



Université du Québec


# École de technologie supérieure

Département du génie logiciel et des T. I.


Travail de session

Comparaison des modèles organisationnels en maintenance

Étudiant	Houcine Jeljeli
Contact	
Codes permanents	JELH27068309
Cours	MGL804
Session	HIVER 2011
Groupe	01
Professeur	Alain April
Chargé de cours	Carlos Teodoro Monsalve
Date	2011-04-21

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 2	PAGES 20

I.	Introduction.....	4
II.	Technologie .....	5
A.	Historique et présentation générale .....	5
B.	Architecture .....	5
1.	Présentation Générale.....	5
2.	Architecture trois niveaux .....	6
3.	Environnement Intégré de Développement .....	7
4.	Couches d'application.....	7
III.	Modèle de Maintenance .....	9
A.	Maintenance assurée par Microsoft .....	9
B.	Maintenance assurée par Sanimax.....	11
1.	Contexte technologique .....	11
2.	Méthodologie <i>Sure Step</i> .....	13
3.	Modifications d'ERP avant le <i>GO-LIVE</i> : Maintenance ou Développement? .....	14
4.	Maintenance après <i>GO-LIVE</i> .....	15
IV.	Discussion .....	16
A.	Observations.....	16
B.	Améliorations .....	17
V.	Conclusion .....	18
VI.	Références.....	20

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 3	PAGES 20

## Liste des figures


---

Figure 1: Architecture Générale <sup>[Ref.3]</sup> .....	6
Figure 2: Architecture 3 niveaux (3 Tiers Architecture) <sup>[Ref.4]</sup> .....	7
Figure 3: Couches d'application <sup>[Ref.5]</sup> .....	8
Figure 4: Outil de comparaison de <i>MorphX</i> .....	11
Figure 5: Diagramme de logiciels chez Sanimax <sup>[Ref.6]</sup> .....	12
Figure 6: Méthodologie <i>Sure Step</i> <sup>[Ref.7]</sup> .....	13

## Liste des tableaux

---

Aucune entrée de table d'illustration n'a été trouvée.

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 4	PAGES 20

## I. Introduction


Ce rapport consiste à faire une enquête afin «*d'Identifier le modèle organisationnel utilisé, en maintenance,*» dans mon organisation. Étudier «*les raisons sous-jacentes avec l'administrateur, et questionnez cette décision en fonction des avantages et des inconvénients des deux modèles organisationnelles,*» <sup>[Réf.1]</sup>

L'entreprise pour laquelle je travaille s'agit de *HAKA Solutions*, une compagnie de consultation qui est née du projet d'implantation du logiciel *Axapta* pour une compagnie de fabrication en 2001, suite à cette implantation l'entreprise s'est spécialisée dans l'implantation d'*ERP*.

Un *ERP* (*Enterprise Resource Planning*) est un «*Progiciel qui permet de gérer l'ensemble des processus d'une entreprise en intégrant l'ensemble de ses fonctions, dont la gestion des ressources humaines, la gestion comptable et financière, l'aide à la décision, mais aussi la vente, la distribution, l'approvisionnement et le commerce électronique.*» <sup>[Réf.2]</sup>

L'*ERP* de prédilection de *HAKA Solutions* est *Microsoft Dynamics AX* autour duquel la compagnie a développé son expertise et son centre de connaissance. La compagnie a contribué à l'amélioration du progiciel en lui intégrant de nouveaux modules dont le module Qualité que *Microsoft* a acheté pour en faire partie intégrante de la version standard de *Microsoft Dynamics AX*.

C'est dans ce cadre donc, que *HAKA Solutions* offre ses services à différents partenaires et clients, et à juste titre l'enquête et les analyses que je fais dans ce rapport ont été réalisés chez un des clients de l'entreprise pour lequel j'ai agi en tant que consultant de Mars 2009 jusqu'au mois de Mars 2011, il s'agit de *Sanimax inc* qui est une entreprise de quelques 1100 employés qui fait du recyclage de sous-produits animaux et des huiles de cuisson usées.

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 5	PAGES 20

*Sanimax* opte pour un modèle d'affaires basé sur la sous-traitance de ses services de support à sa structure informatique, donc mis à part l'équipe de gestion des projets il n'y a pas de ressources techniques à l'interne pour le support informatique.

## II. Technologie


### A. Historique et présentation générale

C'est en 1983, que la compagnie danoise *Damgaard* a commencé le projet de développement de l'*ERP Axapta*, mais c'est en 1998 que la compagnie a officiellement lancé son produit sur le marché, en 2002 *Microsoft* acquit la société *Navision* qui entre temps, en 2000, a fusionné avec *Damgaard*. *Microsoft* intègre *Axapta* dans la famille *Microsoft Dynamics* pour devenir *Microsoft Dynamics AX*.

### B. Architecture

#### 1. Présentation Générale

*Microsoft Dynamics AX* repose sur une plateforme riche et complémentaire, ainsi mise à part les composants principaux qui supportent le fonctionnement de base du système, la plateforme offre aussi des outils qui permettent d'interagir avec des logiciels et applications tierces.

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 6	PAGES 20

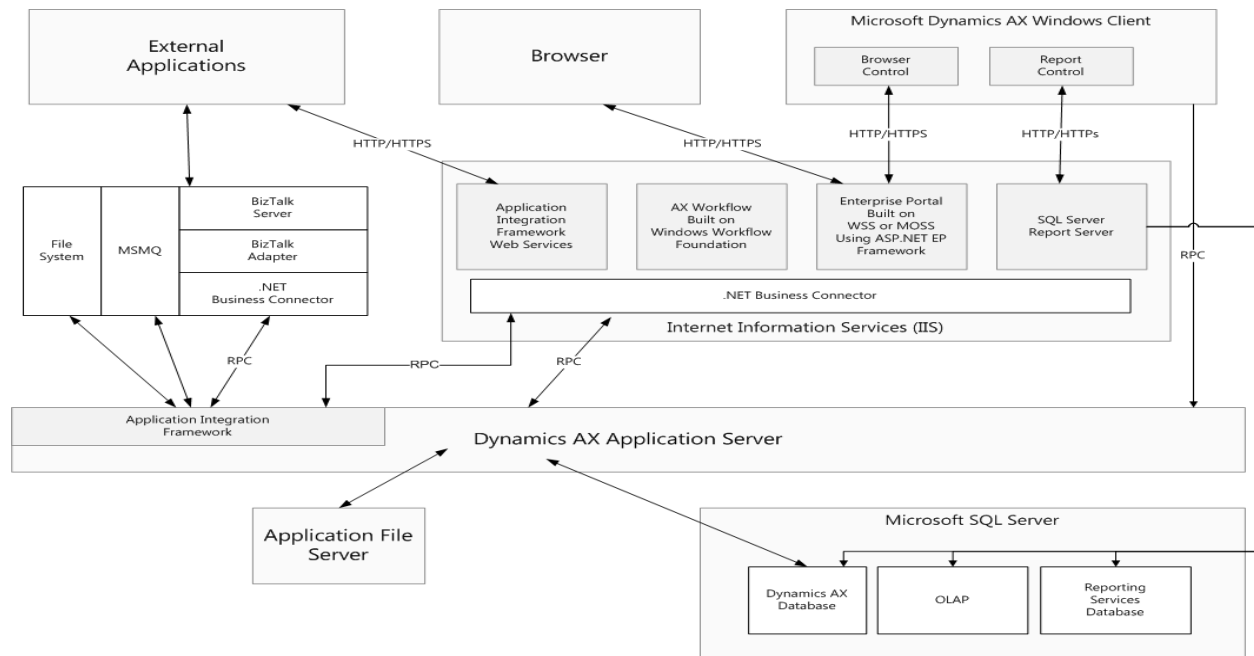



Figure 1: Architecture Générale <sup>[Ref.3]</sup>

Dans le cadre de ce rapport, on ne s'intéresse qu'aux composantes qui forment le noyau dur de l'ERP.

## 2. Architecture trois niveaux

*Microsoft Dynamics AX* repose sur une architecture trois niveaux; le premier niveau comporte le client riche qui gère l'interface utilisateur, le deuxième niveau se compose de l'*AOS (Ax object Server)* qui assure et gère les logiques d'affaires ainsi que le *AX business Logic (Application Layer)* qui répertorie le code source et enfin le troisième niveau représente la base de données.

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session	PAGE 7	PAGES 20	

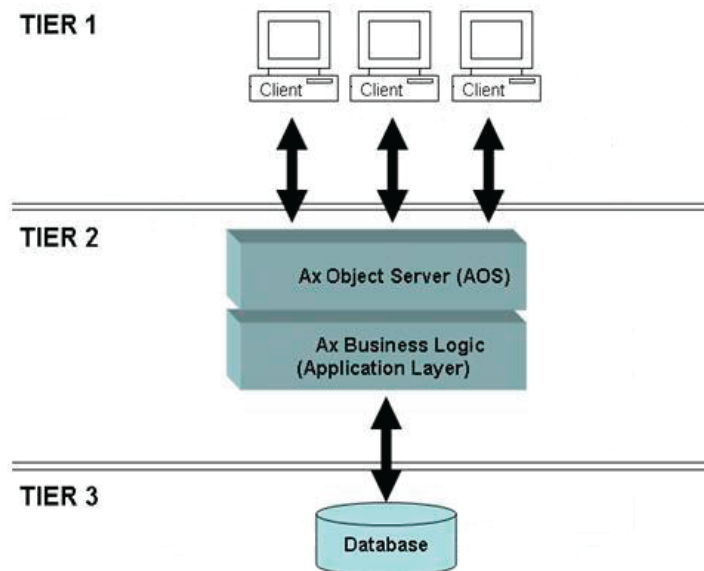


Figure 2: Architecture 3 niveaux (3 Tiers Architecture) <sup>[Ref.4]</sup>

### 3. Environnement Intégré de Développement


Le développement et les modifications apportés aux objets du logiciel sont réalisés à l'aide de la Suite de Développement *MorphX*, qui est un environnement intégré de développement spécifique à *Microsoft Dynamics AX*. Tout le design, les modifications et débogage sont intégrés à *MorphX*.

Le langage de programmation utilisée dans cette suite de développement est le *X++*, qui est un langage orienté objet similaire à *C#*.

### 4. Couches d'application

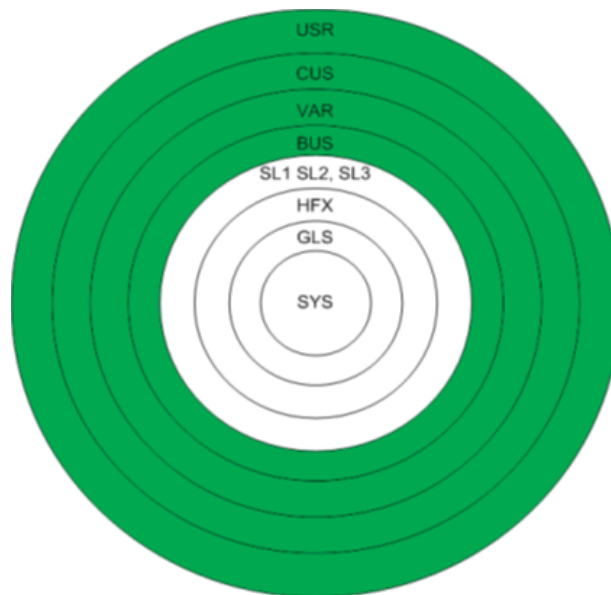
Pour séparer et contrôler les mises à jour et les modifications faites à l'application, le logiciel s'appuie sur une architecture de répartition du code source par couche « *Layered Architecture* ». Les couches sont hiérarchiquement disposées de telle sorte que quand des modifications ou des ajouts sont effectués pour un objet sur une couche, l'objet modifié éclipse l'objet de la couche inférieure.

De telle sorte on s'assure que l'application standard ne sera jamais modifié ni supprimé, car en supprimant un objet il sera effacé de la couche courante et des couches supérieures.

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 8	PAGES 20

Chaque couche d'application est sauvegardée dans un fichier séparé avec une extension .aod.


La figure suivante illustre la représentation des couches d'application pour *Microsoft Dynamics AX*:



**Figure 3:** Couches d'application <sup>[Ref.5]</sup>

Les couches SYS, GLS, HFX, SL1, SL2 et SL3 sont les couches standards de *Microsoft Dynamics AX*. Ces derniers ne sont accessibles que par l'équipe de développement de *Microsoft*. Alors que les couches BUS, VAR, CUS et USR peuvent être modifiées par les développeurs ou les utilisateurs.

Cela dit, pour avoir accès à ces couches les partenaires et/ou clients de *Microsoft* doivent avoir les droits de licence appropriés, dans ce sens, la couche BUS est réservée aux partenaires d'affaires de *Microsoft*, la couche VAR est réservée pour les revendeurs du logiciel, la couche CUS est réservée pour les clients et enfin la couche USR peut être réservée aux divisions internes des clients.

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 9	PAGES 20

### III. Modèle de Maintenance

#### A. Maintenance assurée par Microsoft


*Microsoft* sort une nouvelle version chaque trois ans, actuellement la version disponible sur le marché est *Dynamics AX 2009*, une nouvelle version (*Dynamics AX 2012*); est prévue pour le mois d'août 2011.

Pour chaque nouvelle version, il peut y a des modifications assez importantes qui sont effectués, que ce soit au niveau fonctionnel ou même architectural. Une entreprise qui fait une migration pour une nouvelle version doit gérer cette étape comme un projet de développement, vue la charge de travail nécessaires et les ressources mobilisés. En moyenne les client de *HAKA Solutions* font une migration à un intervalle d'une version pour une moyenne de sept ans entre les deux versions.

La maintenance assurée par *Microsoft* concerne les versions déjà sur le marché. Et cette maintenance est divisée en deux catégories.

La première catégorie est ce qu'on appelle les mises à jour *Rollup (Rollup Update)*; suite à la mise au marché d'une nouvelle version de *Dynamics AX*, les clients notifient *Microsoft* des erreurs et des bogues qu'ils ont trouvés dans la version vanille, *Microsoft* corrige ces erreurs et sort occasionnellement des paquets (*package*) de mises à jour *Rollup* qui contiennent des corrections d'erreurs (*Hot Fixes*), ces paquets sont disponibles aux clients à télécharger du site internet des partenaires (*Partener Source*). Pour la version *Dynamics AX 2009*, *Microsoft* a produit six mises à jour *Rollup*.


La deuxième catégorie est ce qu'on appelle les paquets de services (*Service Pack*); ce sont des sélections complètes de mises à jour d'améliorations ainsi que de nouvelles fonctionnalités que *Microsoft* produit après la sortie officielle d'une version *Dynamics AX*. Ces améliorations touchent des modules qui ont déjà été livré lors de la mise au marché, mais que certaines fonctionnalités n'ont pas été finalisées. Et cela concerne généralement les modules pour lesquelles les règles d'affaires changent d'un pays à un autre tel que la Paye par exemple. Les

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 10	PAGES 20

améliorations dans les paquets de services peuvent aussi concerner l'amélioration de la performance. Contrairement aux mises à jour *Rollup*, les paquets de services sont plus rares, pour la version *Dynamics AX 2009* on n'en compte que deux.

Les clients doivent par la suite installer ces paquets sur leurs serveurs, ces modifications et améliorations vont être installées sur la couche HFX. À ce niveau, un travail de maintenance est nécessaire aussi du côté du client; car dans le cas où le client aurait modifié l'application pour un objet spécifique, et si ce même objet est concerné par la mise à jour, alors le mainteneur du client doit gérer les conflits entre les deux modifications. Cette gestion de conflit se résout dans la majorité des cas par l'adaptation du code réalisé par le client au code de la mise à jour fournit par *Microsoft*.

La suite *MorphX*, offre à cet effet des outils qui aident les mainteneurs dans leurs tâches de gestion de conflit. Un des outils les plus utilisés est l'outil de comparaison, en effet cela permet, pour un objet tel qu'une méthode de classe, de comparer deux couches et identifier quels changements ont été effectués et sur quel couche le changement a été fait et ce en différenciant le code par couleur tel que illustré dans la figure suivante :

		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 11	PAGES 20

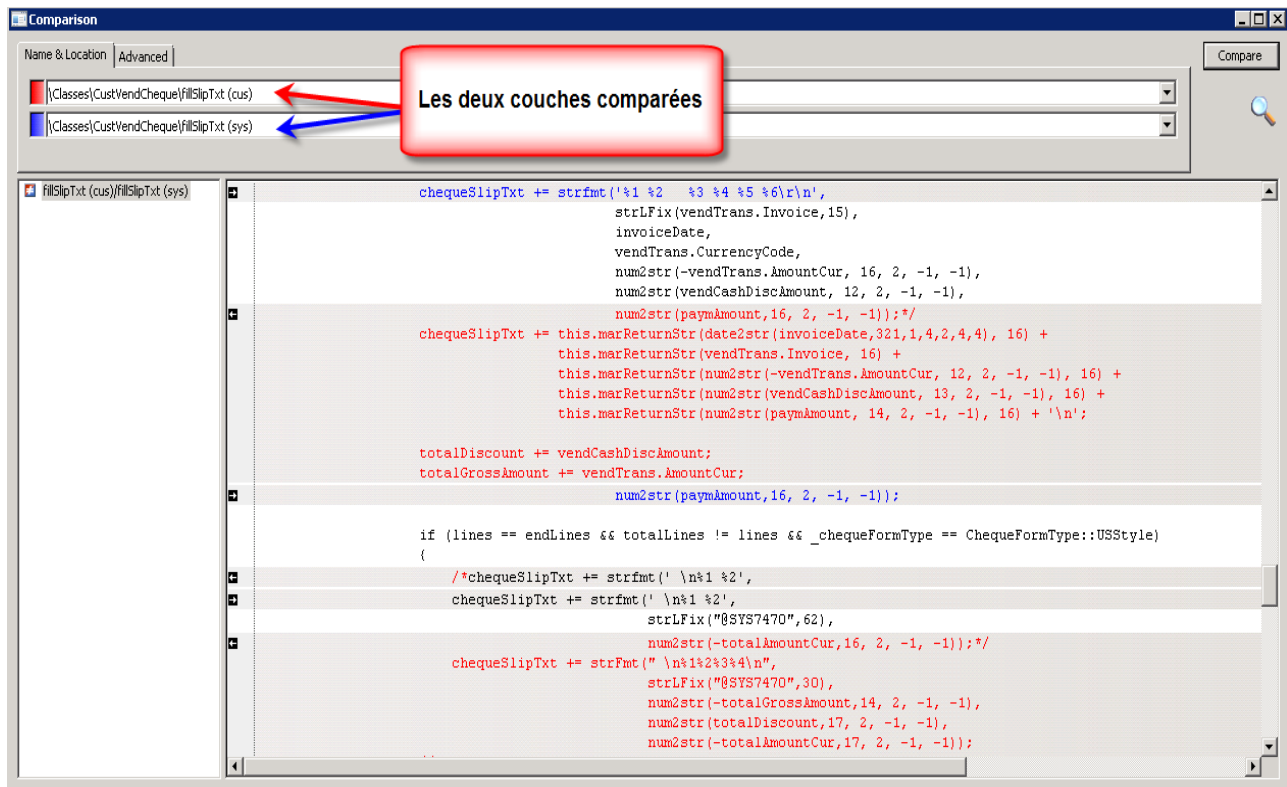



Figure 4: Outil de comparaison de *MorphX*

On peut voir que dans le cas illustré dans la figure 4, que pour la classe *CustVendCheque*, la méthode *fillSlipTxt* a été modifiée dans la couche CUS, et le code qui est en couleur rouge représente la partie exacte qui a été ajouté à cette couche par le client.

## B. Maintenance assurée par Sanimax

### 1. Contexte technologique

Tous les services informatiques de *Sanimax* sont sous-traités et gérés par des compagnies tierces, ainsi toute l'infrastructure informatique est hébergée à l'externe par *Bell*, le bureau de support est assuré par une autre compagnie dénommée *Global Help Desk Services*. Avant l'installation de *Microsoft Dynamics AX* en 2009, *Sanimax* gérait ses opérations avec une multitude de logiciels spécialisés, ou quasiment chaque département avait son logiciel et sa base de données indépendante, ces logiciels n'étaient pas forcément interconnectés. Et puisque tout est sous-traité pour chaque logiciel il y a un fournisseur différent qui assurait la maintenance de son côté.

		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 12	PAGES 20

En installant un *ERP*, Sanimax a pu centraliser ses opérations autour d'un seul système, et du coup elle s'est déchargée de tout le fardeau de maintenir différents systèmes et tous les coûts de maintenance très élevé ainsi qu'une amélioration dans les processus d'affaires. *Microsoft Dynamics AX* est devenu l'élément central du diagramme des logiciels de *Sanimax* tel que l'illustre la figure suivante :

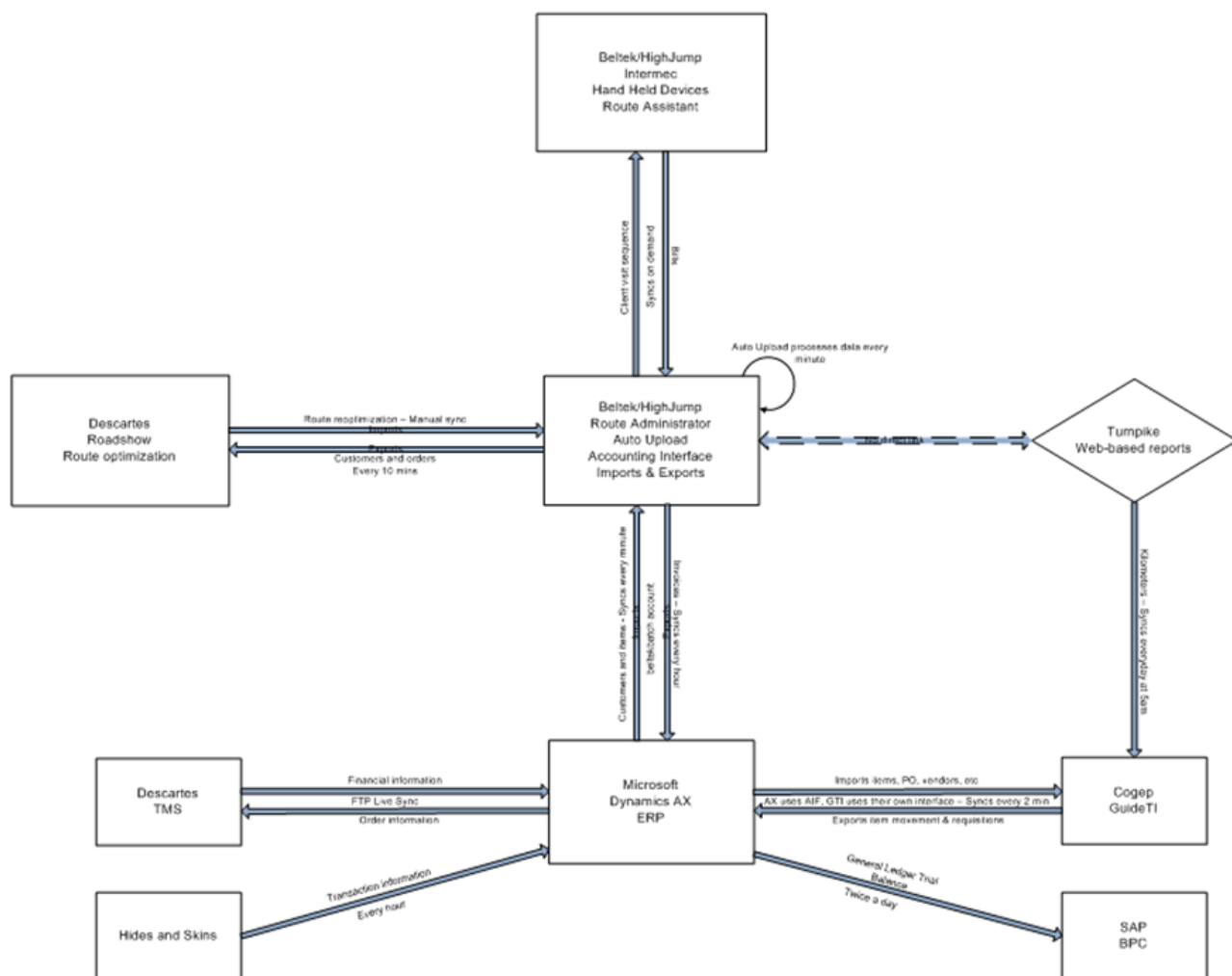



Figure 5: Diagramme de logiciels chez Sanimax<sup>[Ref.6]</sup>

		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 13	PAGES 20

## 2. Méthodologie *Sure Step*

*Microsoft*, à l'instar des autres très grandes entreprises du logiciel dans le monde, a sa propre norme et sa propre méthodologie de processus pour les produits *Dynamics* (GP, NAV, AX). Cette méthodologie détermine, à toutes les phases et étapes du projet, quels sont les processus à suivre et les meilleures pratiques pour garantir le succès de l'implantation de *Dynamics* ainsi que l'après implantation (après *GO-LIVE*) en terme de maintenance.

Donc, une partie importante de cette méthodologie décrit les meilleures pratique d'intégrer les modifications et la maintenance du logiciel ainsi que les mises à jour.

*Sanimax* s'est assuré de suivre les étapes décrites dans cette méthodologie *Sure Step*.

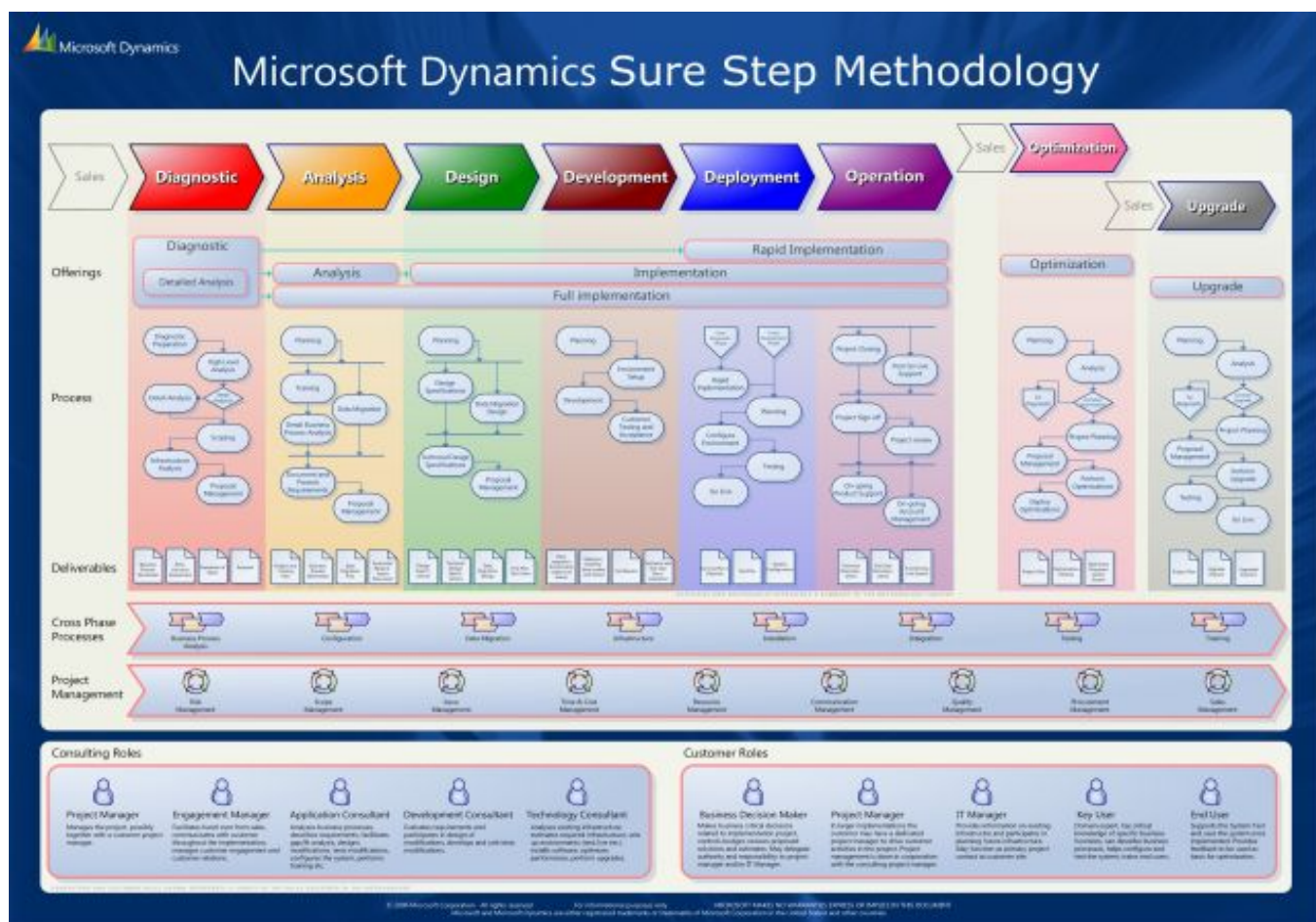



Figure 6: Méthodologie *Sure Step* [Ref.7]

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 14	PAGES 20

### 3. Modifications d'ERP avant le *GO-LIVE*: Maintenance ou Développement?

Le développement de *Microsoft Dynamics AX* est réalisé par l'équipe Recherche et Développement de *Microsoft*, cette solution se veut généraliste et répond aux besoins les plus communes des clients, une fois le système est prêt il est lancé sur le marché, c'est alors que les clients achètent la solution à son état vanille, mais les besoins de chaque client sont toujours uniques, alors les partenaires doivent modifier le système pour l'adapter aux besoins du client.


Cela nous amène à se poser la question suivante : Quand est ce que l'étape de la maintenance pour un *ERP* commence? Est-ce que cela commence dès que les partenaires modifient le code source? Où, Est-ce que cela commence une fois la solution modifié est livrée, autrement dit, après le *GO-LIVE*?

Je considère que le travail de maintenance sur les *ERP* commence après le *GO-LIVE*, car avant cette étape le client n'a pas encore commencé à travailler sur le logiciel, et surtout que la période de modification et d'implantation est trop longue pour qu'on considère ceci comme de la maintenance, cette étape dure en moyenne entre 3 et 10 mois pour une seule phase.

Donc, pendant la période de l'implantation chez *Sanimax*, une équipe de développement s'est chargé de programmer toutes les demandes du client pour changer les applications standards et ajouter des nouvelles fonctionnalités ainsi que la création d'interfaces pour communiquer *Dynamics AX* avec les autres logiciels que *Sanimax* a décidé de garder de l'ancienne infrastructure.

Une équipe d'utilisateurs clés a été affecté au projet à temps plein pour aider l'équipe d'implantation (développeurs et analystes d'affaires) à expliquer les processus et besoins d'affaires de *Sanimax*, ces mêmes utilisateurs clés ont testé les modifications, et c'est encore l'équipe de développement qui a procédé aux modifications pour corriger les erreurs détectés.

Puisqu'on a identifié cette étape étant une étape de développement, il est donc tout à fait logique que ça soit l'équipe de développement qui procède aux modifications.

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 15	PAGES 20


#### 4. Maintenance après *GO-LIVE*

Les premières semaines qui ont suivi le *GO-LIVE*, la structure de l'équipe d'implantation n'a presque pas changé, une fois le système a commencé à se stabiliser, Sanimax a pris la décision de promouvoir l'équipe des utilisateurs clés en *Centre de Compétence*, par contre l'équipe des analystes d'affaires a graduellement quitté le projet pour être remplacé par le centre de compétence.

Pour aider à la gestion de la maintenance, un logiciel de gestion de tickets (*ZenDesk*) a été implanté, à l'aide de ce logiciel la compagnie a eu un meilleur contrôle sur le suivi de stabilité du logiciel ainsi qu'avoir une idée sur la santé du logiciel, le nombre de bogues signalé dans *ZenDesk* donnait un indice révélateur sur la stabilité du système.

*Sanimax* classifie les demandes des utilisateurs en deux catégories; La première catégorie c'est les erreurs, ces bogues sont traités selon l'ordre de priorités et sont par la suite passées à l'équipe de développement pour les corriger.

La deuxième catégorie c'est les demandes d'améliorations, ces demandes sont regroupées par domaines d'affaires et affiliées à des projets (*Releases*), *Sanimax* planifie quatre *Releases* par années, les modules qui touchent les domaines d'affaires les plus critiques sont prioritaires. Cette deuxième catégorie est aussi confiée à l'équipe de développement pour être réaliser.

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 16	PAGES 20

## IV. Discussion

### A. Observations


La maintenance chez *Sanimax* est assurée par les développeurs, ce modèle organisationnel a été une obligation plus qu'un choix, il y a plusieurs facteurs qui expliquent ce modèle et les gestionnaires chez *Sanimax* en sont conscients.

Les raisons qui poussent *Sanimax* à subir ce modèle est directement lié à leur décision d'affaire de sous-traiter les services informatiques. En effet, une fois le projet d'implantation est terminé les compagnies de services mobilisent leurs développeurs pour de nouveaux projets, et la nature du travail fait que les développeurs, qui sont après tous des consultants préfèrent être sur des projets d'implantation, de passer de client en client et de voyager que de travailler pour supporter une plateforme à temps plein.

Le fait qu'après l'implantation plusieurs développeurs se sont succéder pour faire la maintenance, cela a impacté le temps de réaction des développeurs face aux problèmes, et ce malgré leurs expérience en développement, car ils se trouvent dans un système modifié de sa version vanille, le développement n'a pas été réalisé par eux, donc cela prend un peu de temps avec qu'ils se familiarisent avec les modifications qui ont été faites avant leurs arrivés.

*Sanimax* a essayé de remédier à ce problème en voulant engager des ressources à l'interne pour faire la maintenance, mais vu l'engouement affiché pour *Dynamics AX* et la rareté des ressources dans la région montréalaise, ils ne trouvent pas des ressources et ils ont de la difficulté à convaincre des consultants de faire le saut et de lâcher la consultation pour travailler chez eux et ce pour les mêmes raisons évoqués plus haut.

L'autre observation qui a impacté grandement la maintenance et la qualité du code, c'est la suppression de l'équipes des analystes d'affaires. En passant le flambeau à l'équipe du centre de compétence pour gérer les demandes d'amélioration, ils se sont passés trop vite de la compétence des gens qui connaissent la plateforme et tous les enjeux du logiciel. Ce que l'on remarque c'est que l'équipe de de centre de compétence, malgré leurs compétences dans leurs domaines


 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 17	PAGES 20

d'affaires ne maîtrisent pas très bien *Dynamics AX*, ce qui impacte la qualité de l'analyse et des dossiers de spécifications produites pour les demandes d'améliorations. Donc les développeurs se retrouvent avec des demandes qui ne respectent pas le standard de la plateforme, et cela peut avoir un impact néfaste sur la structure du logiciel et va la complexifier et par conséquent rendre la tâche des mainteneurs très difficile.

## B. Améliorations

Avoir des ressources techniques à l'interne pour maintenir un Microsoft Dynamics AX n'est pas une opération facile à réaliser vu le contexte actuel dans la région de Montréal. Une des solutions est de laisser le développement des demandes d'améliorations aux consultants qui représentent l'équipe de développement. En parallèle, pour ce qui est des bogues, il faudrait engager des ressources à l'interne, par contre au lieu de chercher des développeurs d'expérience, il faudrait engager des ressources juniors et les faire grandir à l'interne, en profitant de la présence des consultants pour les former.

L'autre amélioration qu'il faudrait apporter, c'est d'avoir des formations plus approfondies pour l'équipe de centre compétence pour qu'ils aient une image plus complète de l'*ERP* et améliorer la qualité de leurs dossiers de spécifications.

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 18	PAGES 20

## V. Conclusion

La littérature est très avare en ce qui concerne les processus relatifs aux *ERP*, elle l'est encore moins en ce qui concerne la maintenance des systèmes *ERP*. En l'absence d'études indépendantes d'organisme ou des chercheurs reconnus, on se limite aux normes et méthodologies créées par les compagnies fabricantes des *ERP*. Ces méthodologies, à l'instar de *Sure Step*, aident certainement à garder le cap, mais manquent cruellement de clarté et de structure par secteurs.


Il devient donc naturel de se référer aux normes et pratiques décrites en génie logiciel et essayer de faire un parallèle pour les progiciels. À partir de là on peut évaluer les pratiques de la maintenance dans le domaine de consultation en *ERP*.

La conclusion que je peux faire de mon expérience avec une clientèle variée, c'est que la majorité des entreprises mettent l'emphase et toute l'énergie sur la partie implantation, à ce niveau les moyens financiers et humains sont mobilisés. Mais dès lors que le progiciel est mis en production, tous ces moyens diminuent graduellement.

Les entreprises donnent peu d'importance à l'après *GO-Live*, et c'est une fois confronté aux problèmes qu'ils réalisent que les ententes de services qu'ils avaient signé négligent très souvent l'aspect maintenance. Et contrairement à ce que l'on peut penser, c'est par manque de maturité chez les entreprises que par volonté des sous-traitants de négliger ces aspects cruciales. Les clients commencent alors à improviser et gérer les problèmes au fur et à mesure, et c'est là que la qualité du logiciel peut se dégrader.


Il faut noter, que de plus en plus d'entreprise s'attaquent au problème avec la bonne approche, et ce en pensant à l'après *GO-LIVE* avant même de commencer l'implantation, et ce en incluant dans les ententes de services un plan de formations de mainteneurs qui pour la plus part sont juniors.

Je pense que le modèle de séparation des développeurs des mainteneurs est le meilleur modèle organisationnel pour la gestion des *ERP* peu importe la taille de l'entreprise et le nombre de

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 19	PAGES 20

ressources. Mais la structure doit être basée sur un modèle de partenariat avec les firmes de consultation, ainsi pour les projets de développement, les améliorations au système et les migrations il faut confier le travail aux développeurs consultants, tandis que laisser la maintenance quotidienne du système aux mainteneurs locaux.

*Sanimax* a acquis une maturité dans ce sens, un peu tard certes, mais et ils sont conscients que la maintenance est un aspect très important du cycle de vie de leurs *ERP*, et ils considèrent établir des processus pour remédier aux problèmes de maintenance.

 Université du Québec <b>École de technologie supérieure</b>		DOCUMENT NO.	DATE 2012-05-16	VERSIO 0.2
Cours MGL804	TITRE Rapport de session		PAGE 20	PAGES 20

## VI. Références

Réf. 1: Alain April, Améliorer la maintenance du logiciel page 26.

Réf. 2 : Office québécois de la langue française - <http://www.granddictionnaire.com>

Réf. 3, Réf. 4, Réf. 5 : *Microsoft dynamics 2009 Development 1*

Réf. 6 : Documentation infrastructure de Sanimax

Réf. 7 : *Microsoft Partner Source Documentation*