

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

REINGENIERIE D'UN SITE WEB EXISTANT

RAPPORT DE PROJET
PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN GÉNIE LOGICIEL

PAR
STÉLIE LUC

OCTOBRE 2013

REMERCIEMENTS

Je tiens à remercier tout particulièrement et à témoigner ma reconnaissance à Monsieur Normand Seguin, Mon directeur de projet, pour l'aide qu'il m'a apportée par ses commentaires pertinents et ses réflexions afin de réaliser ce projet.

Je tiens aussi à remercier très sincèrement mon co-directeur de projet Monsieur Alain April pour sa disponibilité, ses conseils méthodologiques, son soutien, sa bonne humeur et pour m'avoir fourni la documentation nécessaire pour accomplir ce travail. Je le remercie vivement pour le sujet qu'il m'a proposé et sa participation à l'élaboration de ce rapport, car j'ai pu élargir mes connaissances au niveau de la réingénierie logicielle.

Merci à vous tous !

TABLE DES MATIÈRES

LISTE DES FIGURES	IV
LISTE DES TABLEAUX	VI
LISTE DES ACRONYMES ET ABREVIATIONS.....	VII
RESUME.....	VIII
ABSTRACT	IX
CHAPITRE I.....	1
1.1 INTRODUCTION	1
1.2 PROBLEMATIQUE.....	2
1.2.1 <i>L'organisation</i>	2
1.2.2 <i>Questions de recherche</i>	3
1.3 DEFINITION DU PROJET	4
1.3.1 <i>Motivation</i>	4
1.3.2 <i>Objectif du projet</i>	4
1.3.3 <i>Utilisateurs</i>	5
1.4 METHODOLOGIE.....	5
1.4.1 <i>Portée du projet</i>	5
1.4.2 <i>Planification du projet</i>	6
1.4.3 <i>Exécution</i>	7
1.5 CONCLUSION.....	7
CHAPITRE II	
REVUE DE LITTERATURE	8
2.1 INTRODUCTION	8
2.1.1 <i>Définition</i>	8
2.1.2 <i>Importance de la maintenance</i>	9
2.1.3 <i>Compréhension de programme</i>	10
2.1.4 <i>Information exigée pour la compréhension de programme</i>	11
2.1.5 <i>Le modèle de processus de compréhension</i>	13
2.1.6 <i>Modèle mental</i>	14
2.1.7 <i>Les stratégies de compréhension de programme</i>	14
2.2 REINGENIERIE	15
2.2.1 <i>Rétro-ingénierie</i>	17
2.2.2 <i>Restructuration</i>	19
2.2.3 <i>Ingénierie</i>	19
2.2.4 <i>Réusinage</i>	19
2.2.5 <i>Types de problèmes rencontrés pendant la réingénierie des applications web</i>	20

2.3	CARACTERISTIQUES DES APPLICATIONS WEB	20
2.4	LES DEFIS DE LA COMPREHENSION DE PROGRAMME	21
2.4.1	<i>Utilisation absurde d'intelligence élevée (code délicat)</i>	21
2.4.2	<i>Différents styles de programmation</i>	22
2.4.3	<i>Le manque de mots dans le vocabulaire du domaine du logiciel</i>	22
2.4.4	<i>Représentation du programme</i>	22
2.4.5	<i>Commentaires insuffisants</i>	22
2.4.6	<i>Duplication de code</i>	23
2.4.7	<i>Identification de code mort</i>	23
2.5	AVANTAGES DE LA REINGENIERIE DES APPLICATIONS WEB	23
2.6	SYNTHESE	24
 CHAPITRE III		
REINGENIERIE DU SITE WEB EXISTANT		25
3.1	INTRODUCTION	25
3.2	RETRO-INGENIERIE	25
3.2.1	<i>Installation et configuration du package XAMPP</i>	25
3.2.2	<i>Lecture</i>	26
3.2.3	<i>Exécution</i>	26
3.2.4	<i>Ajout de la fonction « Nouvelles »</i>	28
3.3	MODIFICATION AU LOGICIEL	32
3.3.1	<i>Analyse d'impact des changements</i>	32
3.3.2	<i>Conception</i>	42
3.3.3	<i>Réalisation</i>	44
3.3.4	<i>Tests</i>	47
CONCLUSION		48
4.1	DIFFICULTES RENCONTREES	49
4.2	TRAVAUX FUTURS	49
4.3	CE QUI A ETE UTILE	50
4.4	LIEN ENTRE LE SUJET DU TRAVAIL ET LE GENIE LOGICIEL	51
RÉFÉRENCES		52
 ANNEXES		
ANNEXE A		55
A.1	PREREQUIS LOGICIELS POUR L'INSTALLATION DE PHP	55
ANNEXE B		61
B.1	DIAGRAMME DE CLASSES DE TOUTE L'APPLICATION	61

ANNEXE C.....	65
C.1 DIAGRAMME DE SEQUENCE « MODIFIER NOUVELLE».....	65
ANNEXE D	67
D.1 SCRIPT DE LA TABLE « NOUVELLES »	67
ANNEXE E.....	69
E.1 ÉCRAN DE SUPPRESSION DE NOUVELLE.....	69
ANNEXE F	70
F.1 CODE SOURCE POUR TESTER LA FONCTION VALIDATE (\$NOUVELLE) ET LA FONCTION GETBYNOUVELLEID () DE LA CLASSE NOUVELLEMANAGER	70

LISTE DES FIGURES

Figure 1 : Modèle de processus de compréhension	13
Figure 2 : Modèle général pour la réingénierie logicielle	16
Figure 3 : Vue commune de la réingénierie	17
Figure 4 : Écran de connexion au site d'administration.....	27
Figure 5 : Écran initial d'administration	27
Figure 6 : Modèle d'ajout, de modification et de suppression de nouvelles du logiciel.....	28
Figure 7 : Modèle représentant les classes d'accès à la base de données	29
Figure 8 : Diagramme de séquence Login	30
Figure 9 : Diagramme de séquence Ajouter nouvelle.....	31
Figure 10 : Impact de changement de la photo de la page d'accueil du site S3m	33
Figure 11 : Impact de changement de la photo de la page d'accueil du site S3m	33
Figure 12 : Impact de changement de retrait du wiki et l'agrandissement du blog et des labs	34
Figure 13 : Impact d'ajout de la fonction «Nouvelles »	36
Figure 14 : Impact d'ajout de la fonction « Nouvelles »	36
Figure 15 : Impact d'ajout de la fonction « Nouvelles »	37
Figure 16 : Impact d'ajout de la fonction « Nouvelles »	37
Figure 17 : Impact d'ajout de la fonction « Nouvelles »	38
Figure 18 : Impact d'ajout de la fonction « Nouvelles »	38
Figure 19 : Impact de changement de l'enlèvement des trois livres de maintenance	39
Figure 20 : Impact de changement de la sous fonction gens	40
Figure 21 : Architecture de l'application web	42
Figure 22 : Interface principale des utilisateurs du nouveau système.....	44
Figure 23 : Interface de la sous-fonctionnalité « Gens »	45
Figure 24 : Écran d'ajout de nouvelle.....	46
Figure 25 : Écran de modification de nouvelle	47
Figure 26 : Configuration des propriétés de MySQL Serveur	58
Figure 27: Admin propriétés	59
Figure 28 : Démarrage du serveur MySQL	60
Figure 29 : Connexion de MySQL Serveur	60
Figure 30: Diagramme de classes de l'application	61

Figure 31: Diagramme de classes de l'application	62
Figure 32: Diagramme de classes de l'application	63
Figure 33: Diagramme de classes de l'application	64
Figure 34: Diagramme de séquence de modification de nouvelle	65
Figure 35: Écran de suppression de nouvelle.....	69

LISTE DES TABLEAUX

Tableau 1: différences entre la réingénierie des applications web et les systèmes traditionnels	3
Tableau 2: Caractéristiques des applications web simples et avancées	21
Tableau 3: Tableau de synthèse des techniques	24
Tableau 4: Estimation d'efforts pour le nombre d'heures de travail	41
Tableau 5: Description des classes et pages de l'architecture de l'application web	43
Tableau 6: Tableau de lien entre le sujet du travail et le génie logiciel	51

LISTE DES ACRONYMES ET ABRÉVIATIONS

AQL	Assurance Qualité Logicielle
AMP	Apache, MySQL & PHP
HTML	Hyper Text Markup Language
IEEE	Institute of Electrical and Electronics Engineers
ISO/IEC	International Organization for Standardization/International Electrotechnical Commission
Livreaql	Livre assurance qualité logicielle
MVC	Modèle Vue Contrôleur
S3M	Software Maintenance Maturity Model
SQA	Software Quality Assurance
XAMPP	X Apache MySQL Perl PHP

RÉSUMÉ

De nos jours, la capacité d'un système logiciel à évoluer est fondamentale pour qu'il puisse rester plus longtemps sur le marché. Cependant, dans l'évolution d'un système logiciel le changement est inévitable. Pour implémenter ce changement, la compréhension de programme joue un rôle important dans l'évolution logicielle et la maintenance et ceci n'est pas une tâche facile. Pour atteindre ce niveau de compréhension, il faut trouver une stratégie. La réingénierie peut aider à résoudre ce problème de compréhension. Ainsi, ce rapport de projet présente une approche de réingénierie pouvant aider à améliorer ou de transformer un site web existant afin qu'il puisse être compris, contrôlé et utilisé à nouveau. Pour réaliser ce travail, nous avons étudié comment les programmeurs font pour comprendre un programme et passé en revue des concepts essentiels concernant la réingénierie, la maintenance, le réusinage (refactoring) afin de mieux comprendre les problèmes de compréhension de code et de jeter les bases pouvant nous permettre d'établir notre solution. Finalement le tout a été terminé par la rétro-ingénierie d'un site web existant visant à faciliter la compréhension de l'application dans le but d'obtenir plus d'information nécessaire sur son comportement pour pouvoir le modifier.

Mots clés : Réingénierie, Compréhension de programme, Maintenance, Retro-ingénierie

ABSTRACT

Today, the ability to evolve a software system is essential for it to remain on the market longer. However, change is inevitable in the evolution of a software system. To implement this change, understanding program plays an important role in software evolution and maintenance, and it is not an easy task. To achieve this level of understanding, we must find a strategy. Reengineering can help solve this problem of understanding. Thus, this project report presents an approach to reengineering that can help improve or transform existing web site so that it can be understood, controlled and used again. To accomplish this work, we studied how do programmers to understand a program and reviewing key concepts about reengineering, maintenance and refactoring to better understand the problems of understanding code and to lay the foundations that would allow us to build our solution. Finally it was all over by the reverse engineering of existing website to facilitate the understanding of the application in order to obtain more information on the behavior necessary to edit it.

Keywords: Reengineering, Program understanding, Maintenance, Reverse engineering

CHAPITRE I

INTRODUCTION

1.1 Introduction

En raison de nouvelles possibilités technologiques, commerciales et les commentaires des utilisateurs, les applications web ont tendance à évoluer rapidement et à subir fréquemment des modifications. La possibilité d'évoluer est fondamentale pour un système logiciel de rester plus longtemps sur le marché (Ricca et Tonella, 2001 ; Svahnerg, 2003). Par conséquent pour répondre à l'évolution des besoins des utilisateurs et les changements rapides dans la technologie, les systèmes logiciels doivent évoluer continuellement. Toutefois, dans l'évolution d'un logiciel, les changements sont inévitables et auront tendance à provoquer la dégradation de la structure de l'application. Pour réaliser ces changements, il est essentiel de faire des mises à jour ou des modifications dans le système existant (Ricca, Tonella et Baxter, 2002 ; Tilley et Smith, 1995).

Afin de réaliser les modifications dans le système existant, les ingénieurs logiciels ont besoin d'obtenir un ensemble de compétences nécessaires pour les comprendre. Ce qui est de plus en plus important, dont l'intérêt technique, est le développement dans le domaine de la compréhension de programme et le recouvrement de conception du système d'artefacts existants, tel que le code source. C'est un problème important dans un grand éventail de domaine d'application. De ce fait, les pratiques de maintenance n'ont pas réussi à garder la synchronisation entre la conception et les documents d'architecture avec les modifications apportées au code en cours d'exécution (Tilley et Smith, 1995).

Donc il est important de développer une stratégie pour effectuer ce changement afin de permettre l'évolution continue du logiciel. Alors la réingénierie donne la possibilité de comprendre et de transformer les actifs d'un système existant et aussi offre une approche permettant l'évolution du système (Tilley et Smith, 1995).

Le projet est de modifier l'application Web www.S3M.ca (Modèle d'évaluation de la capacité de la maintenance logicielle) qui est un site dédié aux programmeurs de la maintenance, afin de le transformer en autre site web appelé www.livreaql.org (livre d'assurance qualité logicielle). La transformation envisagée du site est au niveau de l'interface principale, des fonctionnalités existantes de l'application, des contenus des pages et de plus on insérera une nouvelle fonction : « **Nouvelles** ».

1.2 Problématique

1.2.1 L'organisation

Le www.S3M.ca est un site web dédié aux programmeurs de la maintenance qui sont dépassés par les demandes urgentes de maintenance, et qui souhaitent avancer pour améliorer le service qu'ils fournissent à leurs clients. De plus, il offre des services de coaching, de formation, de certification, et d'accès au matériel de recherche dans le domaine de la maintenance logicielle. Il possède des produits comme le logiciel d'évaluation rapide du S3M qui est encore à l'état de prototype. Il permet aussi de s'inscrire pour avoir accès à toutes les ressources (par exemple les publications, les outils pour logiciel, le wiki) de S3M, tel qu'un abonnement de niveau bronze, argent, or, ou partenaire dont le prix varie selon le type.

L'un des problèmes dominants dans ce projet est le besoin de transformer le site web du S3M en un site web de livreaql. L'option de réingénierie a été choisie dans le but de permettre de créer la documentation inexistante et de corriger des problèmes dans le système existant (s'il y en a). Nous avons réalisé une étude préalable du site actuel qui nous a permis de déterminer les changements aux fonctionnalités existantes ainsi que l'ajout de nouvelles. Cette transformation consiste à changer le logo, l'image principale, les fonctionnalités à propos de S3M, les services, les produits, enlever le wiki, les trois livres de maintenance, et l'insertion d'une fonction : « **Nouvelles** » permettant d'ajouter des nouvelles sur le site.

1.2.2 Questions de recherche

Contrairement aux systèmes logiciels traditionnels, les sites web sont des systèmes qui se composent de différentes technologies et de langages de programmation. Aussi les changements dans les sites web se réalisent plus fréquemment et plus radicalement (par exemple dans le contenu et les fonctionnalités) que pour les systèmes traditionnels (Gui, 2005).

Par contre, comprendre l'évolution et gérer la réingénierie d'un grand site web est aussi difficile que l'évolution des grands systèmes logiciels de taille comparable. La réingénierie des sites web diffère de la réingénierie des systèmes existants traditionnels (Kienle et Müller, 2001). Les principales différences trouvées entre la réingénierie des applications web et les systèmes existants traditionnels sont présentées dans le tableau 1 ci-dessous :

Tableau 1:différences entre la réingénierie des applications web et les systèmes traditionnels (Deshpande, Murugesan et al, 2002)

Differences between web applications reengineering and legacy systems	
1. Compressed development schedules	2. Constant evolution with shortened revision cycles
3. "Content is king ", i.e. it is integrated inextricably with procedural processing	4. Insufficient requirement specifications
5. Small teams working to very short schedules	6. Emerging technologies methodologies
7. Lack of accepted testing processes	8. User satisfaction and the threat from one's competition
9. Minimal management support	10. Criticality of performance.
11. Evolving standards to which Web applications should or must comply, depending on the specific circumstances (for example accessibility standards for government sites or IEEE or W3C standards for technological reasons).	12. Understanding of additional disciplines required for Web applications, such as hypertext , graphic design, information architecture
13. Security considerations	14. Legal, social and ethical issues
15. Variety of background of developers	16. Rapidly evolving implementation environment encompassing various hardware platforms

De plus, les applications web sont également caractérisées par une maintenance continue et d'évolution des opérations pour répondre aux nouvelles exigences fonctionnelles et non fonctionnelles dans le cadre d'évolution dans lequel elles sont utilisées. La compréhension logicielle est une phase importante dans le processus de maintenance (Grubb et Takang, 2003 ; Von Mayrhauser et Vans, 1995), en conséquence le problème de compréhension est la majeure partie du problème à résoudre et nous porte à formuler la question principale suivante : comment font les programmeurs pour comprendre un site web existant en vue de le modifier?

Pour répondre à la question principale, on peut soulever deux autres questions plus spécifiques :

- 1) Quelles sont les informations qu'un programmeur a besoin pour comprendre un programme?
- 2) Quels sont les problèmes auxquels un programmeur doit faire face lors de la compréhension de programme?

1.3 Définition du projet

La phase de définition a permis de déterminer la motivation, l'objectif, et les utilisateurs de ce projet.

1.3.1 Motivation

La motivation principale de ce projet est le besoin de faire la réingénierie du site web **S3M**. Donc des modifications dans le système existant ont été proposées. Afin de réaliser ce changement, nous essayons de comprendre comment les programmeurs font pour comprendre un site web existant afin de le modifier.

1.3.2 Objectif du projet

L'objectif du projet est d'apporter des modifications au site web **S3M** pour le transformer en un site web de « **livreaql** ». Pour atteindre l'objectif prévu, nous allons mettre

en œuvre une solution pour effectuer ce changement suivant les demandes du client. De ce fait, on propose de faire la réingénierie du site afin de comprendre, de transformer le site web existant (S3M) pour lui permettre de répondre aux exigences du client.

1.3.3 Utilisateurs

Les résultats de ce projet est un site web qui pourra être utilisé par les deux professeurs ou administrateurs qui sont les principaux clients du site web livreaql et seraient les utilisateurs pour mettre à jour les contenus, et tout le monde pourront consulter et utiliser en tant qu'utilisateur de ce site.

1.4 Méthodologie

Pour parvenir à l'objectif mentionné à la section précédente, nous utiliserons une approche de réingénierie qui comprend les phases suivantes : portée, planification, exécution.

1.4.1 Portée du projet

Ce projet consiste à réaliser des changements dans l'interface principale de l'application, d'ajouter, de modifier des fonctionnalités ainsi que les contenus du site web S3M tout en produisant un nouveau site qui sera le **www.livreaql.org**.

Les modifications à apporter dans l'interface principale du site web sont les suivantes :

- Remplacement du logo du site S3M par le logo de livreaql.
- Remplacement de l'image principale de la page d'accueil du site S3M par les photos des clients du site web de livreaql.
- Enlèvement des trois livres de maintenance et conservation des deux livres d'assurance qualité logicielle. Cette modification implique comme travail le retrait total de l'image initiale et la production d'une autre image similaire par rapport à la première affichant seulement les deux livres d'assurance qualité logicielle.
- Retrait de la fonction du wiki et agrandissement du blog et des labs internationaux.

Les fonctionnalités à modifier sont les suivants :

- La fonction « A propos de S3M en A propos de livreaql » ainsi que ces sous fonctionnalités.
 - ❖ Survol
 - ❖ FAQs
 - ❖ Gens
 - ❖ Nouvelles
 - ❖ Liens utiles

- Les produits et les services.

La nouvelle fonctionnalité qui sera ajoutée est la suivante :

- Permettre de mettre à jour les nouvelles dans un site d'administration (par l'administrateur du site).

Donc ce projet portera spécifiquement sur l'étude de la compréhension de programme d'un logiciel existant dans le contexte de la réingénierie, de maintenance logicielle, et réusinage (refactoring). On ne discutera pas des outils disponibles pour la compréhension de programme.

1.4.2 Planification du projet

Afin d'atteindre l'objectif fondamental du projet, ces étapes ont été identifiées :

- Revue de littérature sur la compréhension de programme et la réingénierie des sites web.
- Rétro-ingénierie du site web existant.
- Modification au logiciel (analyse d'impact des changements, réalisation, conception, tests).

1.4.3 Exécution

Dans cette phase, on procède à la description des modifications du logiciel. Pour cela, on effectue les étapes suivantes :

- Identification des techniques de rétro-ingénierie qui vont servir dans l'étude.
- Étude du système existant.
- Identification des endroits à modifier.
- Propagation de changements.

1.5 Conclusion

Dans cette phase, on tient compte de l'expérimentation, on fait le sommaire et le bilan du projet qui comprend :

- 1) Un bref rappel du travail réalisé.
- 2) Les difficultés rencontrées.
- 3) Une discussion des travaux futurs ou améliorations possibles.
- 4) Ce qui a été utile.
- 5) Un lien entre le sujet du travail et le génie logiciel

CHAPITRE II

REVUE DE LITTÉRATURE

2.1 Introduction

Ce chapitre présente une revue de littérature sur les concepts clés nécessaires à la réalisation du projet notamment la maintenance, la réingénierie, et le réusinage (refactoring). Ce chapitre est composé de quatre sections principales. La première section présente les principales définitions attribuées à ces concepts. Par la suite, un survol de la littérature sera fait où l'on étudiera sur la façon dont les programmeurs ont procédé pour la compréhension de programme à travers les thèmes de la réingénierie, de la maintenance, et de réusinage (refactoring). La troisième section présente les défis de la compréhension de programme. Finalement, on présente les avantages de la réingénierie logicielle et une synthèse résumant les trois premières sections.

2.1.1 Définition

Il existe plusieurs définitions de la réingénierie se rapportant habituellement aux parties d'un système logiciel. La définition utilisée dans ce document est celle de Chikofsky et Cross (1990). La **réingénierie** est l'examen et l'altération d'un système pour le reconstituer dans une nouvelle forme suivit de son implantation (Chikofsky et Cross, 1990).

La **maintenance** des logiciels est définie dans la norme IEEE 1219 comme une modification d'un produit logiciel après la livraison pour corriger les erreurs, pour améliorer les performances ou d'autres attributs, ou pour adapter le produit à un environnement modifié (Bennett et Rajlich, 2000).

Cependant, une autre définition similaire a été donnée par la norme ISO / IEC, encore une fois en insistant sur le caractère post livraison : le produit logiciel subit une modification au code et la documentation connexe en raison d'un problème ou un besoin d'amélioration. L'objectif est de modifier le logiciel existant, tout en préservant son intégrité (Bennett et Rajlich, 2000).

Le **réusinage** (refactoring) est une technique pour restructurer le code existant, pour modifier sa structure interne sans changer¹ son comportement externe (Fowler, 2002).

2.1.2 Importance de la maintenance

Pour garder les logiciels opérationnels et satisfaire pleinement les besoins des utilisateurs, la maintenance a une importance significative. La maintenance doit être faite pour corriger les erreurs, pour prévenir le logiciel des risques et d'apporter des modifications si le matériel est changé. Toutefois, la maintenance est nécessaire pour mettre à jour un logiciel ou l'adapter en conséquence des demandes de changement des utilisateurs. Parfois, des mises à jour dans la documentation dans le code et les bases de données sont nécessaires, par conséquent la maintenance doit être faite pour traiter les problèmes de performance du logiciel ou les besoins d'amélioration de fonctionnalité (Grubb et Takang, 2003).

D'après Corbi (1989), il existe quatre types de maintenance :

1. **Maintenance corrective (20%)** : c'est la correction des fautes et des erreurs qui sont découvertes après que le logiciel ait été livré, ou la cause de ces défauts pourrait être les erreurs de conception, les erreurs logiques ou de codages.
2. **Maintenance adaptative (25%)** : qui est appliquée lorsque les exigences ou l'environnement d'un programme sont modifiés tels que le système d'exploitation ou le matériel.

¹ Dans le projet, nous n'aurons pas à changer le comportement des fonctionnalités du site web (par exemple l'utilisabilité et la performance).

3. **Maintenance perfective** (50% ou plus) : type de maintenance qui traite les changements rapides des exigences et des améliorations provenant des utilisateurs comme des insertions, des modifications et des suppressions.
4. **Maintenance préventive** (5%) : type d'activité qui améliore la maintenabilité et la fiabilité future du système et fournit une base pour l'amélioration à l'avenir.

2.1.3 Compréhension de programme

Une application web est un système logiciel, où la plupart de ses fonctionnalités sont livrées à travers le web. Un site web peut contenir de multiples applications web. Les applications web consistent principalement en un ensemble de pages web statiques ou dynamiques. De nos jours, ces applications sont des systèmes complexes et hétérogènes, la plupart d'entre eux sont des sites web dynamiques contenant des mélanges de programmes qui génèrent dynamiquement des hyperdocuments (Gui, 2005). Pour les modifier, nous avons besoin d'acquérir une compréhension fondamentale de ces programmes.

La compréhension de programme est un processus permettant de savoir ce que le système logiciel fait, comment il est relié à son environnement, afin d'identifier où les changements devront être effectués (Grubb et Takang, 2003). La compréhension de programme est un facteur important dans la réalisation efficace de la maintenance du logiciel et permet l'évolution réussie des systèmes informatiques (Von Mayrhauser et Vans, 1995). La compréhension d'un logiciel est le prérequis dans tous les processus de maintenance et d'évolution d'un système existant (Rajlich, 1997).

Régulièrement, on cite les activités de la compréhension de programme en tant que la tâche la plus difficile et la plus longue pour le personnel de maintenance. On a évalué le pourcentage de temps consacré dans les activités de la compréhension de programme entre 40% à 50% et même à 90% (Tilley et Smith, 1995).

Se familiariser avec le code source n'est pas seulement pour la compréhension de programme. Mais il y a quelques autres artefacts importants qui doivent être connus pour

comprendre le logiciel. Ces artefacts peuvent être des spécifications formelles, des diagrammes de classe, des diagrammes de contexte, de flux de données, l'énoncé de code source. Ces artefacts sont très utiles pour comprendre l'architecture et la conception des logiciels (Grubb et Takang, 2003).

2.1.4 Information exigée pour la compréhension de programme

Il est important de disposer des informations détaillées au sujet de certains aspects du logiciel tels que le domaine de problème, l'exécution d'effet, les relations de cause à effet, les relations de produit-environnement et les caractéristiques d'aide à la décision pour avoir une meilleure compréhension du logiciel. Ces informations peuvent être obtenues de plusieurs sources telles que dans la documentation du système, le code source du programme, les documents de spécification, de conception et d'architecture (Grubb et Takang, 2003).

2.1.4.1 Problème de domaine

On définit le problème comme l'état de difficulté qui doit être résolu et le domaine comme le secteur particulier qui est concerné par la résolution du problème. Dans le processus de maintenance du logiciel et d'évolution, une bonne connaissance du domaine de problème est très importante. Il permet à l'équipe de gestion de sélectionner des ressources appropriées à savoir des compétences d'expertise, les algorithmes, les outils, et les techniques (Grubb et Takang, 2003).

2.1.4.2 Effet d'exécution

Le but de l'effet d'exécution est de vérifier si le changement demandé est mis en application et des résultats prévus sont atteints. L'effet d'exécution est une représentation de niveau abstraite du comportement de système pour un ensemble spécifique des entrées. On le classifie par catégorie en deux niveaux d'abstraction qui sont le haut niveau et le bas niveau. Les résultats globaux des systèmes sont considérés dans le haut niveau d'abstraction alors qu'on étudie pour le comportement spécifique la contribution de module individuelle dans le

bas niveau d'abstraction. Le flux² de contrôle et le flux de données doivent être envisagés pour la réalisation de cette activité (Grubb et Takang, 2003).

2.1.4.3 Relations de cause à effet

Le but des relations de cause à effet est de découvrir les effets des changements prévus sur les autres parties du système. Il contribue également à la prédiction des effets de propagation potentielle. La découverte de la nature des dépendances dans un système a été une autre contribution majeure dans le cas des relations de cause à effet parce que si le changement est petit ou grand, les dépendances causent toujours de nombreuses difficultés et des erreurs (Grubb et Takang, 2003).

2.1.4.4 Relations de produit-environnement

Le but des relations de produit-environnement est de découvrir tous les facteurs externes qui pourraient avoir des incidences sur le système. Les facteurs susceptibles qui peuvent influencer sur un système pourraient être des règles métiers, les réglementations gouvernementales, des logiciels / plates-formes d'exploitation du matériel (Grubb et Takang, 2003).

2.1.4.5 Caractéristiques d'aide à la décision

Ce sont les caractéristiques qui stimulent le pouvoir de décision technique et managériale de l'équipe de maintenance tout en rendant des décisions relatives aux options d'analyse, à la budgétisation et aux allocations de ressources. Ces caractéristiques impliquent la complexité, la maintenabilité (Grubb et Takang, 2003).

² Un flux est une communication avec une extraction appropriée, une traduction, un filtrage et une abstraction, d'un objet de connaissance disponible dans un aspect ou sous-aspect d'une collaboration (Jastroch, Marlowe et al, 2011).

2.1.5 Le modèle de processus de compréhension

Le programmeur doit choisir le processus et l'outil qui conviennent le mieux à son niveau d'expertise et aux circonstances de la tâche en cours assignée. Pour comprendre un programme, trois actions peuvent être prises: lire le programme, c'est-à-dire la documentation, lire le code source et exécuter le programme, par exemple observer l'exécution, obtenir la trace des données, étudier le stockage dynamique. La documentation peut être une aide excellente ou peut nous induire en erreur. Dans l'exécution d'un programme, l'étude de comportement dynamique peut être très utile et peut nettement améliorer la compréhension en indiquant les caractéristiques du programme qui ne peuvent pas être identifiées en lisant seulement le code source. Cependant, le code source est habituellement la source d'information primaire (Corbi ,1989). La figure 1 illustre ces actions avec des exemples.

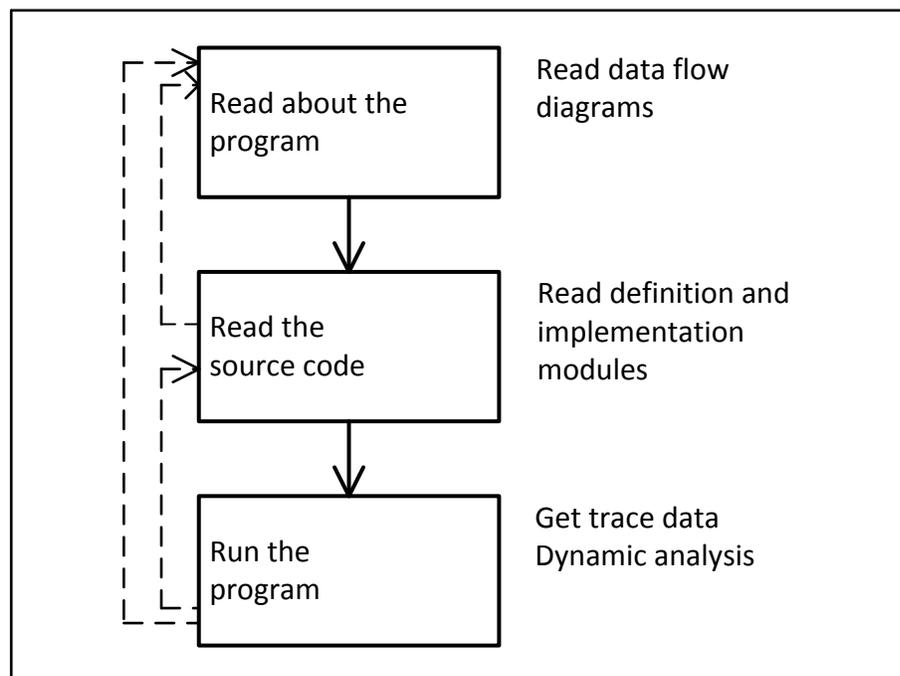


Figure 1 : Modèle de processus de compréhension

Source : Grubb et Takang (2003)

2.1.6 Modèle mental

Le modèle mental est une représentation interne et fonctionnelle du logiciel à l'étude. Il se compose des entités statiques telles que les structures de textes, les plans, les hypothèses, les balises et les règles du discours. Les plans de haut niveau sont raffinés dans des versions plus détaillées ou des structures de textes. Chaque structure de textes à son tour représente une abstraction de plus haut niveau des autres unités d'information ou des structures de textes. La construction des structures de textes associe plusieurs comportements dynamiques, y compris des stratégies, des actions, des épisodes, et des processus (Von Mayrhauser et Vans, 1995).

Le modèle mental est conçu après l'observation, l'inférence ou l'interaction avec le système cible. Il change sans interruption pendant que plus d'informations sur le système cible sont obtenues. Sa complétude et exactitude peuvent être influencées par des facteurs tels que l'expérience précédente de l'utilisateur avec les systèmes semblables et la formation technique (Grubb et Takang, 2003).

Le modèle mental peut contenir des informations insuffisantes, contradictoires ou inutiles sur le système cible. Cependant, il n'est pas essentiel que ce modèle soit complet, il doit donner l'information clé au sujet du système visé (Grubb et Takang, 2003).

2.1.7 Les stratégies de compréhension de programme

Les chercheurs ont entrepris beaucoup d'études, plusieurs modèles cognitifs ont été développés pour expliquer comment les programmeurs comprennent le code source durant la maintenance. Donc, il existe différentes stratégies que les programmeurs utilisent pour comprendre les programmes. Parmi lesquelles on trouve le top-down, le bottom-up et l'opportuniste.

Top-down : la compréhension de programme se fait de haut en bas, en reconstruisant les connaissances du domaine d'application tout en les combinant au code source. Cette stratégie commence par une hypothèse globale sur le programme. Cette hypothèse initiale est raffinée en une hiérarchie d'hypothèse secondaire. La vérification ou le rejet d'une hypothèse dépend fortement de la présence ou l'absence des balises (Storey, Wong et Muller, 1997).

Bottom-up : la compréhension de programme se fait par la lecture du code source, puis mentalement en découpant les artefacts de bas niveau du logiciel dans des abstractions significatives et de plus haut niveau. Ces abstractions sont encore rassemblées jusqu'à ce qu'une compréhension de haut niveau du programme soit formée (Storey, Wong et Muller, 1997).

Opportuniste : c'est la combinaison des deux stratégies top-down et bottom-up. Cette théorie comporte trois volets (Storey, Wong et Muller, 1997) :

- 1) Une base de connaissance qui représente l'expérience de base du programmeur et son expertise de programmation.
- 2) Un modèle mental qui représente la compréhension actuelle du programmeur du programme.
- 3) Un processus d'assimilation qui décrit comment le modèle mental évolue en utilisant la base de connaissance du programmeur et les renseignements sur le programme tel que la structure, les noms des modules, les variables, les fonctions, les procédures, etc.

2.2 Réingénierie

La réingénierie est le moyen par lequel un système est amélioré en utilisant d'abord la rétro-ingénierie pour faciliter la compréhension et permet à un élément de la restructuration, puis l'ingénierie pour produire le nouveau système amélioré (Grubb et Takang, 2003). Elle est l'approche la plus utilisée pour permettre la maintenance évolutionnaire donc elle est étroitement liée à la maintenance. Son objectif est de comprendre le logiciel existant c'est-à-dire la spécification, la conception, l'implémentation, puis à nouveau le mettre en œuvre pour améliorer la fonctionnalité du système, la performance ou l'implémentation. Il y a quatre objectifs de réingénierie : la préparation pour l'amélioration fonctionnelle, l'amélioration de la maintenabilité, la migration, et l'amélioration de la fiabilité (Rosenberg et Hyatt, 1996). La figure 2 fait référence au modèle général de la réingénierie logicielle et illustre la rétro-ingénierie et l'ingénierie.

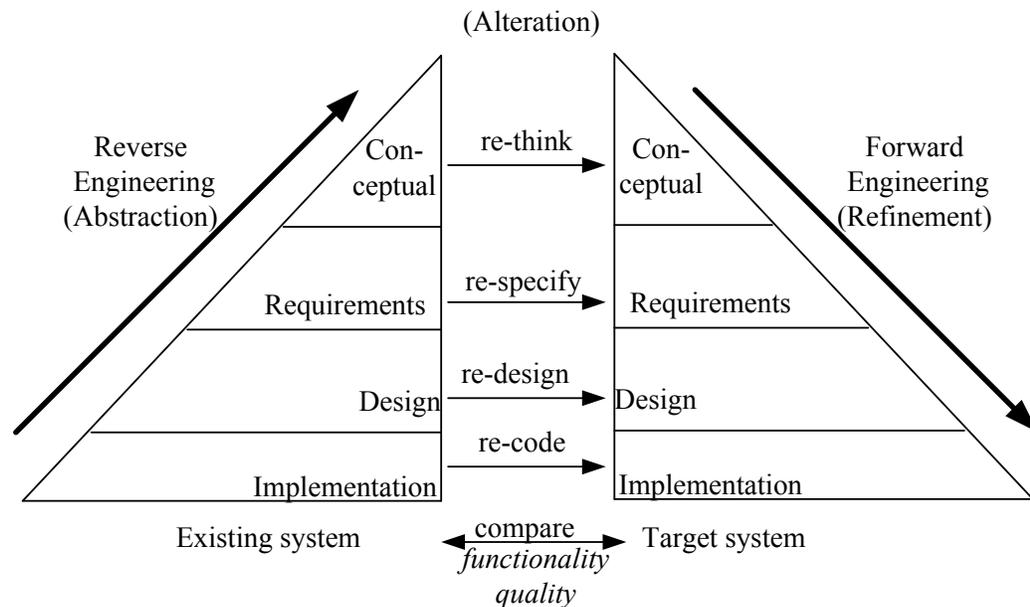


Figure 2 : Modèle général pour la réingénierie logicielle

Source : Rosenberg et Hyatt (1996)

La figure ci-dessus correspond à trois principes de réingénierie : l'abstraction, l'altération et le raffinement. L'abstraction est une augmentation progressive du niveau abstrait du système. Elle produit une représentation qui en souligne les caractéristiques du système par la suppression de certaines informations sur les autres. Le mouvement à la hausse que l'on appelle rétro-ingénierie est lié à des sous-processus, des outils et des techniques. L'altération est l'élaboration d'un ou plusieurs changements à une représentation du système sans changer le degré d'abstraction, y compris l'addition, la suppression et la modification de l'information, mais pas la fonctionnalité. Le raffinement est la diminution progressive du niveau d'abstraction de la représentation du système et est provoqué par la substitution successive d'information du système existant avec des informations plus détaillées. L'ingénierie qui ressemble au développement de logiciel de nouveau code, mais avec quelques améliorations de processus. Toutefois, pour la respécification des exigences,

les techniques de rétro-ingénierie doivent être appliquées pour la mise en œuvre et la conception pour obtenir les caractéristiques fonctionnelles (Rosenberg et Hyatt, 1996).

2.2.1 Rétro-ingénierie

Yang et Ward (2003), définit la rétro-ingénierie comme un processus d'analyse d'un système afin d'identifier ses composants, leurs interrelations et de créer des représentations du système sous une autre forme ou d'un niveau plus élevé d'abstraction (Yang et Ward, 2003). La rétro-ingénierie n'inclut pas la modification du système logiciel. On utilise la rétro-ingénierie pendant le processus de réingénierie logicielle (capture de modèle), dans le cadre d'un effort visant à la documentation d'un système, ou comme une tentative d'extraction des composants réutilisables à partir du logiciel (Bär et al, 1999). La figure 3 suivante montre la vue commune de la réingénierie.

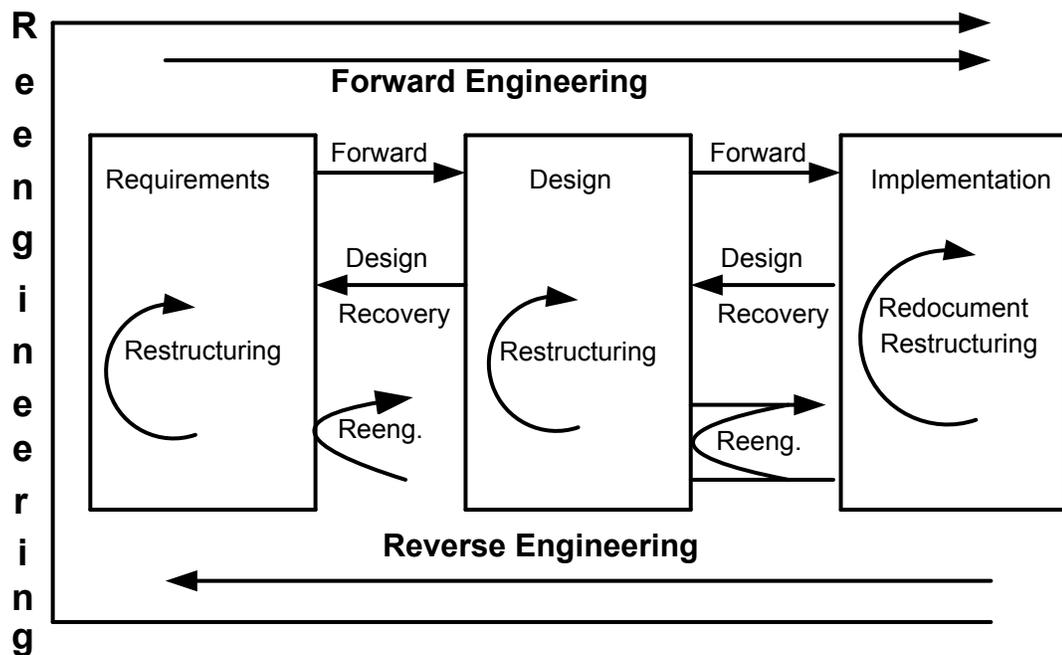


Figure 3 : Vue commune de la réingénierie

Source : Feiler (1993)

Cependant l'objectif principal de la rétro-ingénierie d'un système logiciel est d'accroître la compréhension globale du système à la fois pour la maintenance et pour de nouveau développement. Mais derrière l'objectif principal, on trouve six sous-objectifs clés qui sont : faire face à la complexité, générer des vues alternatives, récupérer les informations perdues, détecter des effets secondaires, synthétiser plus des abstractions, et faciliter la réutilisation (Chikofsky et Cross, 1990).

En règle générale, la rétro-ingénierie nécessite l'extraction des objets de conception et la construction ou la synthèse des abstractions qui sont moins dépendantes de l'implémentation. Elle n'implique pas de modifier le système objet ou la création d'un nouveau système basé sur le système soumis à la rétro-ingénierie. Il s'agit d'un processus d'examen, pas un processus de changement ou de réplique. Il contient plusieurs sous domaines parmi lesquels on peut citer deux types les plus importants : la redocumentation et le recouvrement de conception (Chikofsky et Cross, 1990).

Redocumentation : est la création ou la révision d'une représentation sémantiquement équivalente dans le même niveau relatif d'abstraction. Les formes de représentations résultantes sont généralement considérées comme des vues alternatives (par exemple, flux de données, structure de données, et flux de contrôles) destinées à la compréhension humaine. La redocumentation est la forme la plus simple et la plus ancienne de rétro-ingénierie et peut être considérée comme une forme non envahissante et à faible niveau de restructuration (Chikofsky et Cross, 1990).

Recouvrement de conception : est un sous-ensemble de la rétro-ingénierie dans laquelle la connaissance du domaine, l'information externe, et la déduction ou le raisonnement flou sont ajoutés aux observations significatives de haut niveau d'abstraction sur le système au-delà de celles qu'on obtiendrait directement en examinant le système lui-même. Le recouvrement de conception recrée des abstractions de conception à partir d'une combinaison de code, de documentation existante de conception (si disponible), l'expérience personnelle, et la connaissance générale sur le problème et les domaines d'application (Chikofsky et Cross, 1990).

Rétro-ingénierie des données : est l'utilisation de techniques structurées pour reconstituer les actifs de données d'un système existant. Son objectif est de créer un environnement de données gérées (Davis et Alken, 2000).

2.2.2 Restructuration

La restructuration est la transformation d'une forme de représentation vers une autre forme ayant le même niveau d'abstraction relative tout en préservant le comportement externe du logiciel (fonctionnalité et sémantique). La restructuration est souvent utilisée comme une forme de maintenance préventive afin d'améliorer l'état physique du système, sa structure comme la modification du code dans le sens traditionnel de conception structurée. Il peut aussi s'agir de réglage du système sous réserve de répondre aux nouvelles contraintes environnementales qui ne comportent pas une réévaluation au niveau des abstractions plus élevées (Chikofsky et Cross, 1990).

2.2.3 Ingénierie

L'ingénierie est le processus traditionnel de passer d'un haut niveau d'abstraction et de conception détaillée vers le développement physique du système (Chikofsky et Cross, 1990). Elle consiste à employer des techniques génératives et transformationnelles pour l'obtention automatique du code source en fonction de nouvelles plates-formes ou des langages de programmation (Pérez-Castillo et al, 2011).

2.2.4 Réusinage

Le réusinage (refactoring) est habituellement un petit changement au logiciel, bien qu'il puisse impliquer d'autres changements. Tout d'abord, le but du réusinage consiste à faire le logiciel plus facile à comprendre et à modifier. Sans le réusinage, la conception du programme se détériore. Les changements apportés dans un code pour réaliser des modifications à court terme sans une pleine compréhension de la conception du code provoquent la perte de sa structure. Il devient plus dur pour voir la conception en lisant le code. Donc le réusinage est plutôt une restructuration du code en l'aidant à maintenir sa forme. Le réusinage permet de rendre le code plus lisible. De plus, on l'utilise aussi pour aider à comprendre du code non familier (Fowler et Beck, 1999).

2.2.5 Types de problèmes rencontrés pendant la réingénierie des applications web

Dans ce genre de projets, on peut rencontrer un certain nombre de problèmes lors de la réingénierie de ces applications. Généralement, on trouve des problèmes architecturaux et de conception. Parmi ces problèmes on peut citer (Ducasse, 2003 ; Bär et Ducasse, 1999) :

- **Manque de documentation** : la documentation, soit n'existe pas, ou est incompatible avec la réalité.
- **Stratification incorrecte** : manque de stratification ou inappropriée ce qui entrave la portabilité et l'adaptabilité.
- **Manque de modularité** : couplage fort entre les modules ce qui entrave l'évolution.
- **Fonctionnalité dupliquée** : les fragments de code peuvent être dupliqués, rendant ainsi la fonctionnalité liée délocalisée, par conséquent difficile à réutiliser et conduit à des cauchemars de maintenance.

2.3 Caractéristiques des applications web

Les sites web partagent de nombreux points communs avec les systèmes existants traditionnels, mais en même temps exposent des caractéristiques uniques telles que : une seule instanciation, non-stop, tolérance aux pannes, et progressivement mis à jour (Kienle et Müller, 2001 ; Eichmann, 1999). Les caractéristiques des applications web simples et avancées sont interprétées dans le tableau 2 suivant :

Tableau 2 : Caractéristiques des applications web simples et avancées (Deshpande, Murugesan et al, 2002)

Table 2: Characteristics of Simple and Advanced web Applications	
Simple web-Based systems	Advanced web-Based systems
<ul style="list-style-type: none"> ▪ Primarily textual information in non-core applications ▪ Information content fairly static ▪ Simple navigation ▪ Infrequent access or limited usefulness ▪ Limited interactivity and functionality ▪ Stand alone systems ▪ High performance not a major requirement ▪ Developed by a single individual or by a very small team ▪ Security requirements minimal(because of mainly one-way flow of information) ▪ Easy to create ▪ Feedback from users either unnecessary or not sought ▪ Web site mainly as an 'identity' for the current clientele, and not as a medium for communication 	<ul style="list-style-type: none"> ▪ Dynamic web pages because information changes with time and user's needs ▪ Large volume of information ▪ Difficult to navigate and find information ▪ Integrated with database and other planning, scheduling and tracking systems ▪ Deployed in mission-critical applications ▪ High performance and continuous availability is a necessity ▪ May require a larger development team with expertise in diverse areas ▪ Calls for risk or security assessment and management ▪ Needs configuration control and management ▪ Necessitates project plan and management ▪ Requires a sound development process and methodology ▪ User satisfaction vital ▪ Web site/ application as the main communication medium between the organization and users

2.4 Les défis de la compréhension de programme

2.4.1 Utilisation absurde d'intelligence élevée (code délicat)

On a constaté que des programmeurs écrivent du code délicat qui est très difficile à comprendre par les autres personnes, qui plus tard cause des problèmes de compréhension dont les intentions sont diverses, soit pour montrer leur intelligence ou pour garder la sécurité de leur emploi (Akhlq, 2010).

2.4.2 Différents styles de programmation

Les styles de programmation se diversifient de société en société et de programmeur à programmeur. Dès fois, il est impossible à un programmeur ou à un petit groupe de programmeurs d'écrire un grand programme. Les sociétés de développement de logiciel emploient l'externalisation de développement de logiciel où les programmeurs peuvent complètement contribuer au développement du logiciel, cela est dû en raison des coûts élevés de développement des fournisseurs de services de développement de logiciels locaux. Donc, l'écriture d'un grand logiciel est faite par un groupe de programmeurs ayant des compétences et des expériences de programmation différentes. En conséquence, le produit développé résulte d'une combinaison de différents styles de programmation qui rend le logiciel très difficile à comprendre (Akhlaq, 2010).

2.4.3 Le manque de mots dans le vocabulaire du domaine du logiciel

L'information du domaine de problème est requise pour avoir la compréhension du niveau abstrait en maintenance de logiciel. L'emploi des mots du domaine dans le code source, la documentation et les manuels de l'utilisateur sont envisagés comme une bonne pratique. Donc, le manque de mots du domaine dans le vocabulaire du logiciel est un autre problème dans la compréhension (Akhlaq, 2010).

2.4.4 Représentation du programme

Le bloc de code mal structuré et le type complexe de flux de contrôle rendent la compréhension plus complexe. Les dépendances non évidentes ou peu claires peuvent rendre plus difficile pour le programmeur de comprendre le code. Alors le regroupement des déclarations connexes et des modules pourraient rendre plus facile le processus de la compréhension. Le code peut être plus compréhensible et modifiable par la bonne mise en page visuelle en l'absence de l'auteur (Akhlaq, 2010).

2.4.5 Commentaires insuffisants

Le manque de commentaires et des commentaires périmés sont les principales causes dans l'augmentation du coût de maintenance et de réduction de la compréhension. Une recherche de (Fluri et al, 2007), a révélé que les commentaires et le rapport de code source

ne demeurent pas stables dans l'évolution de code source. Ainsi, il fallait avoir un processus pour faire la coévolution du code source et des commentaires pour formuler leur mise à jour. Cela augmentera l'avantage pour la compréhension (Akhlaq, 2010).

2.4.6 Duplication de code

La duplication de code est un autre problème commun qui rend plus complexe le processus de compréhension de logiciel. La duplication de code pousse la charge entière vers les activités qui peuvent être exercées plus tard dans le logiciel, notamment la maintenance et l'évolution du logiciel. La duplication de code rend la maintenance et l'évolution plus difficile et coûteuse. Bien que plusieurs outils aient été conçus pour la détection de la duplication de code, un bloc de code qui a été copié semble rarement semblable à l'original (Akhlaq, 2010).

2.4.7 Identification de code mort

La maintenance et l'évolution des systèmes par des programmeurs différents à plusieurs reprises pour la plupart contiennent une quantité notable de code mort. Le code mort pourrait comporter des méthodes et des classes qui ne sont plus utilisables, mais l'identification de ces classes et les méthodes n'est pas une tâche facile. Par conséquent, quand un ancien système subit de la maintenance, le code mort devient l'un des problèmes majeurs dans la compréhension du logiciel (Akhlaq, 2010). En outre, le code mort peut avoir des conséquences sur la compréhensibilité, la lisibilité et la taille du code source en augmentant l'effort et le coût de la maintenance (Scanniello, 2011).

2.5 Avantages de la réingénierie des applications web

Il y a un certain nombre d'avantages qui pourraient découler de la réingénierie des applications web qui sont les suivants (Canfora et Cimitile, 2002 ; Olsem, 1998) :

- Aide à la réduction de risque dans le redéveloppement et l'évolution des logiciels.
- Rend le logiciel plus facile à modifier.
- Améliore leur réutilisabilité et leur efficacité.
- Réduit le risque et leurs coûts de maintenance.

2.6 Synthèse

Ce chapitre a mis l'accent et la clarification sur le processus de compréhension de programme. Ainsi le processus de compréhension de programme a une grande importance dans notre étude afin de permettre de comprendre un site web existant pour pouvoir le modifier. Ce chapitre a fait ressortir l'importance de la compréhension de programme comme étant le principal facteur dans la réalisation de la maintenance efficace des logiciels et l'évolution réussie de systèmes informatiques. On a fait un survol de certaines littératures ou une description des informations exigées et les stratégies pour la compréhension de programme ont été fournies. Par la suite, on a fait un survol dans le domaine de la réingénierie qui est une approche la plus utilisée pour permettre la maintenance évolutionnaire, elle est étroitement liée à la maintenance d'où son objectif de comprendre le logiciel existant, et comprend trois étapes : la rétro-ingénierie, la restructuration et l'ingénierie. Les problèmes rencontrés dans ce genre de projets et les caractéristiques des applications web ont été présentés. De plus, les défis de la compréhension de programme incluant les problèmes de compréhension de code trouvés dans la littérature ont été rapportés ainsi que les avantages apportés par la réingénierie des applications web. Le tableau de synthèse qui suit fait un inventaire des techniques que nous allons utiliser dans la réingénierie du système existant.

Tableau 3 : Tableau de synthèse des techniques

Techniques	Application
Problème de domaine	oui
Effet d'exécution	oui
Relations de cause effet	oui
Relations de produit - environnement	non
Caractéristiques d'aide à la décision	non

CHAPITRE III

RÉINGÉNIERIE DU SITE WEB EXISTANT

3.1 Introduction

Ce chapitre a pour but de présenter l'approche utilisée dans le contexte de ce projet. Il comprend deux sections principales, la section 3.2 porte sur l'application de l'approche rétro-ingénierie et la section 3.3 décrit les modifications apportées au logiciel. Comme mentionné dans l'introduction, le but du projet est d'apporter des modifications au site web S3M pour le transformer en un site web de livreaql. Plusieurs approches facilitant la réingénierie des applications web ont été étudiées au chapitre 2.

Afin d'apporter les changements dans les fonctionnalités du système S3M, nous avons opté pour la rétro-ingénierie qui est un processus de réingénierie permettant d'acquérir une pleine compréhension de l'application dans le but d'obtenir plus d'information nécessaire sur son comportement pour pouvoir le modifier.

3.2 Rétro-ingénierie

Dans le cadre de ce projet, l'objectif de la rétro-ingénierie consiste à favoriser la compréhension et à récupérer à partir du code source de l'application web S3M les informations qui pourraient alors permettre d'accomplir les modifications plus efficacement. Ces informations comprennent la spécification fonctionnelle de toutes les exigences implémentées par l'application.

3.2.1 Installation et configuration du package XAMPP

Pour l'expérimentation de ce projet, nous avons choisi d'utiliser le package AMP pour avoir tous les paramètres de configuration du moteur PHP, du serveur HTTP Apache, et pour

automatiquement définir le serveur de base de données MySQL³. L'annexe A contient en détail les étapes d'installation et de configuration du package XAMPP et des autres prérequis logiciels pour l'installation de PHP.

3.2.2 Lecture

Vu qu'on ne possédait pas de documentation pour comprendre le système, on a opté pour la technique de lecture. Cette technique peut être définie comme une série de mesures uniques pour l'analyse textuelle d'un logiciel pour réaliser la compréhension nécessaire pour une mission particulière (Shull et al, 2003). On a utilisé cette technique pour détecter les défauts dans le code du système existant. Cette technique nous a permis de comprendre ce que fait le système et d'abstraire des informations importantes dans le système S3M.

Au cours de cette étape, on a obtenu les vues d'ensemble et locales du programme. Cette vue globale permet d'obtenir une compréhension de haut niveau du système et aussi permet de déterminer également l'étendue de l'effet de changement pourrait avoir sur d'autres parties du système (Grubb et Takang, 2003). La vue locale nous a donné la possibilité de concentrer notre attention sur une partie spécifique du système qui est le fichier de code source « **AdminUsers** ». À partir de cette vue, on a obtenu des informations sur la structure de système S3M, les types de données. Cette technique sera utile pour la maintenance ainsi que pour la construction du nouveau système à partir des composants réutilisables (Grubb et Takang, 2003).

3.2.3 Exécution

Le but de cette étape est d'étudier le comportement dynamique du programme en action, incluant l'exécution du programme pour obtenir des traces de données. Pour lancer l'application web, on a procédé comme suit :

- Premièrement dans **NetBeans IDE** on doit faire une clique droite sur `index.html` ou `index_fr.html`, ensuite cliquer sur « Run » pour afficher l'interface principale en anglais ou en français du site web S3M. Afin d'examiner l'exécution des pages par

³ <https://netbeans.org/kb/docs/php/configure-php-environment-windows.html>

le navigateur. Pour enregistrer les événements observés, c'est-à-dire les résultats de l'exécution comprenant la visualisation des cadres/formes/pages « A propos de S3M, Services, Produits » et la soumission de leurs formes. Elle est nécessaire pour récupérer le contenu réel des pages clientes construites dynamiquement de l'application web. En outre, elle est essentielle pour en déduire les liens entre les pages qui partagent les mêmes menus.

- Deuxièmement pour afficher l'interface d'administration de l'application on doit faire une clique droite sur index.php dans NetBeans IDE ensuite cliquer sur « Run » pour afficher l'écran de connexion à l'application web (voir figure 4). Saisissez comme E-mail «lucfigaro@gmail.com » et comme mot de passe « stelie » et cliquez sur le bouton « Login » pour accéder à l'écran principal d'administration (voir figure 5) afin d'obtenir les différents affichages des pages ainsi que leurs dépendances.

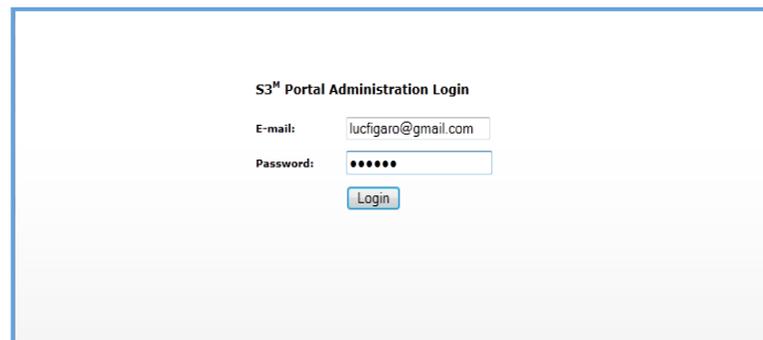


Figure 4 : Écran de connexion au site d'administration

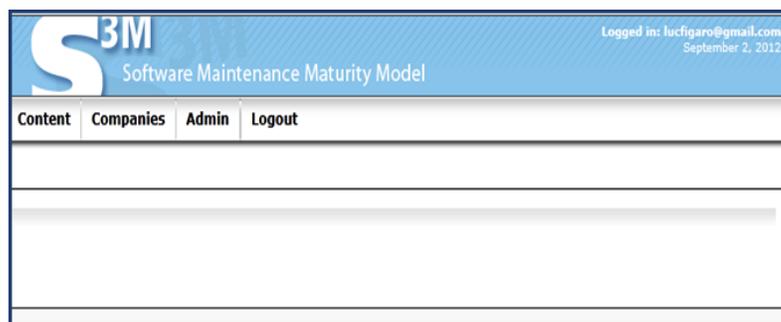


Figure 5 : Écran initial d'administration

3.2.4 Ajout de la fonction « Nouvelles »

Cette section présente les diagrammes de classes et de séquences pour la fonction « Nouvelles ».

- **Diagramme de classes**

Ce modèle montre une vue détaillée de la classe « **NouvelleManager** » avec leurs variables et leurs méthodes.

- **Modèle représentant la fonction d'ajout, de modification et de suppression de nouvelles du logiciel.**

NouvelleManager
+ NOUVELLE_Content :var = 'content' {readOnly} + NOUVELLE_Date :var = 'date' {readOnly} + NOUVELLE_Language :var = 'Language' {readOnly} + NOUVELLE_Title :var = 'title' {readOnly}
+ add(NouvelleBase) :var + delete(HashMap) :var + deleteByNouvelleId(var) :var + edit(NouvelleBase) :var + get(HashMap, var) :var + getByLanguage(var) :var + getByNouvelleId(var) :var + getLanguageNouvelle(var) :var - validate(var) :var

Figure 6 : Modèle d'ajout, de modification et de suppression de nouvelles du logiciel

Ce modèle contient la classe permettant l'ajout d'une nouvelle dans le système. On peut ajouter, modifier et supprimer une ou plusieurs nouvelles dans le système. C'est d'ailleurs le rôle de la méthode « **add(NouvelleBase)** » dans la classe « **NouvelleManager** » permettant d'ajouter des nouvelles. Après l'ajout de ces nouvelles dans le système les utilisateurs peuvent les consulter sur le site web. Nous présentons en grande partie à l'annexe B, le diagramme de classes de toute l'application.

○ **Modèle représentant les classes d'accès à la base de données**

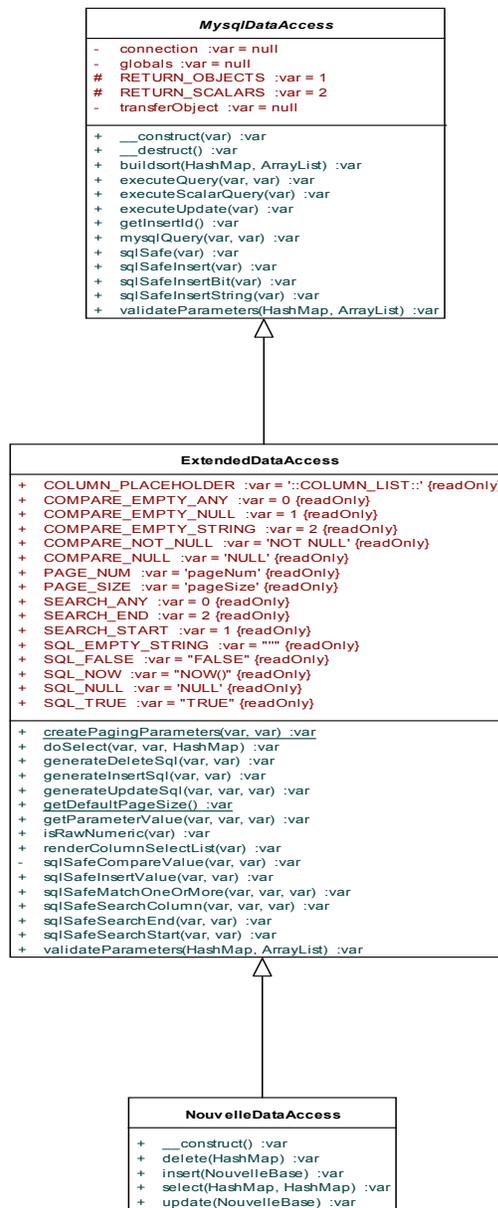


Figure 7 : Modèle représentant les classes d'accès à la base de données

La classe « **NouvelleDataAccess** » hérite de la classe « **ExtendedDataAccess** » représente la classe d'accès aux données des nouvelles à la base de données. La classe « **ExtendedDataAccess** » est un sous-type de la classe « **MysqlDataAccess** », et est utilisée

pour gérer les résultats d'une requête dans la base de données, et fournit aussi des interfaces d'accès au tableau.

- **Diagrammes de séquence**

Pour ajouter, modifier, supprimer les nouvelles, il faut que l'administrateur se connecte dans le système, donc créons tout d'abord le diagramme de séquence login.

- **Diagramme de séquence « Login »**

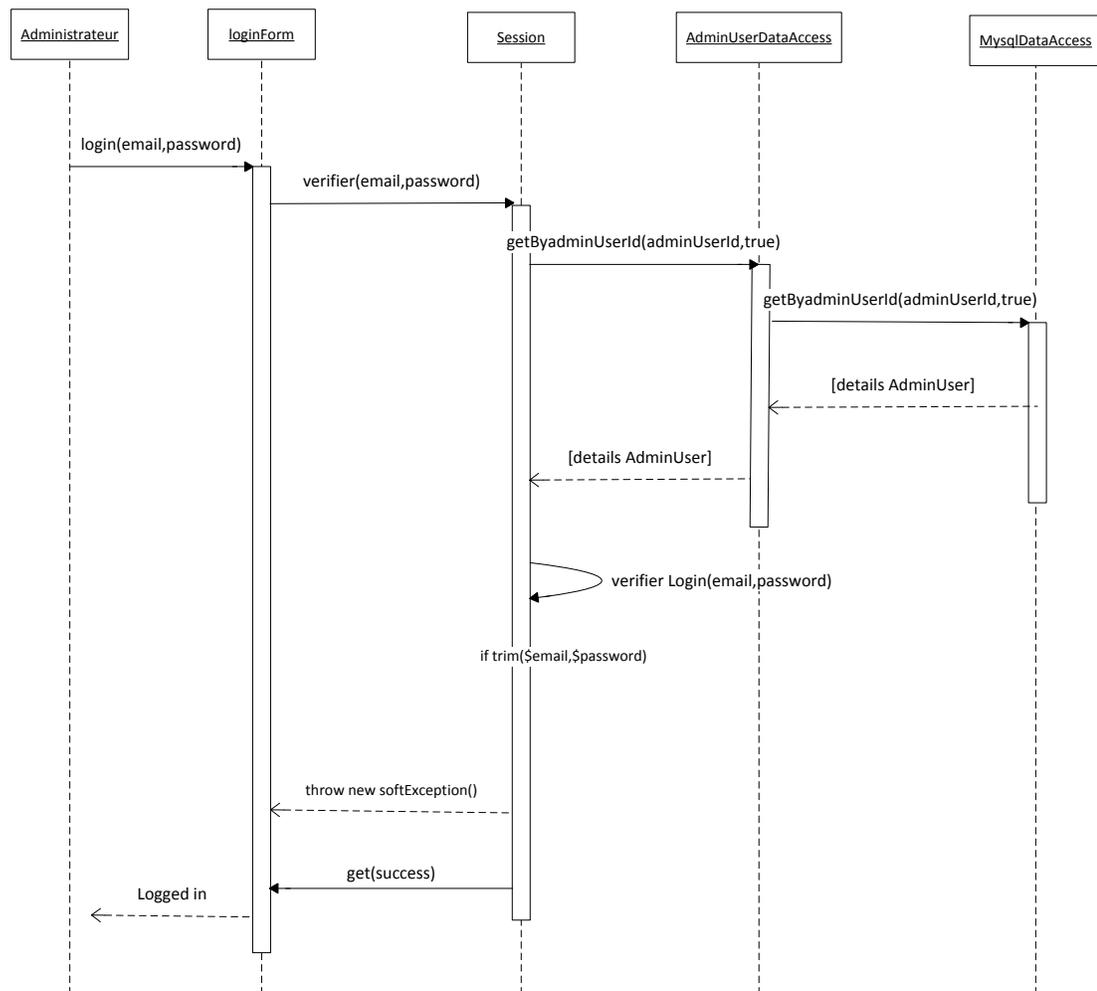


Figure 8 : Diagramme de séquence Login

Dans le diagramme de séquence ci-dessus l'administrateur entre son email et son mot de passe, puis clique sur le bouton de connexion de la page login. La classe Session reçoit les détails AdminUserId (firstName, lastName, password) de la classe AdminUserDataAccess et vérifie AdminUserId et le mot de passe. Si les informations sur l'administrateur ne sont pas valides, l'administrateur ne sera pas autorisé à se connecter et un message d'erreur s'affiche, sinon il sera connecté.

○ **Diagramme de séquence « Ajouter nouvelle »**

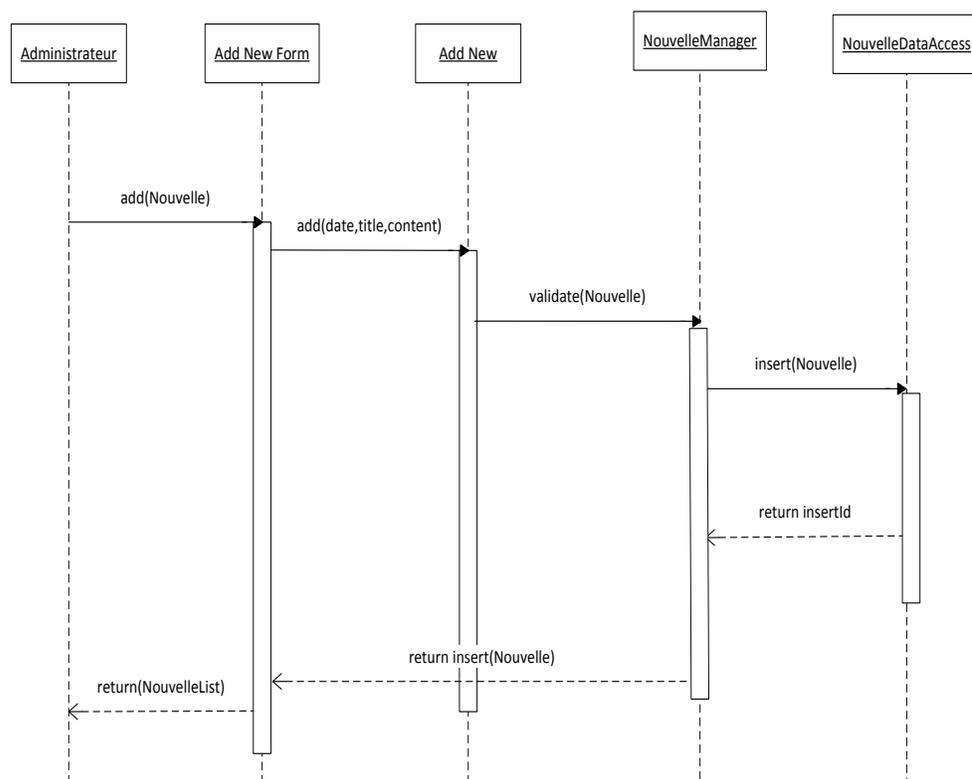


Figure 9 : Diagramme de séquence Ajouter nouvelle

Une fois authentifié, l'administrateur peut accéder à travers toutes les fonctions du système. Il peut ajouter des nouvelles ainsi que les détails dans la base de données. Une fois que la nouvelle est validée, elle sera insérée dans la base de données et la classe

NouvelleDataAccess retourne un Id à la nouvelle qui vient d'être enregistrée et le système affiche la liste des nouvelles avec une nouvelle de plus. Le diagramme de séquence ci-dessus (figure 9) illustre le processus. Voir l'annexe C pour plus d'information sur le diagramme de séquence « Modifier nouvelle ».

3.3 Modification au logiciel

Nous présentons dans cette section l'analyse d'impact des changements des exigences fonctionnelles, la conception et la réalisation des composants du nouveau système ainsi que les tests.

3.3.1 Analyse d'impact des changements

Cette section comporte la description des différentes demandes de changement qui ont eu lieu tout au long du projet. Pour chacune de ces modifications, nous allons effectuer une analyse d'impact sur le système existant en nous concentrant sur les parties les plus importantes de l'interface principale et les grandes fonctions de ce projet.

- **Remplacement de la photo de la page d'accueil du site S3M par les photos des clients du site web de livreaql**

La modification qui a été demandée est de remplacer la photo de la page d'accueil du site S3M par les photos des clients du site web de livreaql. Premièrement, nous avons d'abord identifié les attributs « alt="Intro image" », « width="721" height="228" », l'iframe « <iframe src="http://www.s3m.ca/extranet/index.php?action=securityPublicSubscriptionContent&lang=fr" frameborder="0" scrolling="no"></iframe> » et la barre de menu de navigation de l'interface principale tels que « Accueil, A propos de S3M, Services, produits, Contactez-nous » qui vont être impactées par le changement de l'image « img_mainbanner_fr.jpg » (voir figure 10, 11). Pour cela, nous avons utilisé le logiciel **Adobe Photoshop express** pour redimensionner la photo qui va remplacer celle qui était initiale afin de faciliter l'ajout. Par la suite, nous avons modifié les attributs width à 718 et height à 350, et exécuté l'application pour observer les résultats dans le but de nous assurer que la modification est réalisée correctement.

```

html body div div div div div img
173 <div id="containerTopNavigation" class="mainSize">
174 <div style="float: left; background: transparent url('images/interface/nav_bg_left.gif') no-x
175 <ul id="dropMenn" class="nav">
176
177 <li><a href="index_fr.html">Accueil</a></li>
178
179 <li><a href="#">À propos de S<span class="s3mSuperscript">3M</span></a>
180 <ul>
181 <li><a href="fr/aboutUs/overview.html">Survol</a></li>
182 <li><a href="fr/aboutUs/faq.html">FAQs</a></li>
183 <li><a href="fr/aboutUs/people.html">Gens</a></li>
184 <li><a href="fr/aboutUs/news/listing.php">Nouvelles</a></li>
185 <li><a href="fr/aboutUs/relatedLinks.html">Liens utiles</a></li>
186 </ul>
187 </li>
188
189 <li><a href="#">Services</a>
190 <ul>
191 <li><a href="fr/services/coaching.html">Coaching</a></li>
192 <li><a href="fr/services/training.html">Transfert de connaissances</a></li>
193 <li><a href="fr/services/sred.html">RS&amp;DE</a></li>
194 <li><a href="fr/services/certification.html">Certification</a></li>
195 </ul>
196 </li>
197
198 <li><a href="#">Produits</a>
199 <ul>
200 <li><a href="fr/products/s3massess.html">S<span class="s3mSuperscript">3M</span>Assess</a></li>
201 <li><a href="fr/products/s3mdss.html">S<span class="s3mSuperscript">3M</span>DSS</a></li>
202 <li><a href="fr/products/qassess.html">Outil d'évaluation rapide S<span class="s3mSuperscript">3M</span></a></li>
203 </ul>
204 </li>
205
206 <li><a href="fr/contactUs.html">Contactez-nous</a></li>

```

Figure 10 : Impact de changement de la photo de la page d'accueil du site S3m

```

html body div div div div div div div iframe
257 <script>new pauseScroller(pauseContent, "newsScroller", "someclass", 3000)</script>
258
259 <div style="width: 369px; height: 191px; class="floatRight" style="padding-right: 2px; usemap="#Map" alt="Disponible maintenant - Amiesecur;liorer la maintenance du
260
261 <div style="width: 369px; height: 191px; class="floatRight" style="padding-right: 2px; usemap="#Map" alt="Disponible maintenant - Amiesecur;liorer la maintenance du
262 <div style="width: 369px; height: 191px; class="floatRight" style="padding-right: 2px; usemap="#Map" alt="Disponible maintenant - Amiesecur;liorer la maintenance du
263 <div style="width: 369px; height: 191px; class="floatRight" style="padding-right: 2px; usemap="#Map" alt="Disponible maintenant - Amiesecur;liorer la maintenance du
264 38,62,306,0" href="http://www.youtube.com/watch?v=x0TdjIV2N1w" target="_blank" alt="En anglais" />
265
266
267
268
269
270 <div id="containerMainRight">
271 <iframe src="http://www.s3m.ca/extranet/index.php?action=security.showPublicSubscriptionContent&lang=fr" frameborder="0" scrolling="no"></iframe>
272 </div>
273
274 </div>

```

Figure 11 : Impact de changement de la photo de la page d'accueil du site S3m

- **Retrait de la fonction du wiki et agrandissement du blog et des labs internationaux**

La prochaine modification qui a été demandée est le retrait de la fonction du wiki et l'agrandissement du blog et des labs internationaux. Les endroits qui vont être impactés par les changements des images « `img_bg_blog.gif` » et « `img_bg_labs.gif` » sont les propriétés « `background : repeat-x bottom left` », « `margin-right : 0` » et les paragraphes contenant les textes (voir figure 12).

Afin de réaliser ce changement, nous avons dû modifier les attributs de hauteur et de largeur : « `height à 117px, et 40px` », « `width à 300px` » pour l'image « `img_bg_blog.gif` » et aussi les attributs « `height à 67px, width à 115px` » pour l'image « `img_bg_labs.gif` » dans l'objectif d'avoir la dimension appropriée. Pour maîtriser et maintenir l'effet de l'arrière-plan de l'image à l'état initial nous avons ajouté les propriétés « `background-size:contain` » et l'attribut « `min-width à 357px` ». Finalement, nous avons mis en œuvre l'application pour visualiser et analyser la soumission des formes des images pour pouvoir certifier que le changement correspond aux résultats attendus.

```

html
103 
105 <ul>
106 <li style="background: #FFFFFF url('images/interface/img_bg_blog.gif') repeat-x bottom left;">
107 <div>
108 <div style="float: right; width: 1px; height: 7.3em; padding: 0;"></div>
109 <div style="float: right; width: 125px; height: 40px; padding: 0; clear: right;"></div>
110 <h3><a href="http://drapril.blogspot.com/">Blog</a></h3>
111 <p>Un moyen rapide de s'inscrire et de s'inscrire et de s'inscrire
112 avec les auteurs.</p>
113 </div>
114 </li>
115 <li style="background: #FFFFFF url('images/interface/img_bg_labs.gif') repeat-x bottom left;">
116 <div>
117 <div style="float: right; width: 1px; height: 4.2em; padding: 0;"></div>
118 <div style="float: right; width: 115px; height: 70px; padding: 0; clear: right;"></div>
119 <h3><a href="fr/aboutUs/labs_map.html">Labs internationaux</a></h3>
120 <p>Les laboratoires associatifs au S<sup>3M</sup> travaillent
121 conjointement avec les Universités et les industries
122 pour faire avancer le domaine.</p>
123 </div>
124 </li>
125 <li style="background: #FFFFFF url('images/interface/img_bg_wiki.gif') repeat-x bottom left; margin-right:
126 <div>
127 <div style="float: right; width: 1px; height: 3.6em; padding: 0;"></div>
128 <div style="float: right; width: 80px; height: 70px; padding: 0; clear: right;"></div>
129 <h3><a href="dokuwiki/index.php">Wiki</a></h3>
130 <p>Une base de connaissance croissante de s'inscrire et de s'inscrire
131 pour les membres du S<sup>3M</sup>.</p>
132 </div>

```

Figure 12 : Impact de changement de retrait du wiki et l'agrandissement du blog et des labs

- **Ajout de la fonction « Nouvelles »**

La demande de changement qui a été demandée ici est d'offrir la possibilité d'ajouter des nouvelles à partir du site d'administration. En effet, nous avons étudié la possibilité de ne rien modifier dans les classes, les modules, les fonctions du système existant afin de respecter le « principe ouvert/fermé ». Pour ce faire, nous avons relevé les méthodes qui seront touchées par ce changement : « `handleRequest()`, `doAction()`, `includeTemplate()`, `ObjectFactory ::import()` » et les fonctions « `require()` » et « `include()` ».

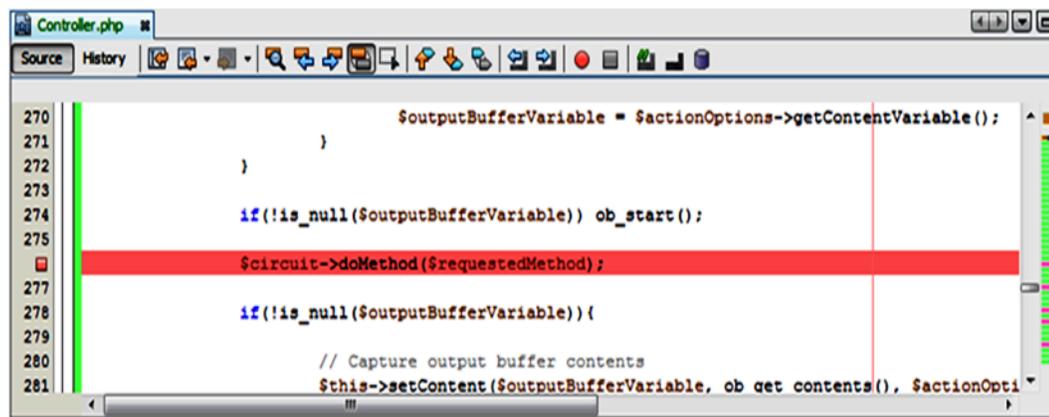
Pour effectuer cette modification, nous avons réutilisé les codes sources du package `adminUsers` afin de garder la cohérence avec les autres parties du système comme les interfaces externes en usage y compris les interfaces utilisateurs. Ensuite, nous avons créé la classe « `NouvelleDataAccess` » dans laquelle on a défini tous les paramètres d'insertion, de modification et suppression des nouvelles.

De plus, nous avons créé la classe « `NouvelleManager` » qui est la classe centrale dont sa responsabilité est de sauvegarder et de restaurer les données des nouvelles. Dans cette même classe, nous avons créé la fonction « `getByNouvelleId ($NouvelleId)` » qui permet d'obtenir des informations sur les nouvelles selon leurs identifiants, la fonction « `getAll(HashMap $sort)` » qui donne la possibilité d'avoir la liste de toutes les nouvelles qui ont été enregistrées et la fonction `add(NouvelleBase $Nouvelle)` qui permet d'ajouter une nouvelle dans la base de données. Une fois que la classe « `NouvelleManager` » est déclarée, nous avons créé une instance de cette classe qui est « `$NouvelleManager = new NouvelleManager ()` » pour pouvoir exploiter l'objet « `$NouvelleManager` », et nous avons fait appel à l'opération « `return $NouvelleManager->getByNouvelleId ($NouvelleId,true)` » pour pouvoir retourner la nouvelle avec son id en question. Il nous a fallu faire appel à l'opération « `controller ->addCircuit (new Circuit ('Nouvelles'))` » qui permet d'ajouter les nouvelles dans un circuit valide (répertoire). Enfin, nous avons créé la table « `Nouvelles` » qui permet de stocker toutes les données concernant les nouvelles (voir l'annexe D pour le script de la table « `Nouvelles` »).

Pour garantir que les modifications réalisées n'affectent pas d'autres parties du système, nous avons exécuté l'application, après nous avons procédé par une vérification de

la fonctionnalité « Nouvelles » afin de nous assurer que les changements sont effectués correctement et correspondent aux résultats recherchés. Les figures ci-dessous illustrent en rouge les impacts de changement pour la fonction « Nouvelles ».

Cette figure montre seulement la méthode « **doMethod()** » qui sera affectée par l'ajout de la fonction « Nouvelles ».



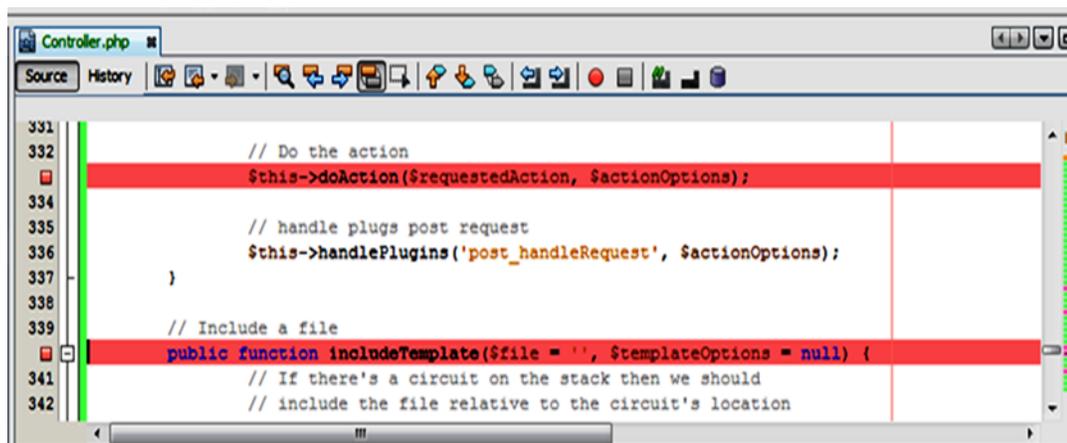
```

270         $outputBufferVariable = $actionOptions->getContentVariable();
271     }
272 }
273
274     if(!is_null($outputBufferVariable)) ob_start();
275     $circuit->doMethod($requestedMethod);
276
277     if(!is_null($outputBufferVariable)){
278         // Capture output buffer contents
279         $this->setContent($outputBufferVariable, ob_get_contents(), $actionOpti
280
281

```

Figure 13 : Impact d'ajout de la fonction « Nouvelles »

Les figures 14 et 15 affichent la méthode « **doAction()** », les fonctions « **includeTemplate()** » et « **include()** » qui seront touchées par l'ajout de la fonction « Nouvelles ».

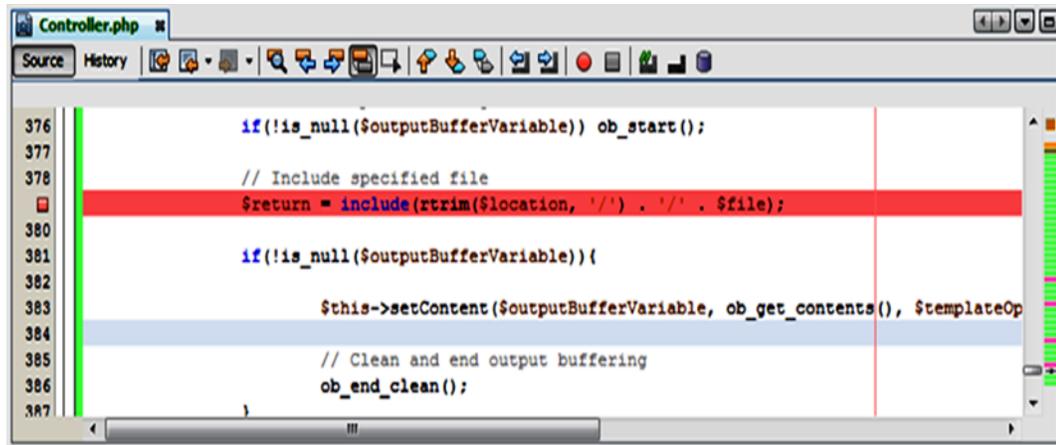


```

331     // Do the action
332     $this->doAction($requestedAction, $actionOptions);
333
334     // handle plugs post request
335     $this->handlePlugins('post_handleRequest', $actionOptions);
336 }
337
338 // Include a file
339 public function includeTemplate($file = '', $templateOptions = null) {
340     // If there's a circuit on the stack then we should
341     // include the file relative to the circuit's location

```

Figure 14 : Impact d'ajout de la fonction « Nouvelles »



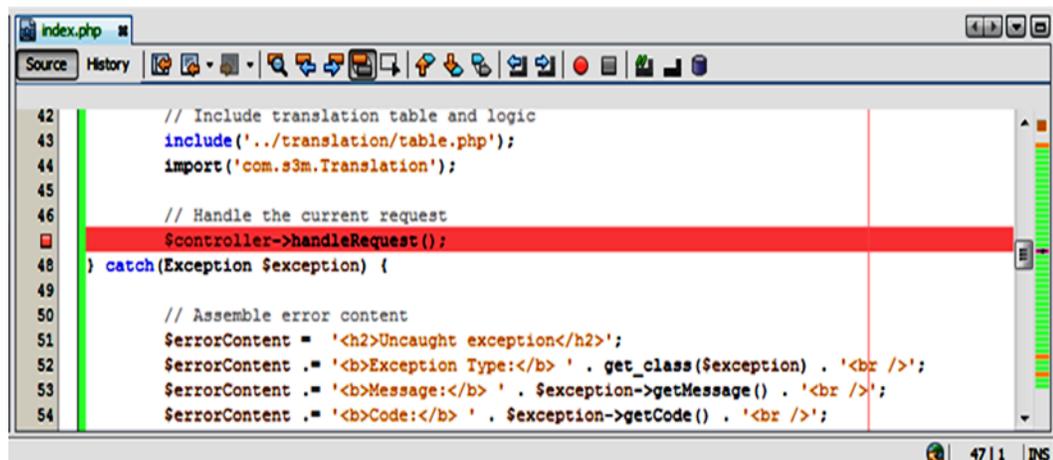
```

376         if(!is_null($outputBufferVariable)) ob_start();
377
378         // Include specified file
379         $return = include(rtrim($location, '/') . '/' . $file);
380
381         if(!is_null($outputBufferVariable)){
382
383             $this->setContent($outputBufferVariable, ob_get_contents(), $templateOp
384
385             // Clean and end output buffering
386             ob_end_clean();
387

```

Figure 15 : Impact d'ajout de la fonction « Nouvelles »

Les figures 16 et 17 montrent la méthode « **handleRequest()** » et la fonction « **require()** » qui seront affectées suite à l'ajout de la fonction « Nouvelles » .

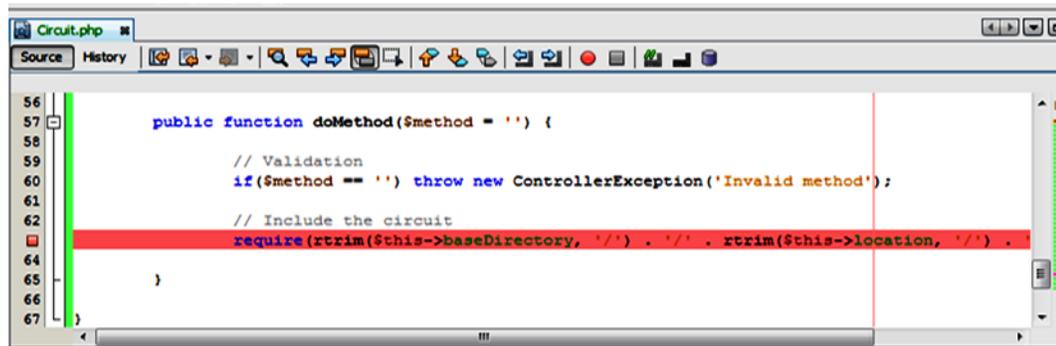


```

42         // Include translation table and logic
43         include('../translation/table.php');
44         import('com.s3m.Translation');
45
46         // Handle the current request
47         $controller->handleRequest();
48     } catch(Exception $exception) {
49
50         // Assemble error content
51         $errorContent = '<h2>Uncaught exception</h2>';
52         $errorContent .= '<b>Exception Type:</b> ' . get_class($exception) . '<br />';
53         $errorContent .= '<b>Message:</b> ' . $exception->getMessage() . '<br />';
54         $errorContent .= '<b>Code:</b> ' . $exception->getCode() . '<br />';

```

Figure 16 : Impact d'ajout de la fonction « Nouvelles »



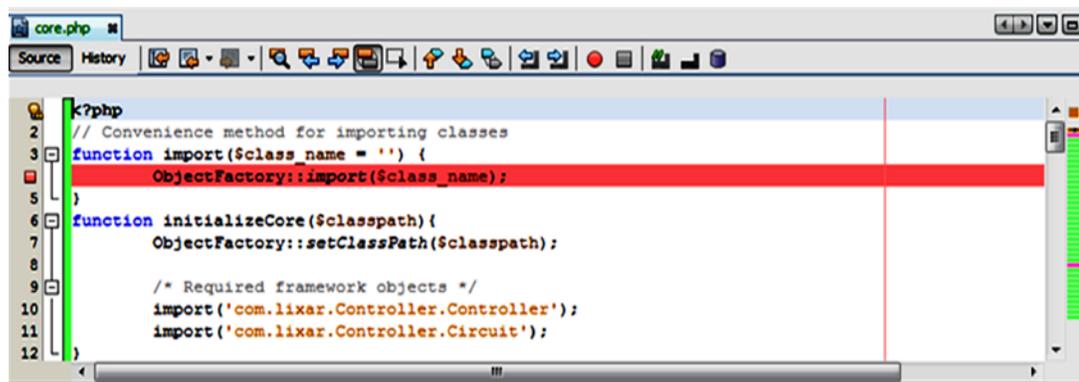
```

56
57
58     public function doMethod($method = '') {
59
60         // Validation
61         if($method == '') throw new ControllerException('Invalid method');
62         // Include the circuit
63         require(rtrim($this->baseDirectory, '/') . '/' . rtrim($this->location, '/') .
64
65     }
66
67 }

```

Figure 17 : Impact d'ajout de la fonction « Nouvelles »

Cette figure affiche la méthode « **ObjectFactory::import()** » de la fonction import qui va avoir de l'impact par l'insertion de la fonction « Nouvelles ».



```

core.php
Source History
1 k7php
2 // Convenience method for importing classes
3 function import($class_name = '') {
4     ObjectFactory::import($class_name);
5 }
6 function initializeCore($classpath){
7     ObjectFactory::setClassPath($classpath);
8
9     /* Required framework objects */
10    import('com.lixar.Controller.Controller');
11    import('com.lixar.Controller.Circuit');
12 }

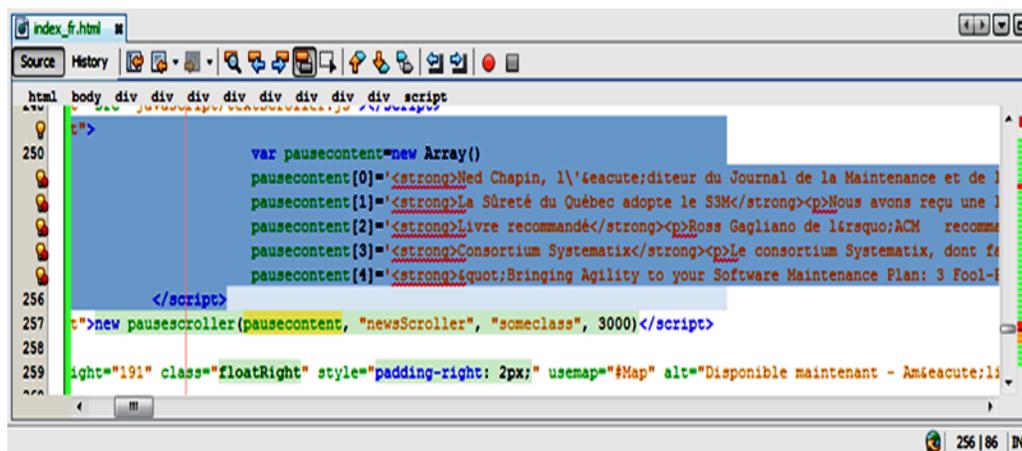
```

Figure 18 : Impact d'ajout de la fonction « Nouvelles »

- **Enlèvement des trois livres de maintenance et conservation des deux livres d'assurance qualité**

La demande de modification du client était d'enlever les trois livres de maintenance et de conserver les livres d'assurance qualité logicielle. Pour effectuer ce changement, nous avons trouvé le tableau « `pausecontent=new array ()` » qui sera atteint par ce changement. Pour implémenter ce changement, nous avons utilisé le logiciel *Adobe Photoshop express* pour nous faciliter à modifier l'image « `bookad_video_fr.jpg` » afin de conserver une similarité par rapport à l'image initiale. La figure suivante montre les impacts de changement

pour l'enlèvement des trois livres de maintenance et la conservation des deux livres d'assurance qualité logicielle.



```

html body div div div div div div div div div script
250
    var pausecontent=new Array()
    pausecontent[0]='<strong>Med Chapin, l'&eacute;diteur du Journal de la Maintenance et de l'
    pausecontent[1]='<strong>La S&uacute;ret&eacute; du Qu&eacute;bec adopte le S3M</strong><p>Nous avons re&uacute; une l
    pausecontent[2]='<strong>Livre recommand&eacute;</strong><p>Ross Gagliano de l'rsquo;ACM recomm
    pausecontent[3]='<strong>Consortium Systematix</strong><p>Le consortium Systematix, dont fe
    pausecontent[4]='<strong>quot;Bringing Agility to your Software Maintenance Plan: 3 Fool-f
256
    </script>
257
    =>new pausescroller(pausecontent, "newsScroller", "someclass", 3000)</script>
258
259
    ight="191" class="floatRight" style="padding-right: 2px;" usemap="#Map" alt="Disponible maintenant - Am&eacute;li

```

Figure 19 : Impact de changement de l'enlèvement des trois livres de maintenance

- **Modification de la fonctionnalité « A propos du site S3M en A propos de livreaql » ainsi que ces sous fonctionnalités**

La modification qui a été demandée est de modifier la fonctionnalité « A propos du site S3M en A propos de LivreAQL » ainsi que ces sous fonctionnalités. Par conséquent, il n'y avait aucun endroit qui sera affecté par cette modification. Nous avons employé la même stratégie qui était appliquée auparavant dans l'application web pour pouvoir conserver la même structure des pages HTML. Pour les sous fonctionnalités de « A propos de S3M », nous avons seulement trouvé les attributs suivants : « Alt = "Prof. Dr.-Ing. Alain April" », « alt="Prof. Dr.-Ing. Alain Abran" », « width = "103" », « height = 121 » et les paragraphes contenant les textes dans la fonctionnalité « Gens » qui vont être touchés suite à la modification des photos « img_drapril.jpg et img_drabran.jpg » (voir figure 20). Afin d'effectuer ce changement, nous avons mis en pratique le même principe que nous avons utilisé pour le remplacement de la photo de la page d'accueil du site S3M par les photos des clients du site de livreaql (voir même section). Nous avons dû modifier les attributs « width à 126 et height à 99 » pour maintenir la structure des pages à l'état qu'il était au début.

```

html body div img
112
113 0 20px; text-align: justify;">
114 <div class="subContentContainer" -->
115 <div class="bio">
116 <div class="bioImage">
117 <div class="bioContent">
118 <h2>Prof. Dr.-Ing. Alain April</h2>
119 <p>Prof. Dr.-Ing. Alain April est professeur de géoacoute;nie logiciel &grave; l'école de technolog
120
121 </div>
122 <div class="bio">
123 <div class="bioImage">
124 <div class="bioContent">
125 <h2>Prof. Dr.-Ing. Alain Abran</h2>
126 <p>Prof. Dr.-Ing. Alain Abran est professeur et directeur du laboratoire de géoacoute;nie logiciel &grave; l'école de technolog
127
128 </div>

```

Figure 20 : Impact de changement de la sous fonction gens

- **Modification des services et des produits et de leurs fonctionnalités**

Pour les services, les produits et leurs sous fonctionnalités, il n'y a aucun endroit qui sera affecté suite aux demandes de changements. Le tableau suivant présente l'estimation d'efforts pour le nombre d'heures de travail investi dans la réalisation du projet.

Tableau 4 : Estimation d'efforts pour le nombre d'heures de travail

Tâches	Effort estimé(en heures)
Préparation (installation et configuration d'environnement, etc.)	6 heures
Compréhension du logiciel	223 heures
Recherche documentaire	80 heures
Modification du logo du site S3M en logo de livreaql.	6 heures
Remplacement de la photo de la page d'accueil du site S3M par les photos des clients du site de livreaql	2 heures
Retrait de la fonction du wiki et l'agrandissement du blog et des labs internationaux	8 heures
Ajout de la fonction « Nouvelles »	35 heures
Enlèvement des trois livres de maintenance et conservation des deux livres d'assurance qualité logicielle.	6 heures
Modification de la fonctionnalité « A propos du site S3M en A propos de livreaql » ainsi que ces sous fonctionnalités.	10 heures
Modification des services, les produits et leurs sous fonctionnalités	10 heures
Documentation du logiciel (diagramme de classes, diagramme de séquences)	15 heures
Conception (architecture)	8 heures
Réalisation	3 heures
Tests	8 heures
Révision-correction du rapport	80 heures
Total	500 heures

3.3.2 Conception

Pour la construction de l'architecture du nouveau système, nous avons respecté les principes du Modèle -Vue -Contrôleur (MVC) auxquels nous avons additionné une couche d'accès aux données. L'objectif principal de l'architecture MVC est de séparer la logique métier et les données d'application à partir des données de présentation à l'utilisateur. Il exige la séparation entre les données, les traitements et la présentation pour donner trois parties fondamentales dans l'application finale : le modèle, la vue et le contrôleur (Marston, 2004). La figure suivante montre les quatre packages composant l'architecture de l'application web dont les classes vues sont regroupées dans le package « view », les classes applicatives dans « controller », les classes du domaine de l'application sont regroupées dans « model » et celles d'accès à la base de données dans « dataAccess ». Le modèle est chargé de gérer les données, il interagit avec la couche DataAccess, stocke et récupère les entités utilisées par l'application à partir de la base de données. La vue est chargée d'afficher les données fournies par le modèle dans un format spécifique. Le contrôleur est responsable d'administrer les couches de la vue et du modèle pour qu'il puisse fonctionner ensemble.

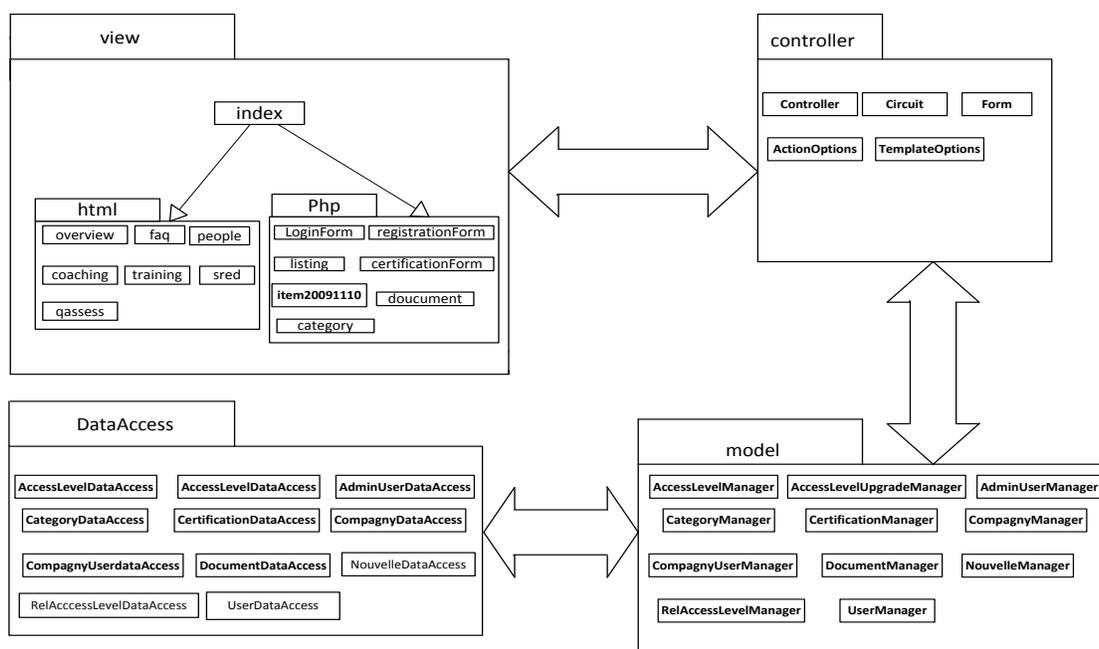


Figure 21 : Architecture de l'application web

Pour chacun des packages, ce modèle architectural montre les principales pages et classes participant dans la construction de l'application web du nouveau système. Dans le tableau suivant nous allons seulement présenter le rôle et la description pour chacune des pages et classes qui sont liées à la fonction « Nouvelles »: index, people, listing, item20091110, NouvelleManager, NouvelleDataAccess, Controller, et Circuit.

Tableau 5 : Description des classes et pages de l'architecture de l'application web

Package	Classe/Page	Description/Rôle
view	index	Cette page représente l'interface qui contient les éléments de la page d'accueil et des menus. C'est la page mère de l'application web. Elle permet d'afficher la liste des nouvelles défilantes ainsi que les livres d'assurance qualité qui sont disponibles sur le site (voir figure 22).
	people	Cette page permet d'afficher les informations sur les gens qui sont des spécialistes dans le domaine d'assurance qualité logicielle.
	listing	Cette page est responsable d'afficher la liste des nouvelles provenant de la base de données.
	Item20091011	Cette page permet d'afficher, de récupérer et de consulter uniquement la nouvelle qui correspond avec l'Id en question.
model	NouvelleManager	Cette classe est le point d'accès central pour la gestion de la persistance des nouvelles.
dataAccess	NouvelleDataAccess	Cette classe gère l'accès aux données des nouvelles à la base de données. C'est dans cette classe que sont définis tous les paramètres d'insertion, de modification et de suppression de nouvelle.
controllor	Controller	Cette classe permet de localiser les circuits, les templates, et de réutiliser une forme. Elle étend les fonctionnalités exitAction à une classe (absolu / relatif, etc), donne la possibilité pour les instances sous différents formats et effectue les potentiels phases plug-in : -pre_doAction -post_doAction -pre_handleRequest

Package	Classe/Page	Description/Rôle
		-post_handleRequest.
	Circuit	Cette classe permet de vérifier qu'un circuit a un nom et un emplacement (répertoire).

3.3.3 Réalisation

Nous présentons dans cette section l'interface principale des utilisateurs du nouveau système dans laquelle nous montrons la fonctionnalité « A propos AQL » incluant une de ses sous fonctionnalités : « Gens » et les interfaces graphiques des deux sous fonctions de la nouvelle fonctionnalité du site d'administration : Add et Edit New.

- **Interface principale des utilisateurs du nouveau système**



Figure 22 : Interface principale des utilisateurs du nouveau système

L'interface principale des utilisateurs affiche les exigences de fonctionnalités pour les utilisateurs qui sont les suivants : Accueil, A propos AQL, Services, Produits, Contactez-nous. « Gens » est une sous fonction de la fonctionnalité « A propos AQL ». Pour accéder à cette fonction il faut sélectionner le menu « A propos AQL » et cliquer sur « Gens » pour afficher la page ci-dessous.

Figure 23 : Interface de la sous-fonctionnalité « Gens »

- **Add New**

Cette fonctionnalité permet d'ajouter une nouvelle. Pour ajouter une nouvelle, l'administrateur doit se connecter dans le système. L'administrateur doit activer le menu « News » et cliquer sur le lien « Add a New » pour afficher l'interface permettant de saisir les données (voir figure 24). Il peut insérer les détails concernant les nouvelles telles que « Date, Title, Content » et cliquer sur le bouton « Save » pour enregistrer ces informations.

The screenshot shows the 'Add a New' form in the SQA application. The form includes the following elements:

- Date:** A text input field.
- Title:** A text input field.
- content:** A large text area with a vertical scrollbar.
- Buttons:** 'Save' and 'Cancel' buttons at the bottom.

Figure 24 : Écran d'ajout de nouvelle

- **Edit New**

Cette fonction permet de modifier une nouvelle. Pour modifier une nouvelle, l'administrateur doit activer le menu « News » et cliquer sur le lien « edit a New » pour afficher l'interface appropriée. Après avoir fini de modifier la nouvelle, cliquer sur le bouton « Save » pour mettre à jour les informations. Voir l'annexe E pour la suppression d'une nouvelle.

The screenshot displays the 'Edit a New' form in the SQA web application. The form is titled 'Edit a New' and contains the following fields:

- Date:** 2012-09-09
- Title:** Consortium Systematix
- content:** Le consortium Systematix, dont fait partie le SQA, qualifié par le Ministère du Développement économique, Innovation et Exportation du Québec dans le consortium Systematix.

At the bottom of the form, there are 'Save' and 'Cancel' buttons. The page header includes the SQA logo and navigation tabs for Content, Companies, News, Admin, and Logout. The user is logged in as lucfigaro@gmail.com on October 4, 2012.

Figure 25 : Écran de modification de nouvelle

3.3.4 Tests

Nous avons réalisé les tests pour tester les fonctions « `validate($Nouvelle)` » et « `getByNouvelleId()` » qui sont deux des fonctions qui composent la classe `NouvelleManager`. Pour ce faire, on a créé la classe test « `NouvelleManagerTest` » qui hérite de la classe mère `PHPUnit_Framework_TestCase` venant compléter notre classe de test. Premièrement, nous avons ajouté la méthode « `setUp ()` » pour initialiser la classe `NouvelleManager`. Ensuite, nous avons créé la fonction `testNouvelleDate()` pour permettre de tester si la nouvelle a une date, puis la fonction `testNouvelleTitle()` pour tester l'existence du titre de la nouvelle et la fonction `testNouvelleContent()` pour tester si la nouvelle a un contenu. Enfin, nous avons défini la méthode « `assertAttributeEmpty ()` » dans chacune de ces fonctions pour vérifier que les attributs existent. Pour la fonction `getByNouvelleId` nous avons défini la méthode « `assertEquals ()` » pour assurer que cette fonction retourne une nouvelle correspondant. L'annexe F contient le code source PHP de la classe `NouvelleManagerTest` qui permet de tester les fonctions `validate ($Nouvelle)` et `getByNouvelleId()` de la classe `NouvelleManager`.

CONCLUSION

Nous avons réalisé ce travail dans le but de modifier un site web existant pour le transformer en un autre site web : www.livreaql.org. Pour atteindre cet objectif, nous avons adopté une méthode de réingénierie visant à étudier sur la façon dont les programmeurs ont procédé pour comprendre un programme. Nous avons montré à travers les éléments théoriques abordés, comment le processus de compréhension des systèmes logiciels dans le programme en particulier, est au cœur de toutes les activités potentielles de maintenance et prend plus de la moitié du temps et d'effort consacré à effectuer des changements. Plusieurs stratégies et méthodes telles que le top-down, le bottom-up, l'opportuniste, la réingénierie, la restructuration, le réusinage (refactoring) pouvant aider à résoudre les problèmes que confrontent les programmeurs durant la compréhension d'un programme ont été présentés. Les défis et les problèmes de compréhension de programme ont été identifiés ainsi que les avantages de la réingénierie des applications web.

Nous avons proposé une approche rétro-ingénierie visant à permettre de comprendre le site web existant, afin de retirer le maximum d'information possible à partir du code source et l'exécution de l'application pour pouvoir réaliser les activités de modification. Pour effectuer les modifications, nous avons fait une analyse d'impact des demandes de changement. Ensuite, nous avons fait la conception et la réalisation du nouveau site web le www.livreaql.org qui va permettre de mettre à jour les nouvelles à partir du site d'administration et finalement nous avons procédé à la réalisation de tests.

Compte tenu de notre champ de recherche, pour comprendre un logiciel existant, il ne suffit pas de se rendre familier avec le code source de l'application, il faut avoir des artefacts comme les diagrammes de classes, diagrammes de séquence et les flux de données. Pour augmenter la flexibilité et acquérir l'objectif de réingénierie d'un logiciel existant, la liste suivante énumère les activités nécessaires qui doivent être prises en compte (Moore, 1993) :

- Identifier des informations de dépendance entre les composants de l'application ;
- Analyser les exigences du système à partir d'un point de vue fonctionnel et envisager une technologie appropriée pour améliorer les pratiques courantes ;
- Évaluer le système d'application avec l'intention de découvrir ce qui est digne d'être conservé pour une utilisation ultérieure ;
- Évaluer le code, la documentation, et les mesures appropriées pour déterminer l'état actuel du système.

4.1 Difficultés rencontrées

Nous avons rencontré certaines difficultés dans la mise en œuvre de l'application. Les premières difficultés concernent l'installation. Lors de l'installation, nous avons pensé que le package **AMP** qui comprend Apache, MySQL, PHP suffisait pour mettre en œuvre l'application. Cependant, nous avons rencontré des problèmes concernant la configuration du module PHP, pour cela nous avons dû éditer le fichier « php.ini » afin de faire fonctionner correctement XAMPP pour pouvoir lancer l'application. Une deuxième difficulté était l'inaccessibilité à la base de données MySQL. Il nous était difficile de trouver la solution, après avoir passé des heures à chercher, nous avons constaté que le problème venait de la non-définition des paramètres d'accès au serveur de base de données (l'adresse du serveur, le nom d'utilisateur, le mot de passe, le port, le nom de la base de données) dans le fichier « Globals.php » où nous avons spécifié notre environnement de projet.

4.2 Travaux futurs

L'application web ne regroupe que des sous fonctions concernant de services hors ligne. Voici les améliorations possibles à réaliser dans le site web existant :

- Ajout de la sous fonction *coaching en ligne*, fonction essentielle permettant de prendre des formations en ligne dans le domaine d'assurance qualité ;

- Ajout d'une sous fonction de *certification en ligne*, permettant de faire des certifications en ligne dans le domaine de l'assurance qualité.

4.3 Ce qui a été utile

La réalisation de ce projet nous a été très utile, premièrement au niveau technologique et ensuite sur le plan de l'apprentissage théorique de la réingénierie de logiciel.

Dans la phase pratique, nous avons utilisé des outils qui nous étaient jusqu'à ce moment inconnus pour mettre en œuvre l'application. Tel est le cas de NetBeans IDE qui est un environnement de développement intégré qui nous a donné la possibilité de différencier les différents composants de l'application web par son isolation de l'interface du code source et nous a permis d'acquérir de l'expérience dans la construction des interfaces graphiques. Au niveau de la manipulation des outils web, l'apprentissage a été aussi très bon.

Au niveau théorique, nous avons compris davantage que la réussite d'un projet de réingénierie logiciel passe d'abord par la compréhension de l'état du système actuel et souhaité, et la disponibilité de technologie de réingénierie en identifiant et en contrôlant les risques du système. De ce fait, la fouille que nous avons réalisée dans la littérature concernant le processus de compréhension de programme nous a permis de retenir les leçons suivantes :

- Avant de faire des changements, quels qu'ils soient, il est essentiel de bien comprendre le logiciel et de bien saisir l'impact qu'auront ces changements. En lisant la documentation du programme, en examinant le code source et les demandes de changement afin de déterminer l'ampleur des changements, c'est à dire les parties du système qui pourront être touchés par la demande de modification ;
- Exécuter le programme pour bien comprendre son comportement dynamique.

En outre, la réingénierie n'est pas un processus commun. Autrement dit, la réingénierie de chaque application envisagée est différente. Tout comme la culture varie d'une organisation à une autre. Mais avec l'adoption d'une bonne stratégie et d'une méthodologie adaptée, on peut

diminuer considérablement les risques et gérer plus efficacement la réingénierie d'une application web.

4.4 Lien entre le sujet du travail et le génie logiciel

Le génie logiciel occupe une place importante dans la réalisation de ce rapport. Le tableau 6 qui suit établit un lien entre le sujet du travail et le génie logiciel.

Tableau 6 : Tableau de lien entre le sujet du travail et le génie logiciel

Lien	Cours	Rôle
Analyse et spécifications / documentation de l'application web	Exigences et de spécifications de systèmes logiciels	Ce cours a aidé à la réalisation de l'analyse des besoins de l'application web existant afin de spécifier les exigences techniques, les comportements et les contraintes du système. De plus il a permis à la réalisation de la documentation du logiciel (diagramme de classes, diagramme de séquences).
Conception de l'application du nouveau système	Principes et applications de la conception de logiciels	Ce cours a contribué dans la réalisation de la conception afin de représenter clairement les éléments structuraux et comportementaux de l'architecture de l'application web du nouveau système.
Maintenance et réalisation	Réalisation et maintenance de logiciels	Ce cours a facilité à améliorer ou à transformer le logiciel existant tout en préservant son intégrité conceptuelle. En outre, il a donné la possibilité de réaliser la rétro-ingénierie de l'application web. Il a permis de réaliser plus facilement les activités de changements et aussi de récupérer et de réutiliser les ressources du logiciel existant(S3M), et d'établir une base pour l'évolution future du logiciel.
Tests	Vérification et assurance qualité de logiciels	Ce cours a participé à l'anticipation des défauts, de les détecter au bon moment et de s'assurer de l'adéquation des fonctionnalités de l'application web du nouveau système.

RÉFÉRENCES

- Akhlaq, U. 2010. «Impact of Software Comprehension in Software Maintenance and Evolution». *Blekinge Institute of Technology, Master Thesis, no: MSC-2010:15*.
- Bär, H., et S. Ducasse. 1999. *The FAMOOS Object Oriented Reengineering Handbook*: Forschungszentrum Informatik an der Univ. p.
- Bennett, K.H., et V.T. Rajlich. 2000. *Software maintenance and evolution: a roadmap: Proceedings of the Conference on the Future of Software Engineering*. ACM, 73-87 p.
- Canfora, G., A. Cimitile et P.B. Lucarelli. 2002. «Software maintenance». *Handbook of Software Engineering and Knowledge Engineering*, vol. 1, p. 91-120.
- Chikofsky, E.J., et J.H. Cross. 1990. «Reverse engineering and design recovery: A taxonomy». *Software, IEEE*, vol. 7, no 1, p. 13-17.
- Corbi, T.A. 1989. «Program understanding: Challenge for the 1990s». *IBM Systems Journal*, vol. 28, no 2, p. 294-306.
- Davis, K.H., et PH Alken. 2000. *Data reverse engineering: A historical survey: Reverse Engineering, 2000. Proceedings. Seventh Working Conference on*. IEEE, 70-78 p.
- Deshpande', Y., Murugesan, S., Ginige, A., Hansen, S., Schwabe, D., Gaedke, M. and White, B. 2002. "Web engineering", *Journal of Web Engineering*, Vol. 1 No. 1, pp. 61-73.
- Ducasse, S. 2003. «Reengineering object-oriented applications». *University of Bern, Institut für Informatik und Angewandte Mathematik, 2003, <http://www.iam.unibe.ch/~scg/Archive/Papers/Duca01cHab.pdf>*.
- Eichmann, D. 1999. *Evolving an engineered web: Proc. of the International Workshop on Web Site Evolution*. Citeseer.
- Feiler, P.H. 1993. «Reengineering: An engineering problem».

- Fowler, M., et K. Beck. 1999. *Refactoring: improving the design of existing code*: Addison-Wesley Professional p.
- Grubb, P., et A.A. Takang. 2003. *Software maintenance: concepts and practice*: World Scientific Publishing Company Incorporated p.
- Gui, G.Q. 2005. «Extending a Web authoring tool for Web site reverse engineering». University of Victoria.
- Jastroch, N., T. Marlowe, V. Kirova et M. Mohtashami. 2011. «Dams, Flows and Views: Cross-Aspect Use of Knowledge in Collaborative Software Development». *Journal of Systemics, Cybernetics and Informatics*, vol. 9, no 5, p. 36-40.
- Kienle, H.M., et H.A. Müller. 2001. *Towards a web site architecture framework for reengineering: First ASERC Workshop on Software Architecture*. Citeseer.
- Marston, T. (2004). The model-view-controller (mvc) design pattern for php
- Moore, T.K. (1993). Information Systems Criteria for Applying Software Reengineering: Guidelines for Identifying Information Systems for Software Reengineering, DTIC Document
- Olsem, M.R. 1998. «An incremental approach to software systems reengineering». *Journal of Software Maintenance*, vol. 10, no 3, p. 181-202.
- Pérez-Castillo, R., I.G.R. de Guzman, M. Piattini et C. Ebert. 2011. «Reengineering Technologies». *Software, IEEE*, vol. 28, no 6, p. 13-17.
- Rajlich, V. 1997. *Comprehension and Evolution of Legacy Software: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING*. IEEE COMPUTER SOCIETY, 669-670 p.
- Ricca, F., et P. Tonella. 2001. *Analysis and testing of web applications: Software Engineering, 2001. ICSE 2001. Proceedings of the 23rd International Conference on*. IEEE, 25-34 p.
- Ricca, F., P. Tonella et I.D. Baxter. 2002. «Web application transformations based on rewrite rules». *Information and Software technology*, vol. 44, no 13, p. 811-825.
- Rosenberg, L.H., et L.E. Hyatt. 1996. «Software re-engineering». *Tekninen raportti, Software Assurance Technology Center, lokakuu*.

- Scanniello, G. 2011. *Source code survival with the Kaplan Meier: Software Maintenance (ICSM), 2011 27th IEEE International Conference on*. IEEE, 524-527 p.
- Shull, F., J. Carver, G.H. Travassos, J.C. Maldonado, R. Conradi et V.R. Basili. 2003. «Replicated studies: building a body of knowledge about software reading techniques». *SERIES ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING*, vol. 12, p. 39-84.
- Storey, M.A.D., K. Wong et H.A. Muller. 1997. *How do program understanding tools affect how programmers understand programs?: Reverse Engineering, 1997. Proceedings of the Fourth Working Conference on*. IEEE, 12-21 p.
- Svahnberg, M. 2003. «Supporting Software Architecture Evolution». *Blekinge Institute of Technology Dissertation Series*, vol. 3.
- Tilley, S.R., et D. Smith. 1995. «Perspectives on legacy system reengineering».
- Von Mayrhauser, A., et A.M. Vans. 1995. «Program comprehension during software maintenance and evolution». *Computer*, vol. 28, no 8, p. 44-55.
- Yang, H., et M. Ward. 2003. *Successful evolution of software systems*: Artech House Publishers p.

ANNEXES

ANNEXE A

A.1 Prérequis logiciels pour l'installation de PHP

1. Prérequis logiciels

Pour installer PHP les prérequis suivants sont nécessaires :

- Une version récente de Java est exigée (au moins la version de Java 6).
- Le NetBeans IDE 7.1.1 pour PHP, téléchargeable à partir du site de NetBeans : <http://bits.netbeans.org/download/trunk/nightly/latest/>.
- Un serveur web Apache http server inclus dans XAMPP Windows dont les informations concernant son installation sont disponibles sur le site d'Apache : <http://httpd.apache.org/docs/2.2/install.html>
- Un moteur PHP inclus dans XAMPP Windows dont la version supportée est PHP5, téléchargeable à partir de <http://www.php.net/downloads.php>.
- Une version récente du debugger PHP est exigée (au moins la version XDebug 2.0). Un serveur de base de données MySQL server version 5.0 est recommandé.

2. Installation de Java 6

Pour réaliser l'installation de Java 6 sur Windows, on a suivi les étapes suivantes :

- Télécharger une version récente de Java à partir du site : http://java.com/en/download/help/windows_manual_download.xml.
- Exécuter l'installeur en cliquant sur Run.

- Sauvegarder le fichier à installer en cliquant sur save, ensuite choisir l'emplacement du dossier et enregistrer le fichier sur le système local.
- Faire une clique sur le bouton d'installeur pour accepter les termes de la licence et poursuivre l'installation.
- Presser sur le bouton «**Suivant**» pour continuer l'installation après avoir assuré que les programmes souhaités sont sélectionnés.
- Faire une clique sur fermer dans la boîte de dialogue précédente pour compléter processus d'installation de Java.

3. Installation de NetBeans IDE 7.1.1 sur Windows

Pour installer NetBeans sur Windows, on a réalisé les étapes suivantes :

- Télécharger une version récente de NetBeans IDE.
- Cliquer pour sur l'installeur pour l'exécuter.
- Cliquer sur suivant au page d'accueil de bienvenue wizard.
- Spécifier le répertoire d'installation au page d'installation de NetBeans IDE : C:\Program Files (x86)\NetBeans 7.1.1.
- Vérifier la liste des composants à installer sur la page de résumé et ensuite cliquer sur ok pour finir l'installation.

4. Installation et configuration du package XAMPP

Pour l'installation et la configuration du package XAMPP on a procédé comme suit :

- Télécharger le package d'installeur de XAMPP à partir du site:
<http://www.apachefriends.org/en/xampp-windows.html>.
- Exécuter le « fichier.exe ».
- Spécifier le dossier de destination du fichier pour installer XAMPP : C:\xampp.

- Sélectionner l'option pour installer le serveur Apache et le serveur de base de données de MySQL comme des services.
- Exécuter le fichier « XAMPP_HOME/xampp-control.exe » après la configuration en ouvrant le panneau de commande de XAMPP. A noter que les modules qui sont installés pendant les services sont déjà en fonctionnement.

5. Installation du debugger XDebug

- Télécharger une version récente de XDebug et compatible avec la version de PHP.
- Copier le « fichier .dll » dans le repertoire C:\xampp\php\ext.
- Ouvrir le fichier active de « php.ini » pour XAMPP
- Localiser dans le fichier active de « php.ini » les lignes suivantes et les effacer ou bien les marquées comme commentaire.

[Zend]

```
;zend_extension_ts="C:\ProgramFiles\xampp\php\zendOptimizer\lib\ZendExtensionManager.dll"
```

```
;zend_extension_manager.optimizer_ts="C:\ProgramFiles\xampp\lite\php\zendOptimizer\lib\Optimizer"
```

```
;zend_optimizer.enable_loader = 0
```

```
;zend_optimizer.optimization_level = 15
```

```
;zend_optimizer.licence_path =
```

- Ne pas commenter les lignes suivantes dans le « fichier php.ini » pour attacher le XDebug au moteur de PHP.

```
zend_extension = C:\xampp\php\ext\php_xdebug-2.1.4-5.3-vc9.dll
```

```
xdebug.remote_enable = on
```

```
xdebug.remote_handler = dbgp
```

```
xdebug.remote_host = localhost
```

```
xdebug.remote_port = 9000
```

- Sauvegarder le « php.ini » et exécuter le panneau de commande de XAMPP.
- Redémarrer le serveur Apache.

6. Configuration des propriétés de MySQL Server

Avant de pouvoir accéder au serveur de base de données MySQL dans NetBeans IDE, il est important de configurer les propriétés du serveur MySQL.

- Cliquez droit sur le nœud des bases de données dans la fenêtre « Services » et sélectionnez « Register MySQL server » pour ouvrir la boîte de dialogue de « MySQL Server Properties » suivante.

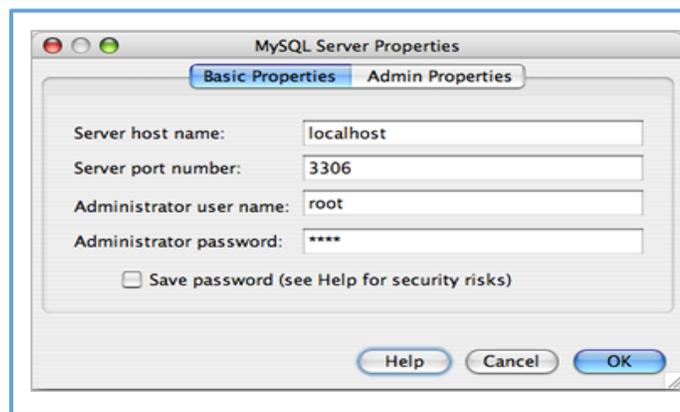


Figure 26 : Configuration des propriétés de MySQL Serveur

- Vérifiez que le nom d'hôte du serveur et le port sont corrects.
- Insérez le nom d'utilisateur de l'administrateur « root ».
- Saisissez le mot de passe de l'administrateur. La valeur par défaut est à blanc.
- Cliquez sur l'onglet « Admin Properties » dans le haut de la boîte de dialogue.
- Dans le « Path/URL » au champ outil d'administration, tapez ou naviguez jusqu'à l'emplacement de l'application Administration MySQL comme l'outil d'administration de « MySQL », « PhpMyAdmin », ou d'autres outils d'administration basés sur le Web.
- Dans le « Path » de la commande de démarrage, tapez ou naviguez jusqu'à l'emplacement de la commande de démarrage de MySQL. Pour trouver la

commande de démarrage, recherchez « mysqld » le dossier « bin » du répertoire d'installation de MySQL.

- Dans le « Path » pour arrêter le champ de commande, tapez ou naviguez jusqu'à l'emplacement de la commande d'arrêt MySQL. Il s'agit généralement du « Path » d'accès à la commande « mysqladmin » dans le dossier « bin » du répertoire d'installation de MySQL. Si la commande « mysqladmin » est dans le champ Arguments, saisissez « -u root stop » pour accorder des permissions « root » pour arrêter le serveur.
- Lorsqu'on a terminé, l'onglet « Admin Properties » ressemble à la figure suivante et cliquez sur « OK ».

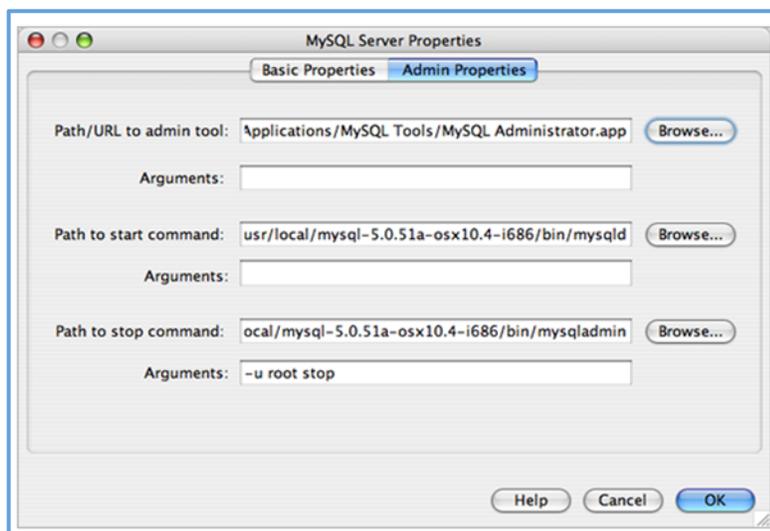


Figure 27: Admin propriétés

- Pour que l'application se connecte au serveur de base de données MySQL, Il est important de nous assurer que le serveur de base de données MySQL est exécuté sur la machine. Le nœud de « MySQL Server » dans la fonction indique si le serveur de base de données MySQL est connecté, comme illustré dans l'image suivante.

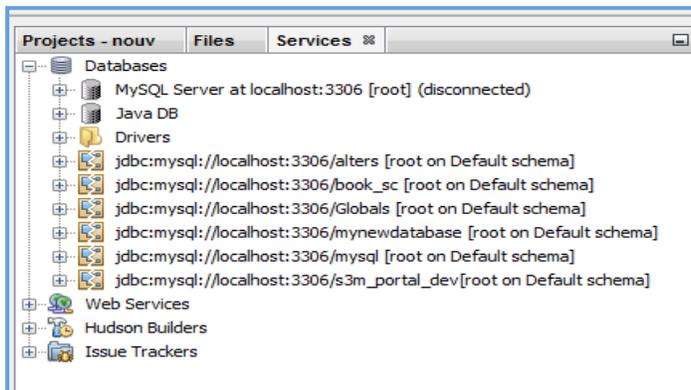


Figure 28 : Démarrage du serveur MySQL

- Après avoir vérifié que le serveur de base de données MySQL est exécuté sur la machine, Faire une clique droite sur le nœud de « MySQL Server » des bases de données dans la fenêtre « Services » et choisissez « Connect ». Etendre le nœud de « MySQL server » pour afficher toutes les bases de données de MySQL disponibles parmi lesquelles le «s3m_portal_dev » qui est la base de données du projet, montrez dans la figure suivante.

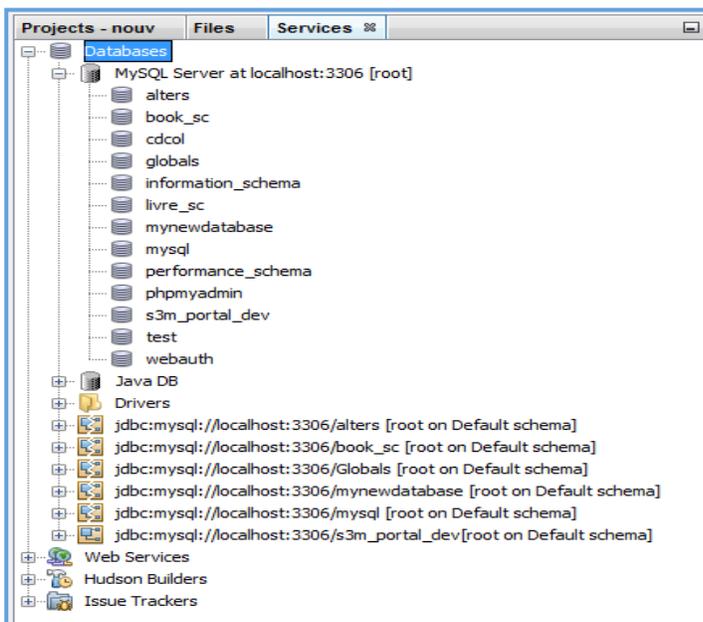


Figure 29 : Connexion de MySQL Serveur

ANNEXE B

B.1 Diagramme de classes de toute l'application

Dans cette annexe, nous présentons en grande partie le diagramme de classes de toute l'application.

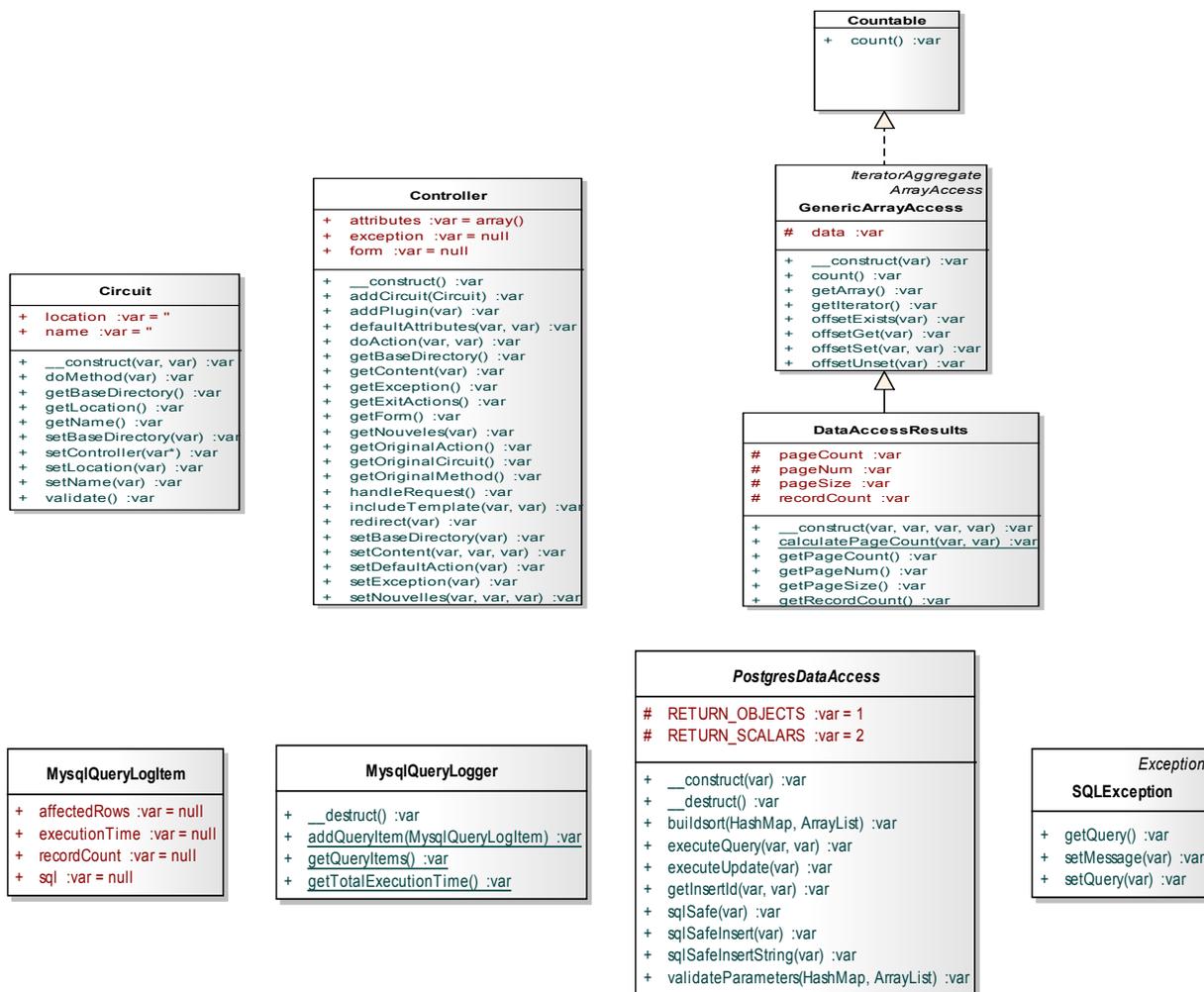


Figure 30: Diagramme de classes de l'application



Figure 31: Diagramme de classes de l'application



Figure 32: Diagramme de classes de l'application



Figure 33: Diagramme de classes de l'application

ANNEXE C

C.1 Diagramme de séquence « Modifier nouvelle »

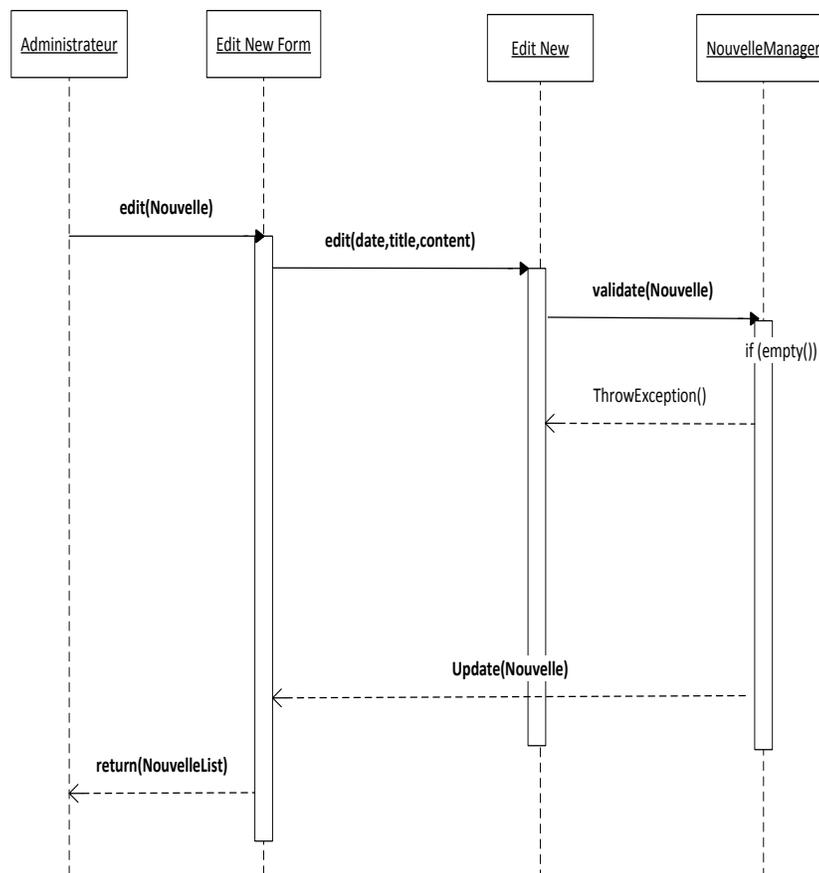


Figure 34: Diagramme de séquence de modification de nouvelle

Pour modifier une nouvelle l'administrateur sélectionne la nouvelle à modifier. La fenêtre contenant les détails de la nouvelle apparaît. Ensuite l'administrateur saisie les

informations correspondantes concernant la nouvelle à modifier. Des contrôles sur le domaine s'effectue c'est à dire les remplissages de tous les champs, s'il n y a pas de champs de vide (date, titre, contenu). Si tous les champs ne sont pas remplis le système envoie une exception demandant de saisir les informations concernant les champs vides. Si les tous les champs sont remplis la modification de la nouvelle sera validée, et les informations concernant la nouvelle seront mises à jour.

ANNEXE D

D.1 Script de la table « Nouvelles »

Nous présentons dans cette annexe le script de création de la table «Nouvelles» dans la base de données de s3m_portal_dev. Un script d'ajout de colonnes pour marquer comme supprimé au lieu de simplement supprimer l'enregistrement actuel de la table nouvelles dans la base de données est également ajouté dans cette annexe.

```
-- CREATION DE LA TABLE NOUVELLES

CREATE TABLE NOUVELLES IF NOT EXISTS (

ID INT AUTO_INCREMENT NOT NULL PRIMARY KEY,

DATE DATE NOT NULL,

TITLE VARCHAR (255) NOT NULL,

CONTENT MEDIUMTEXT NOT NULL,

CREATEDON DATETIME NOT NULL,

CREATEDBY INTEGER NOT NULL,

MODIFIEDON DATETIME NOT NULL,

MODIFIEDBY INTEGER NOT NULL

) ENGINE=INNODB;
```

```
-- AJOUT DE COLONNES POUR MARQUER COMME SUPPRIMÉ AU LIEU DE  
SUPPRIMER L'ENREGISTREMENT ACTUEL
```

```
ALTER TABLE `NOUVELLES` ADD `IS_DELETED` TINYINT (1) UNSIGNED NOT  
NULL DEFAULT '0';
```

ANNEXE E

E.1 Écran de suppression de nouvelle

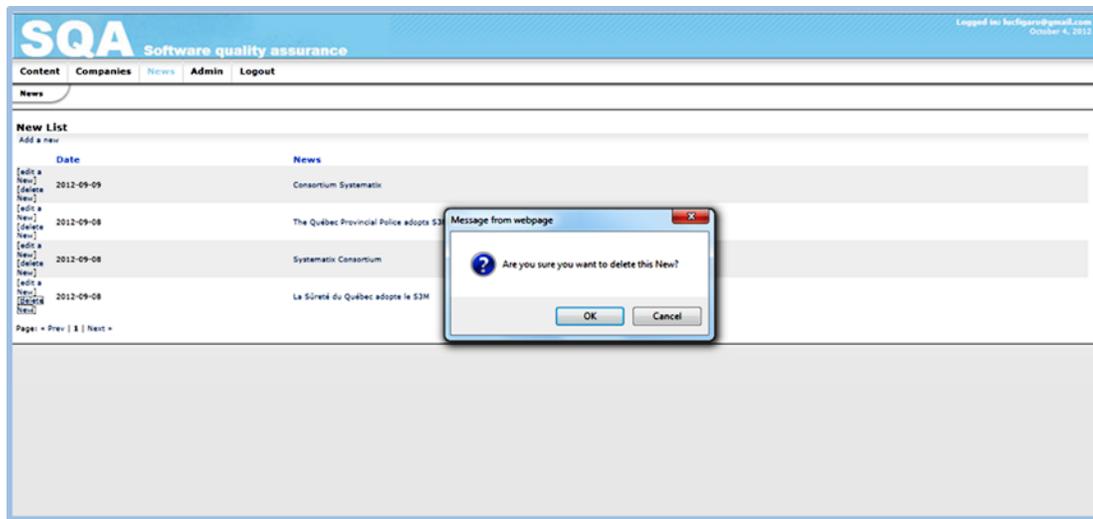


Figure 35 : Écran de suppression de nouvelle

Pour supprimer une nouvelle l'administrateur doit cliquer sur le lien « delete new » et un message apparait demandant de confirmer la suppression. Ensuite on doit cliquer sur le bouton « ok » pour confirmer ou sur le bouton « Cancel » pour annuler la suppression.

ANNEXE F

F.1 Code source pour tester la fonction valide (\$Nouvelle) et la fonction getByNouvelleId () de la classe NouvelleManager

```
<?
class NouvelleManagerTest extends PHPUnit_Framework_TestCase {
public function setUp() {
require_once 'NouvelleManager.php';
}
/*
* La nouvelle a une date
*/
public function testNouvelleDate() {
$NouvelleManager = new NouvelleManager('2012-10-10');

// vérifier que l'attribut existe
$this->assertAttributeEmpty('2012-10-10', 'date', $NouvelleManager);
}
/*
* La nouvelle a un titre
*/
public function testNouvelleTitle() {
$NouvelleManager = new NouvelleManager('Assurance qualite');

// vérifier que l'attribut existe
$this->assertAttributeEmpty('Assurance qualite','title', $NouvelleManager);
}
}
```

```
/*  
 * La nouvelle a un contenu  
 */  
public function testNouvelleContent () {  
    $NouvelleManager = new NouvelleManager('Assurance qualite logicielle');  
  
    // vérifier que l'attribut existe  
    $this->assertAttributeEmpty ('Assurance qualite logicielle','content', $NouvelleManager);  
}  
/*  
 * test pour obtenir une nouvelle par id  
 */  
public function testgetByNouvelleId () {  
    $this->assertEqual ($this->NouvelleManager->getByNouvelleId(1), 1);  
}  
  
}  
?>
```