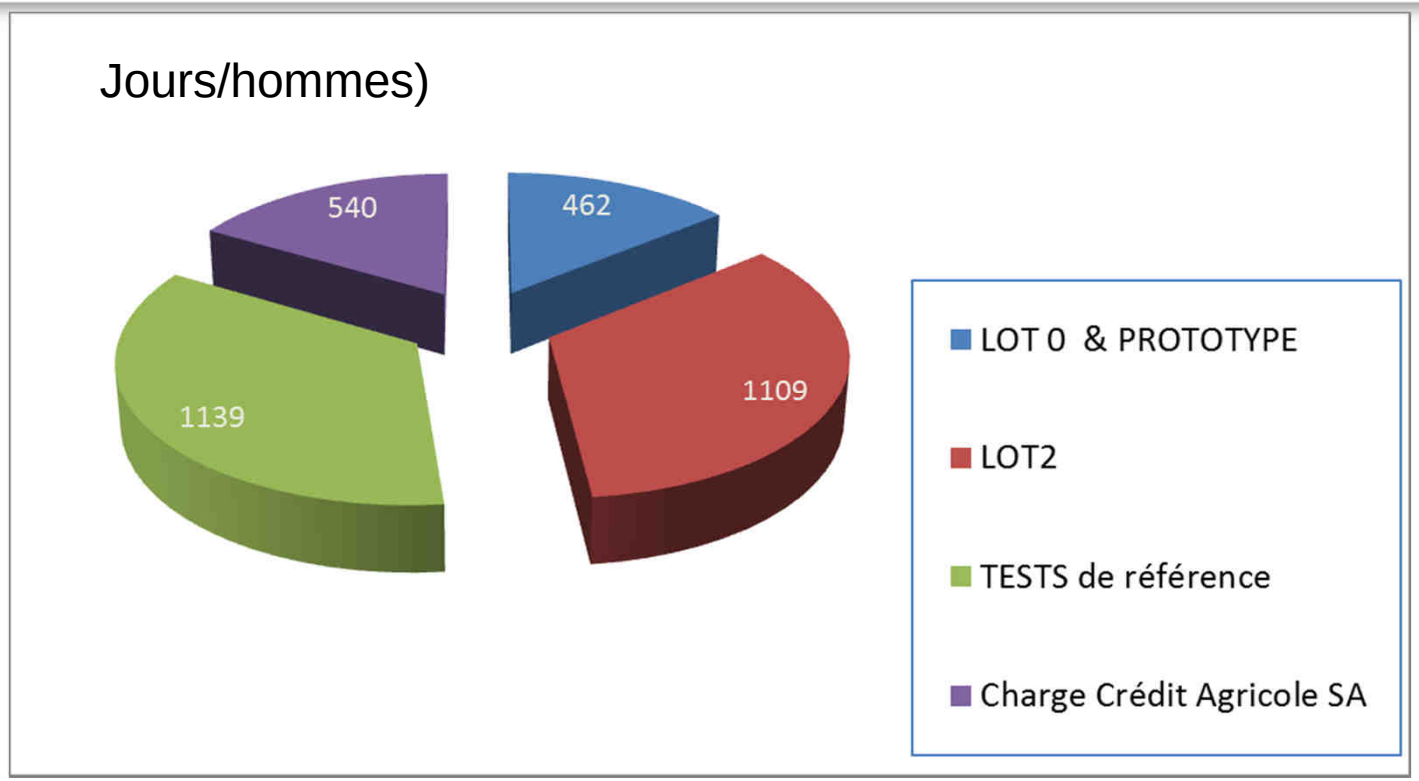
plan de migration industriel



Exemple des points « durs »

- Les applications Natstar utilisent le module IM. Ce module peut générer le schéma de la base à partir de description de tables, mais il nécessite la mise en place de tables techniques.
  - Comme les bases sont utilisées par d'autres applications, elles ne doivent pas contenir ces tables techniques. Cela implique également qu'il ne sera plus possible de modifier les schémas des différentes bases.
- Dans le code on trouve des éditions Impromptu, composant distribué par Cognos permettant de faire de requêtes utilisateurs directement sur les tables DB2 et des éditions faites par le composant Nat System NS-Report.
  - Les éditions NS-Report devront être portées vers le composant BIRT. L'utilisation d'Impromptu ne répond pas à un besoin particulier, les éditions correspondantes devront également être portées vers BIRT.

Avoir un stock suffisant



La démarche outillée pilotée par les modèles requiert un minimum d'applications à prendre en compte dans le projet.
En effet la rentabilité de cette démarche suppose un stock à migrer, car l'étude et le paramétrage (lot 0 et prototype)sont des étapes analytiques trop coûteuses si le stock à migrer n'est pas suffisant.

Perspectives

- Les possibilités offertes par ce refactoring piloté par les modèles auraient très bien pu s'appliquer dans le cadre d'un projet de ré-urbanisation de ces applications.
  - Pour définir une couche métier bien séparée, pour déplacer les applications dans la géographie de leur situation dans le système d'information (la « réseau graphie »).
  - Pour élargir leur utilisation dans un découpage en services utilisable dans un système orienté SOA
  - …
- Et si le refactoring systématique vers un dictionnaire de modèles devenait le modèle permanent pour gérer le parc existant des nombreux AGL devenus caducs par les lois bizarres, ou cruelles du marché ?
  - Il ne resterait de l'obsolescence à gérer que le déplacement vers un autre modèle plus récent à choisir dans les possibilités d'implémentation du système, présent dans le dictionnaire des modèles cibles possibles, en somme un référentiel de modèles.

Perspectives

- Ce référentiel idéal contiendrait d'une part les modèles issus du parc existant ou d'un parc informatique que l'on absorbe (suite à une fusion d'entreprise par exemple…), d'autre part les modèles retenus et applicables dans le système d'Information que l'on gère, et surtout les modèles futurs.
- La gestion de notre existant, si lourd à porter parfois, pourrait devenir séduisante, et l'agilité de l'intégration de services devenir enfin notre réalité quotidienne.

Merci