



Université du Québec


École de technologie supérieure

Département du génie logiciel et des T. I.

Travail de session #3

Étude du modèle de l'organisation de la maintenance

Étudiant	Richard Moreau Labbée
Contact	richard.moreau-labbee.1@ens.etsmtl.ca
Codes permanents	MORR11058103
Cours	MGL804
Session	HIVER 2011
Groupe	01
Professeur	Alain April
Date	2011-04-14

 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 2	PAGE 24

HISTORIQUE DES RÉVISIONS

Version	Date	Auteur	Sections	Résumé des modifications.
0.1	03-03-2011	Richard M	TOUTES	Template
0.2	03-04-2011	Richard M	1, 2	Introduction + développement
0.3	13-04-2011	Richard M	3	Maintenance
1.0	14-04-2011	Richard M	TOUTES	Document final



 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 3	PAGE 24

Table des matières

1. INTRODUCTION	5
MÉTHODOLOGIES DU TRAVAIL.....	5
DESCRIPTION DE L'ENTREPRISE	6
<i>Modèle de développement</i>	6
<i>Développement générique</i>	7
<i>Méthode du développement et déploiement</i>	8
PROJETS DE MISE À NIVEAU.....	9
2. ÉTAT DE LA MAINTENANCE.....	11
ORGANISATION DE LA MAINTENANCE.....	11
PROBLÉMATIQUES.....	12
JUSTIFICATION DU MODÈLE DE LA MAINTENANCE	16
ACTIONS POUR AMÉLIORER LA MAINTENANCE DANS L'ENTREPRISE.....	17
3. PRÉSENTATION D'UN GROUPE DE MAINTENANCE.....	19
ANALYSE DES RÉACTIONS DE L'ENTREPRISE.....	21
4. CONCLUSION	23
5. RÉFÉRENCES.....	24


 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance	PAGE 4	PAGE 24	

Liste des figures

Figure 1: Déploiement de version.....	9
Figure 2: Mise-à-jour d'une version	10
Figure 3: Interaction pour le support	15
Figure 4: Contexte de la maintenance S3M.....	19

Liste des tableaux

No table of figures entries found.

 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 5	PAGE 24

1. Introduction

Ce document présente les résultats recueillis lors de diverses entrevues effectuées envers plusieurs personnes ressources de mon organisation. Ces entrevues ont été effectuées pour analyser le modèle actuel de la maintenance et de le comparer avec un autre modèle de maintenance requérant un plus grand niveau de maturité.

Description du travail de session

Indiquez le modèle organisationnel utilisé, en maintenance, dans votre organisation, étudiez les raisons sous-jacentes avec l'administrateur, et questionnez cette décision en fonction des avantages et des inconvénients des deux modèles organisationnels.¹


Méthodologies du travail

Les informations et les commentaires, par rapport à la maintenance, ont été recueillis lors de discussions menées indépendamment avec quatre membres de l'organisation ayant des rôles distincts dans l'organisation. L'objectif étant d'obtenir la perception de la maintenance à divers niveau au travers l'entreprise (gestion, réalisation, exécution, etc.) et de ressortir les problématiques actuelles dans le travail de maintenance de l'application.

Voici les quatre rôles ayant été interrogés :

- Analyste de développement projet
- Chargé de projet
- Directeur du développement
- Directeur des technologies d'information

¹ Améliorer la maintenance du logiciel, travail de session 3, page 26

 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 6	PAGE 24

Les intervenants n'avaient pas de questionnaire précis à répondre, mais la discussion envers chacun fût orientée avec différents thèmes :

- Leur impact sur le développement
- Problématique actuelle sur la maintenance
- Comment améliorer la maintenance
- Interprétation de la création d'un groupe spécialisé en maintenance

Description de l'entreprise

Nom de l'organisation : GIRO inc.

L'entreprise œuvre dans la vente, le support et le développement de logiciel de gestion opérationnel dans le domaine du transport en commun et du domaine postal. L'entreprise est le chef mondial dans ce créneau principalement grâce à la force des algorithmes d'optimisation de solution. L'entreprise fait affaire avec diverses universités pour mettre en œuvre des algorithmes optimisés pour minimiser les coûts d'opérations des entreprises offrant des services de transport.


Autour des algorithmes, GIRO développe une suite logicielle pour faire de la planification opérationnelle et de la gestion journalière des opérations. Cette suite est en constante évolution depuis sa création initiale dans les années 80s.

Modèle de développement

L'entreprise compte plus de 230 employés, dont environ 90 développeurs d'application à temps plein. Pour effectuer le développement des applications, on peut considérer deux groupes distincts :

1. Développement générique
2. Projets

Le premier groupe est le groupe qui développe le cœur des applications qui est présenté aux clients. Le cœur de l'application générique est livré annuellement et est configurable pour certains besoins fréquemment retrouvés chez divers clients œuvrant dans le domaine d'affaires. Cette version est présentée comme une application utilisant les meilleures pratiques dans la

		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance	PAGE 7	PAGE 24	

gestion opérationnelle. Le deuxième groupe est un groupe qui adaptera une version générique aux besoins du client. Ce groupe est responsable d'effectuer l'élaboration des exigences, d'analyser les besoins, et de développer tout ce qui n'est pas jugé comme étant une nouvelle fonctionnalité générique.


L'entreprise fait ses revenus en effectuant des ventes de licence d'utilisation d'une version générique qui est évaluée en fonction du nombre d'effectifs de l'organisation qui sera gérée par l'application achetée. Au coût de la licence, on ajoute des coûts de développement requis pour que l'application réponde à tous (maximum) de besoin du client. En dernier lieu, les clients ont droit à un contrat de maintenance renouvelable à vie : ce contrat de maintenance est nécessaire pour être éligible à des projets de mise à niveau de leur application.

Développement générique

De façon générale, le développement se fait en silo avec une vision axée sur les fonctionnalités. L'application repose sur une plateforme de base qui fournit les classes et les méthodes pour gérer les transactions avec la base de données, les interfaces usager, les commandes et d'autres classes utilitaires : ceci est commun pour tous les produits développés par l'entreprise.

Lorsque l'on développe l'application au niveau de la gestion du modèle d'affaires, c'est la responsabilité des groupes de développement (par produit) de gérer les entités représentant le modèle d'affaires. Le modèle d'affaires est partagé par plusieurs produits, cependant la gestion des interactions du modèle est principalement gérée par produit, donc à la verticale. Il y a un lien d'héritage entre les classes représentant une même entité d'affaire réutilisable dans les divers produits, mais la gestion de ces objets est malheureusement maintenue par produit. Ceci fait qu'un changement dans un objet peut impacter dans d'autres produits de l'application. Ceci est un phénomène fréquent dans l'entreprise alors qu'une mise en production brise d'autres fonctionnalités, ce qui est généralement détecté par le groupe de test.

Les documents disponibles dans l'intranet sont faits pour répondre aux besoins et sont rédigés sous forme d'analyse fonctionnelle et organique (structure du code). Ces analyses sont très ciblées sur une fonctionnalité ou architecture, ce qui fait qu'un portrait global de l'application est difficile à faire. La documentation usagée est maintenue, mais pas la documentation technique (modèle de données, relation entre les objets de l'application). Ceci fait que les documents d'analyses sont rapidement dépassés par la réalité de l'application. Les seniors restent toujours les meilleures ressources pour obtenir une explication d'ensemble, mais sont plus ou moins exacts puisque les changements subséquents peuvent avoir modifié le comportement depuis. L'entreprise n'a aussi aucun intérêt à ajouter la tâche de maintenir la documentation de modèle


 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 8	PAGE 24

aux développeurs. Ceci est dommage, car cette pratique était faite dans l'entreprise il y a une dizaine d'années seulement. Elle a été abandonnée pour axer les développeurs vers les besoins des clients pour livrer plus rapidement les solutions promises.

Depuis peu, un analyste-programmeur (AP) faisait les analyses d'affaires, organique et le code. Maintenant, certains groupes de développement innovent en séparant les tâches en ayant plus de personnel dédié au rôle strictement d'analyse, ce qui permet une plus grande ouverture vers le maintien des modèles d'affaires. Encore ceci dépend des groupes de développement et de la vision du rôle d'analyste, car d'autres groupes donnent une fonctionnalité exclusivement à un seul AP.

Méthode du développement et déploiement

Suite à la vente d'une licence, le développement projet doit ajouter des fonctionnalités au code de la version vendue. Depuis les spécifications clients et des besoins, il est possible que certaines fonctionnalités soient développées dans les versions génériques postérieures à la version vendue au client. En exemple, voici un cas de vente d'une nouvelle licence de l'application 2010 que l'entreprise s'engage à déployer en 2012 avec les fonctionnalités supplémentaires pour les besoins du client :

		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 9	PAGE 24

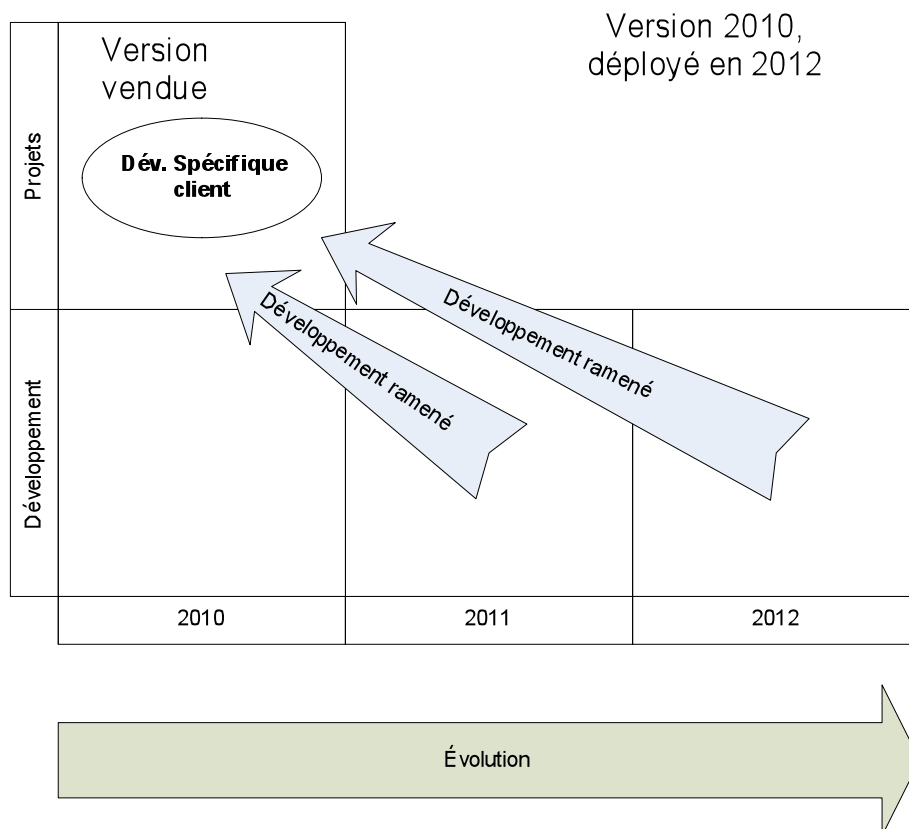



Figure 1: Déploiement de versions

Plusieurs développements ont été délégués au département de développement générique et ces développements doivent être refaits dans la version vendue au client. Le groupe projet doit donc tenir compte des changements architecturaux apportés aux multiples versions et de tester les modifications refaites. Plus le développement provient d'une version éloignée, plus il est difficile de la refaire dans une version cliente.

Projets de mise à niveau

Si le client demande une mise à niveau de sa licence pour 2014 pour bénéficier des réductions des coûts de migration, tout en étant soucieux du suivi technologique, l'organisation fait face à deux problématiques :

1. Maintenir les développements qui sont propres (spécifiques) aux clients

		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 10	PAGE 24

- Analyser les développements qui ont été rapatriés de versions postérieures à la version de vente.

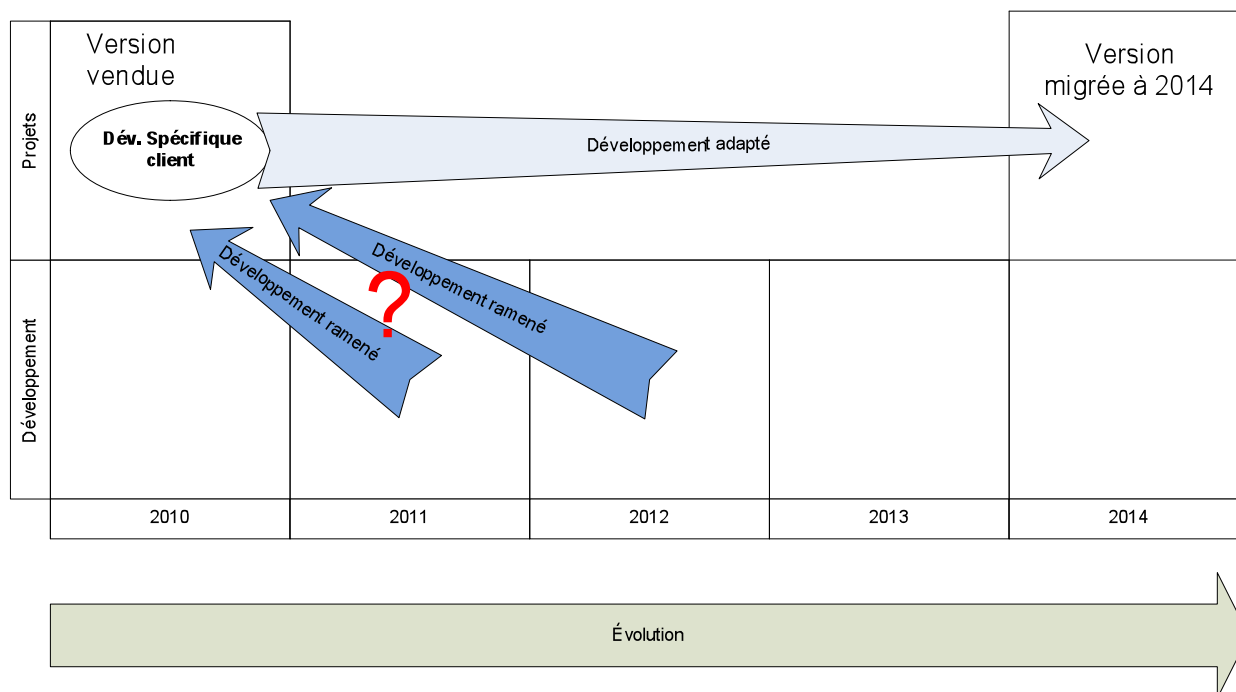



Figure 2: mise à jour d'une version

 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 11	PAGE 24

2. État de la maintenance

La méthode de développement introduit divers problèmes pour le maintien de l'application. Cette section explique comment la maintenance de l'application est gérée par l'organisation.


Organisation de la maintenance

Il existe deux modèles types pour l'organisation de la maintenance. Le modèle le plus commun est que ce soit le développeur qui maintienne son développement au fil du temps, car il est considéré le spécialiste de son développement et de son code : GIRO adopte très fidèlement ce modèle organisationnel de la maintenance.

Dans l'entreprise, on identifie les développeurs en tant qu'analyste-programmeur (AP) car on demande qu'une personne puisse analyser des fonctionnalités, du code et de coder les changements nécessaires au code source pour intégrer les nouvelles fonctionnalités. Il est rare qu'une personne hérite d'une tâche uniquement de codage provenant d'un analyste ou d'un autre analyste-programmeur : le rôle d'analyste est plus un rôle d'analyste des besoins/architectural et non fonctionnel.

Les demandes de changement sont traitées par un responsable de produit ou un analyste, mais les priorités sont originalement catégorisées par l'auteur du billet. La priorité est validée par le responsable des billets dans les divers groupes de développement. Si la priorité est assez importante, le billet est assigné au développeur de la fonctionnalité pour qu'il puisse corriger le problème ou à un autre développeur dépendamment du degré de complexité. Le développeur a donc la responsabilité de corriger le problème dans un délai assez court. Cependant, si le développeur estime que le temps de développement est grand, ou que le problème nécessite des changements plus larges, il est probable que le problème nécessite une analyse supplémentaire ou que le ticket soit classé pour être corrigé plus tard lors d'un développement futur.

Dans les tickets, le niveau de détail est relatif à la personne qui la crée (toujours une personne de l'organisation, les clients de l'entreprise ne peuvent eux même ouvrir de ticket, mais peuvent soumettre aux chargés de projet une problématique). Il n'y a pas d'étapes formelles au travers l'organisation pour la gestion des tickets, ceci dépend des différents groupes de développement qui ont leur propre méthode (se ressemblent, mais exigences différentes). Les demandes de

		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance	PAGE 12	PAGE 24	

correction au niveau de la version générique provenant des groupes de développement projet sont toujours prioritaires, car ceci touche une version déployée ou bloquante pour le développement des versions clientes. Dans les ententes de maintenance, le client indique quelle personne (ou plutôt quel rôle de l'employé chez le client) peut envoyer des requêtes de changement

Les tickets qui ne sont pas des demandes de correction (perfectif, amélioration, prévention) sont rarement complétés et sont majoritairement gardés comme étant des observations de l'état du code. En pratique, rien n'est explicitement fait pour fermer ces types de billets. 87% des bogues majeurs ou critiques (correctifs) sont corrigés dans des délais acceptables. Sous ce chiffre, l'organisation juge que l'effort de la maintenance est plus qu'acceptable.

La petite maintenance n'est pas vraiment gérable, mais une gestion des priorités est faite pour autoriser les correctifs. Ce qui est critique et expédié au client est c'est de la responsabilité du client de faire les tests sur la version corrigée avant de l'utiliser dans les opérations. Ce qui est majeur et accepté comme correctif est packagé dans une version livrée périodiquement (3-4 forfaits par années).


La justification de cette organisation est justement basée sur les statistiques qui prouvent que la gestion des bogues est efficace. Demander à l'auteur de corriger un problème qu'il a lui-même introduit via ses changements est efficace, car il connaît le code, l'architecture ainsi que la séquence des appels des fonctions et méthodes. Il faut aussi tenir compte qu'un développement doit être livré dans une version et qu'une version est livrée dans le temps d'une année. Les tests d'intégrations sur une fonctionnalité sont effectués avant la fin de la version, mais pas sur les versions subséquentes. Généralement, avant la fin de la version, le développeur est toujours à l'emploi et est donc en mesure de corriger les problèmes prioritaires détectés lors des tests d'intégrations.

L'entreprise identifie deux activités principales de la maintenance :

- Entretien correctif
- Support (documentation)

Problématiques

Les problématiques sont multiples face à la maintenance des applications de l'entreprise. Dépendamment du niveau d'intervention dans la hiérarchie de l'entreprise, la vision diverge sur

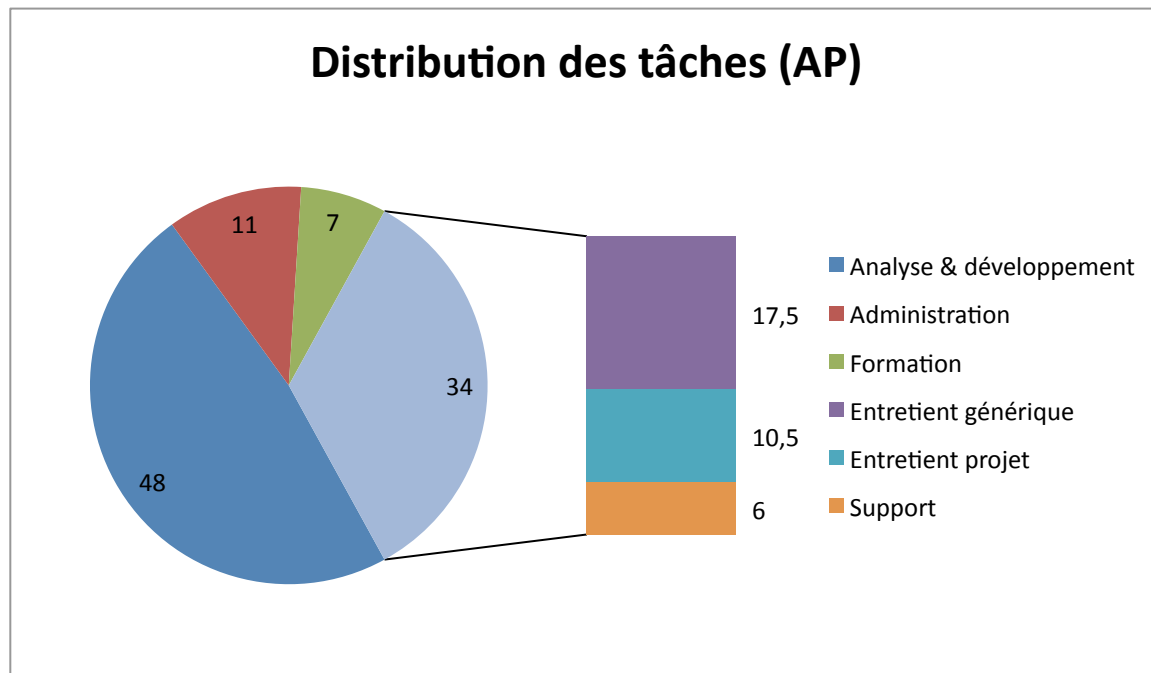
		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance	PAGE 13	PAGE 24	


les problèmes. De façon générale, la maintenance est mal vue, car, bien que nécessaire, elle est une distraction aux activités principales et rentables de l'entreprise.

Voici une liste sommaire :

- Gestion des priorités difficile à faire (quoi corriger, pour qui pour quoi)
- Interruption dans les développements
- Empiète sur les délais de livraison
- Expertise volatile
- Les groupes dans le développement fonctionnent différemment
- Documentation relative sinon approximative (dépend des analystes) autant dans le code que dans les documents d'analyses

Dans les hautes instances, on considère principalement que la maintenance coute trop cher à l'entreprise. Trop d'effort est mis dans la correction de bogues et dans le support lors du développement. Pour le rôle typique d'analyste-programmeur (AP), le tier du temps est investi dans des tâches reliées à la maintenance :



		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 14	PAGE 24

Ce temps réduit considérablement la productivité d'un AP alors qu'il ne peut qu'investir que 48% de son temps dans les tâches dont il est spécifiquement assigné et spécialisé. Un fait à noter est que la documentation qui est rédigée dans le processus d'analyse et de développement n'est pas incluse dans les chiffres de la maintenance. En tenant compte des temps des analyses et de la rédaction de ces dernières, il est plus réaliste de considérer 25% du temps en analyse/développement pour la rédaction de la documentation. Ceci monte l'effort d'un AP au alentour de 46% de son temps dans les tâches identifiées à de la maintenance et réduit le temps d'analyse et de réalisation à 36% : les chiffres s'inversent.


Du côté de la réalisation de la maintenance, les problèmes surviennent alors que l'auteur du code n'est plus disponible dans le groupe responsable de la fonctionnalité touchée par une demande de correctif (changement de groupe, ne travaille plus dans l'entreprise). Il devient alors difficile de comprendre comment la mécanique fonctionne pour ne pas risquer que briser d'autre fonctionnalité et d'être efficace (temps pour le correctif). Ceci est un constat fréquent avec ce modèle d'organisation.

Outre les demandes de corrections qui peuvent survenir n'importe quand dans le cycle de vie de l'application, il y a plusieurs problèmes dans l'entreprise lors des projets de mise à niveau des installations des versions clientes. Ce type de projet est une source de revenue peu négligeable, mais qui peut se transformer en cauchemar lors de la réalisation :

- Difficulté à migrer d'une version ancienne à récente (rattrapage de l'écart technologique)
- Migration minimale dirigée par le département de marketing (budget)
- Chaque demande de mise à niveau additionnelle pour un client est pire que la précédente
- Seulement les spécifications d'origines sont disponibles
- Support exclusivement verbal, documentation n'est pas maintenu

Pour résoudre les problèmes des processus de changement, les mainteneurs maintiennent un babillard pour regrouper une liste de correctif pour les diverses fonctionnalités. Cependant, le maintien est relatif et est toujours prompt à d'autres problèmes (oublie, construction de la liste réactive et incomplète).

Suite aux demandes des clients, voici les interactions possibles suivant la demande de support à partir d'un client :

		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 15	PAGE 24

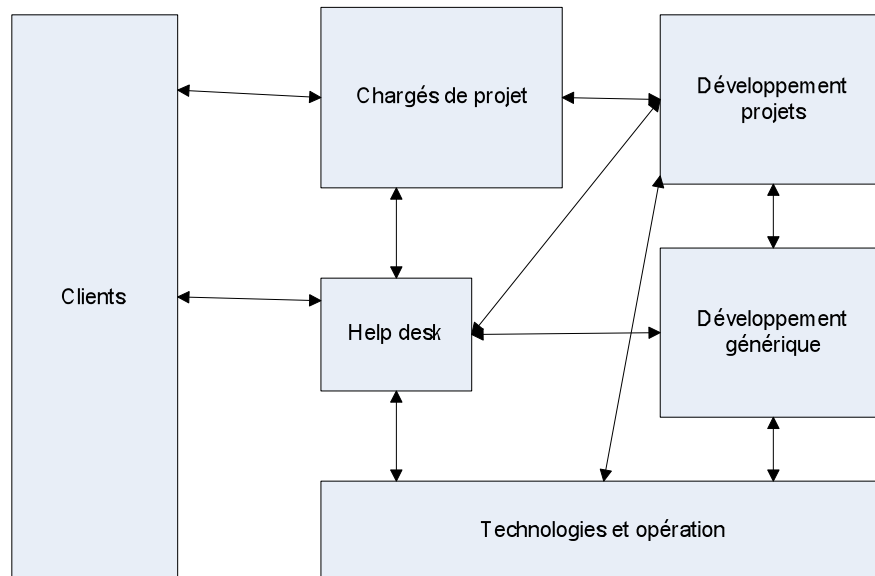



Figure 3: Interaction pour le support

On voit qu'à l'interne, il n'y a pas de point central pour répondre à des questions de support. C'est possiblement approximatif comme demande d'aide bien qu'au fil du temps, une personne est capable d'identifier directement la bonne personne ressource pour obtenir la bonne information rapidement.

L'origine provient du modèle de développement et des différents documents réalisés lors des différentes phases de développement. La grande majorité des livrables fournit une idée architecturale en documentant les nouvelles classes et méthodes. Les diagrammes sont plus ou moins détaillés, si ces diagrammes existent : ils sont souvent présentés par du pseudo-code. Par la suite, les documents ne sont pas mis à jour lors de la mise en production pour qu'ils reflètent ce qui existe réellement dans le code de l'application.

Le fait que les documents ne sont pas cohérents avec ce qui existe dans le code inculque un certain cynisme face à l'information que l'on retrouve dans les documents d'analyse. L'idée de base reste probablement viable dans le document, mais les détails sont souvent dépassés. Dans le cas d'un développeur qui doit maintenir du code que plus aucune personne ne connaît, on doit généralement se rabattre avec des essais-erreurs pour comprendre comment le code fonctionne et pour introduire un changement efficace : il est probable que plus d'un correctif soit nécessaire, car l'évaluation des impacts est souvent bâclée puisqu'aucune documentation ne peut aider précisément à ce niveau.

		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 16	PAGE 24

Justification du modèle de la maintenance


C'est la direction qui aligne le modèle de l'entreprise et ne considère guère une avenue vers la création d'un groupe axé spécifiquement sur la maintenance de l'application. De prime abord, les critères d'embauche sont relativement élevés ce qui fait que les employés demandent du travail intéressant. Le travail de mainteneur est vu comme un travail de second degré et peu intéressant. Outre pour former un employé au code existant dans l'application, les responsables ne voient pas l'intérêt d'assigner une personne à la maintenance. Il n'est pas envisageable de baisser les critères d'embauche (collégiale) car les groupes de développement préfèrent un profil universitaire pour combler les responsabilités d'analyste.

La direction ne voit pas de problème avec la maintenance, car l'entreprise corrige tout de même la majorité des problèmes importants. Ce qui est problématique avec la maintenance est qu'elle coûte cher et que ceci peut être amélioré en coupant dans le support. Il faut simplement être moins bonasse face aux clients (internes?) et refuser de répondre à tout prix dans le simple but de le rendre satisfait.

La direction considère aussi que la documentation (outre la documentation pour les clients) est peu utile, principalement par ce que les spécialistes restent longtemps dans l'entreprise. Elle se rend aussi compte que la documentation est peu utilisée suivant la revue d'analyse.

Les départements sont toujours dirigés par des budgets et il n'existe pas ou très peu de fond pour des tâches de rénovation de l'application. Il serait bien généralement d'avoir des projets de rénovation, mais ils sont souvent utilisés pour des changements architecturaux en prévision d'ajout de fonctionnalités dans les années à venir. Ce type de projet ressemble donc à des projets de développement préparatoire pour l'intégration d'autres fonctionnalités encore en phase de besoin.

L'entreprise se perçoit comme étant une entreprise de développement de logiciel ce qui fait qu'il est plus avantageux d'investir dans des meilleures pratiques de développement. Il y a plusieurs comparaisons faites avec le modèle CMMI pour améliorer la gestion des projets. Tant que les projets de développement sont contrôlés, le reste devrait suivre... GIRO est une boîte de développement et il faut se concentrer sur les activités principales de l'entreprise. On ne veut pas trop en faire au niveau des procédures non plus, car en ajoutant des processus, on bureaucratise la tâche des employés et ceci devient fastidieux et ennuyant : on garde toujours l'objectif d'avoir du travail le plus intéressant pour un employé. Dans les faits, les rôles au sein de l'entreprise n'ont pas évolué outre une quantification d'expérience à même les rôles (niveaux). Il semble que la gestion de la croissance de l'entreprise se fait lentement et toujours de façon réactive aux

		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance	PAGE 17	PAGE 24	


problèmes. Par contre, l'évolution des rôles, ou la division des responsabilités des employés sont plus subtiles en termes de rentabilité dans l'entreprise, car elle n'est pas quantifiée. On est capable d'évaluer approximativement la productivité d'un employé, mais pas des rôles. On est plus enclin à dissuader un employé à faire une tâche peut intéressante qui n'est pas rentable pour l'entreprise qui d'expliquer les raisons en quoi l'employé a dû effectuer le travail. Puisque rien n'est standard au travers les multiples groupes de développement, les employés ayant un même rôle ont des situations complètement différentes. On laisse aux personnes responsables des groupes inculquer leur propre méthode de gestion et de fonctionnement ce qui rend aussi le partage des ressources plus difficile vu l'adaptation nécessaire suite à un transfert d'employé. On veut laisser libre court aux talents de l'employé pour garder un intérêt élever dans le travail. L'entreprise est fière du niveau de satisfaction de ces employés et peut être vérifiable par les bonnes relations entre employés.

Actions pour améliorer la maintenance dans l'entreprise

À tous les niveaux, il est toujours désirable de diminuer la maintenance. Du côté des TI, les statistiques démontrent que le processus est efficace, mais d'un côté de la gestion des effectifs, on désire augmenter la productivité en diminuant le temps dans les tâches de maintenance.

Suite aux discussions, il ne semble pas y avoir d'action concrète pour répondre à la problématique du temps en maintenance et support. Comme l'organisation considère que la petite maintenance est non gérable, mais nécessaire, en plus que les chiffres démontrent que le processus est efficace, ce n'est pas cette facette de la maintenance qui est à améliorer. Ce qui est problématique pour (les chiffres de) l'entreprise est de justifier le temps investi dans le support aux clients. Ce support touche autant le support directement aux clients de l'entreprise que le support entre les groupes de l'entreprise (support développement aux projets en exemple).


Les actions sont souvent prises suite à des chiffres comptables avec une vision marketing. En aucun cas une action ne fût clairement justifiée par une analyse d'impact sur les processus découlant de la maintenance. On suit des objectifs comptables de haut niveau sans trop savoir si le développement sera affecté. L'entreprise est généralement bien structurée pour le développement, mais les processus d'amélioration continuels semblent déficients en se limitant à un niveau réactionnaire. Le problème avec la gestion de maintenance est qu'elle ne doit pas être faite de façon réactionnaire, mais continue. Il ne sera pas évident de savoir quand le processus de maintenance sera à revoir comme il n'y a aucun intérêt à le comprendre. Selon les statistiques, 87% des bogues prioritaires sont corrigés dans un délai acceptable. Cependant, le nombre de bogues prioritaire augmente chaque année et cette tendance semble s'accroître plus rapidement

 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 18	PAGE 24

que le nombre de développement et de ressource disponible. L'action de l'entreprise ici est de développer plus de suite de test pour découvrir ces problèmes le plus rapidement possible, sans pour autant évaluer les impacts sur les développements des groupes de développement alors qu'ils seront interrompus continuellement.

Malheureusement, comme les groupes sont gérés indépendamment, il est impossible de savoir le moment exact que les groupes de développements, suivant les phases de tests en fin de livraisons, peuvent reprendre les tâches d'analyses et de développement. Par contre, il est évident que les premiers mois sont peu productifs justement par les interruptions fréquentes demandées par l'intégration des correctifs.

On tente de minimiser la perception d'une maintenance couteuse aux clients par les coûts faibles de contrats de maintenance (18% du coût de licence). Le département est aussi très actif en encourage les projets de mise à niveau (requiert de mettre à jour les développements spécifiques au client uniquement). Encourage les updates aux 3 ans – réduction des coûts de la licence, coûts du développement propre au client moins difficile (écart de la version de base moins grand). Explique un suivit technologique avec les plateformes (OS, SGBD), bonne pratique.

		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance	PAGE 19	PAGE 24	

3. Présentation d'un groupe de maintenance

Le contexte de la maintenance tel qu'élaboré par le S3M a été expliqué et vulgarisé aux intervenants.

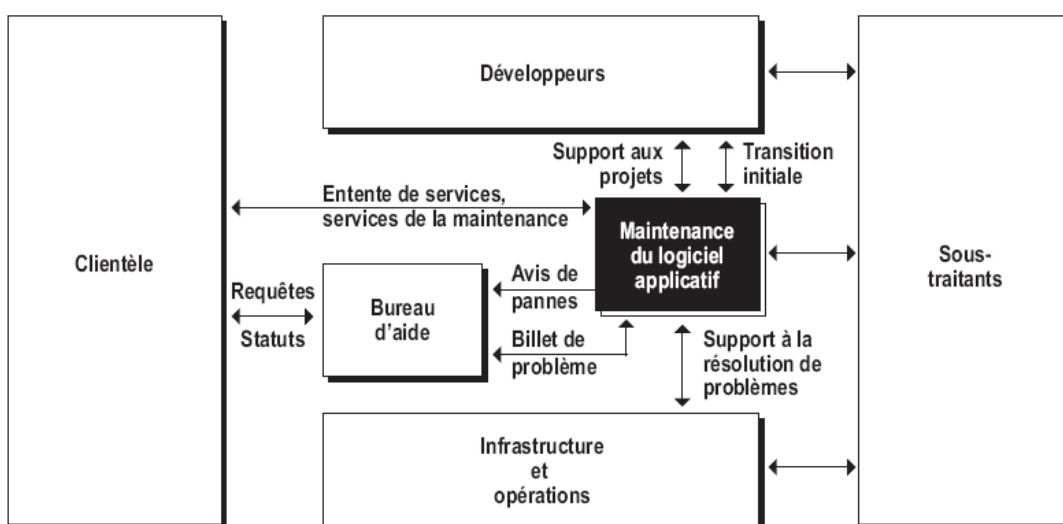



Figure 4: Contexte de la maintenance S3M²

Dans les discussions avec les intervenants, l'intégration d'un groupe de maintenance a été discutée. Voici un sommaire des réactions :


- Direction
 - Travail pas intéressant, on veut offrir du travail motivant
 - On ne veut pas investir dans la maintenance, mais bien la réduire

² Améliorer la maintenance du logiciel, p. 138

 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance	PAGE 20	PAGE 24	

- Les problèmes actuels ne sont pas la correction de bogue, mais le support (manque la notion proactive de la maintenance)
- La documentation n'est pas utilisée, donc on ne veut pas en faire plus
- Pas de budget pour les rénovations (adaptatif)
- Chargé de projet de développement
 - Améliorerait la collaboration pour les développements
 - Orienterait les groupes pour avoir des relations de fournisseur-client et non seulement fraternel
 - Potentiellement une personne de plus dans les prises de décision
- Analystes
 - Pourrait valoriser les employés moins spécialisés dans le développement (DEC)
 - Élimine le travail inintéressant de la tâche des analystes-développeurs (plus productif, car constamment concentré sur leur mandat)
 - Un personne de plus dans les prises de décision, mais ce n'est pas elle qui est de trop dans la procédure actuelle (voir les responsables fonctionnelles et l'ingérence dans le code)
 - Aiderait pour la migration des versions clients, car la connaissance horizontale serait plus centralisée
 - Le groupe de maintenance sera divisé par produit ultimement, car c'est la culture de l'entreprise d'avoir une vision exclusivement verticale (manque de confiance dans le management)

Il est évidant que la compréhension de l'utilité du groupe ainsi que les processus entourant le groupe ne sont pas complètement assimilés par les interviewés, mais les impacts semblent avoir été compris par les niveaux plus techniques. Il y a un blocage qu'en a la volonté des membres de la direction de voir les avantages d'un groupe de maintenance ce qui est un peu dérangeant. Il s'agit probablement d'une conséquence de l'organisation même alors que ce sont les employés les plus anciens au sein de l'entreprise qui détiennent les postes d'impacts. Les changements sont très lents à implanter et ça doit rester convivial. D'un autre côté, le côté de la culture fraternel

 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 21	PAGE 24

pour maintenir des relations de travail le plus agréable possible semble limiter la possibilité de challenger les prises de décision.


Analyse des réactions de l'entreprise

Il y a plusieurs contradictions dans les objectifs et les actions des membres de la direction face à la maintenance. L'observation la plus évidente est qu'aucune personne n'est capable d'associer une statistique à une action concrète. Le concept de mesure est inexistant : nous ne sommes qu'au niveau des statistiques et des chiffres comptables. L'entreprise est clairement réactive aux diverses problématiques qu'elle rencontre et ce comportement est visible à tous les niveaux (gestion, développement, maintenance).

Il y a un manque flagrant de compréhension de diverses causes à effet dans le manquement du processus de maintenance dans l'entreprise, principalement dans le maintien des documents. Le fait de vouloir couper dans le support client, ou plutôt dans la méthode de support, peut être légitime dans le sens que l'entreprise répondait à des questions qui n'ont aucun lien avec ce qu'elle vendait ou lorsque le client manquait littéralement de formation. Cependant, il est discutable des effets bénéfiques de ces coupures sur le chiffre d'affaire de l'entreprise : est-ce que ça augmentera vraiment la productivité des employés ou ceci réduira l'indice de satisfaction du client? On veut offrir moins de support, donc on désire que les employés soient moins utilisés pour expliquer le mode de fonctionnement d'un produit, mais on ne veut pas plus documenter pour rediriger les demandes de support vers des documents pertinents...

Le fait que la direction considère que la documentation est peu utile ne peut que renforcer l'une des problématiques premières de la maintenance énumérée par la direction même, soit que l'expertise est volatile. Elle se base strictement sur les connaissances d'un expert en espérant que l'expertise puisse se répandre au fil du temps avant que l'expert quitte l'entreprise. Ceci a aussi un autre effet alors que cet « expert » change de fonction, il doit continuer d'offrir le dû support tant que le transfert de connaissance n'est pas complété. Dans la situation actuelle de la maintenance de l'entreprise, il est clair que le transfert de connaissance ne sera jamais complet, ce qui fait que l'expert sera continuellement forcé à offrir le support. L'expert sera continuellement interrompu, car la maintenance n'est que réactive suivant une interruption par la détection de bogue.

D'un autre côté, est-ce que le fait de savoir que les documents sont incohérents avec le code lors de la mise en production d'un développement justifie le simple fait de ne pas utiliser la documentation par la suite? Il ne sert à rien d'utiliser quelque chose que l'on sait non-fonctionnel ce qui fait que cet axiome prouve que la documentation ne sera pas utilisée par la suite.


 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 22	PAGE 24

Au contraire, les plans d'action demandent de passer moins de temps dans les phases de discussion et plus en réalisation : bref, augmenter la productivité. Pour ce faire, il fût demandé de faire des documents plus concis, mais en optimisant les réunions avec diverses mesures. Il reste à voir les impacts de ces changements dans les documents et les réunions, mais rien ne prouve que ceci est en train de s'améliorer.

Pour l'instant, la direction estime que l'entreprise est trop bonasse face à ses clients: toutes les demandes de changements sont (étaient) acceptées ce qui faisait que des problèmes mineurs coupaient sur des temps de livraison et de développement. Ceci est signe d'un problème de management et du processus des demandes de changement qui est manquant face aux clients externes. Entre le groupe de développement et les groupes de projets, si les projets créent un ticket avec une date de livraison, il est convenu qu'elle doit être corrigée dans les délais, peu importe l'importance vue par le groupe de développement. On se fit sur le bon jugement de la personne ayant créé le ticket. NOTE : avant une livraison, il est fréquent que certains groupes de projet créé une multitude de tickets avec des délais très courts, ce qui monopoliserait le groupe de développement. En conséquence, le groupe de développement réajuste les priorités des demandes de correction au déplaisir du demandeur.

Les groupes de développement générique sont généralement plus fermés à la collaboration directe dans les revues d'analyse des développements projets vu qu'ils n'offriraient que du support et que leur mandat serait impacté. Les responsables du développement proviennent d'un profile plus techniques, analyste et manque un certain niveau de connaissance de gestion de projet. Il y a une divergence apparente dans la terminologie entre les responsables du développement et les chargés de projet alors que les chargés de projet voient d'un bon œil l'implication des analyses de développement dans les réunions pour valider les développements spécifiques. Les groupes de développement préfèrent faire un support sur ce qui a été fait que d'être impliqué dans les réunions des autres groupes. De plus, les développements dans les versions de développement sont souvent faits sans le consentement des groupes de projet ce qui leur impose une architecture.

Le niveau technique semble être enthousiaste à la création d'un groupe uniquement pour la maintenance. L'argument premier fût cependant pour que les développeurs ne fassent plus de la maintenance, car c'est peu motivant. Il est certain que le niveau technique ne peut pas forcément voir l'effet global d'un groupe dédié, mais garde des réserves pour le maintien des documents. Les développeurs pourraient être réfractaires à maintenir leur document d'analyse suivant un développement pour corriger l'écart entre l'analyse et le code en production. Il pourrait s'agir d'un simple problème de culture puisqu'aucune personne n'effectue ce travail en plus que peu de gens se basent sur ces documents présentement.

		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance		PAGE 23	PAGE 24


4. Conclusion

L'entreprise est relativement bien structurée pour mettre à terme un projet de développement. Sans être certifié CMMI avec un niveau quelconque, elle a su intégrer à ses processus certaines pratiques pertinentes pour réussir à livrer des applications de qualité. Cependant, GIRO est orienté vers la livraison de ses projets et malheureusement, tout le niveau de direction adopte la vision marketing de l'entreprise. On part d'une liste de besoin et on tente de livrer des solutions.

La gestion du développement laisse un peu à désirer. Les diverses phases d'un développement existent à partir de l'élaboration des exigences jusqu'à la livraison, en incluant les tests. Cependant, la suite du maintien de l'application est approximative et purement réactive. On se contente de sortir des statistiques qui fluctuent dans le temps, les chiffres montent d'une année à l'autre, mais nous ne sommes pas capables de justifier les variations sauf lors de projet de maintenance majeur (création énorme de bogue) et que l'application devient de plus en plus grosse (plus de bogues en conséquence).

Avant de pouvoir évoluer dans une meilleure gestion de la maintenance, vu la fermeture à justifier ses avantages, il faudrait passer par une étude d'efficacité et de productivité du processus de développement. L'organisation a un même modèle organisationnel depuis longtemps et a peu changé. En faisant ressortir que la situation actuelle n'est pas très efficace (si elle n'est pas efficace), l'entreprise devra prendre certaines décisions majeures à même la structure de l'organisation, car l'utilisation de bonne pratique ne sera pas suffisante pour améliorer la productivité.

Il faut aussi un certain changement de culture, car bien que l'on crée des documents d'analyse, suite à l'acceptation d'une solution, l'effort dans le document est perdu, car il n'est pas maintenu. Le fait que ces documents ne soient pas maintenus force constamment à se reposer les mêmes questions. Si le contexte de la maintenance du S3M est insuffisant pour prouver les avantages d'un bon processus de maintenance, il faut donc passer par la base de l'efficacité des livrables de l'entreprise.

 Université du Québec École de technologie supérieure		DOCUMENT NO.	DATE 2012-05-16	VERS 1.0
Cours MGL804	TITRE Étude du modèle de l'organisation de la maintenance	PAGE 24	PAGE 24	

5. Références

- April A., Abran A., Améliorer la maintenance du logiciel. Loze-Dion, 2006.