



Le génie pour l'industrie

***MGL940 –Projet en génie logiciel***

***MTR871 – Lectures dirigées***

***Nom du professeur : Alain April***

**Outils de contrôle de la qualité logiciel (logiciel  
libre)**

**Ahmed Sedjai**

**Date: 2012-04-08**

## SUIVI DES CHANGEMENTS

**\*A – AJOUTÉ M – MODIFIÉ S – SUPPRIMÉ**

NUMÉRO DE VERSION	DATE AAAA-MM-JJ	NUMÉRO DE FIGURE, TABLE OU SECTION	A* M S	BRÈVE DESCRIPTION DU CHANGEMENT	NUMÉRO DE DEMANDE CHANGEMENT
1.0	2012-01-05	TOUT LE DOCUMENT	A	ÉBAUCHE INITIALE	
2.0	2012-01-11	SECTION 2	A	AJOUT DES SECTIONS	
3.0	2012-01-17	SECTION 3	A	AJOUT DES SECTIONS	
4.0	2012-01-21	SECTION 3	M	AJOUT DES SECTIONS	
5.0	2012-01-24	SECTION 3	M	AJOUT DES SECTIONS	
6.0	2012-01-28	SECTION 3	M	AJOUT DES SECTIONS	
7.0	2012-02-01	SECTION 4	A	AJOUT DES SECTIONS	
8.0	2012-02-12	SECTION 5	A	CORRECTIONS	
9.0	2012-02-27	SECTION 6	A	CORRECTIONS	
10.0	2012-03-09	SECTION 7	A	CORRECTIONS	
11.0	2012-04-08	TOUT LE DOCUMENT	M	RÉVISION ET CORRECTIONS	

## STATUT ET NIVEAU DE QUALITÉ DU DOCUMENT

**\*NI – NON INSPECTÉ I – INSPECTÉ E – ESTIMÉ C – CALCULÉ**

NUMÉRO DE VERSION	DATE AAAA-MM-JJ	STATUS *NI I	NOMBRE DE DÉFAUTS (NOMBRE /PAGE DE 300 MOTS OU NOMBRE / 300 LOC)	* E C
1.0	2012-01-05	NI	N/A	
2.0	2012-01-11	NI	N/A	
3.0	2012-01-17	NI	N/A	
4.0	2012-01-21	NI	N/A	
5.0	2012-01-24	NI	N/A	
6.0	2012-01-28	NI	N/A	
7.0	2012-02-01	NI	N/A	
8.0	2012-02-12	NI	N/A	
9.0	2012-02-27	NI	N/A	
10.0	2012-03-09	NI	N/A	
11.0	2012-04-08	I	2	E

## Table des matières

1.	Introduction .....	1
2.	Acronymes et définitions.....	1
3.	Avantages du libre pour les utilisateurs .....	1
4.	Contexte de projet .....	2
5.	Définition du champ de l'étude .....	2
6.	Travaux (articles et publications) existants et visant la même problématique de notre projet.....	2
6.1.	Quelques sources bibliographiques et travaux ayant traité partiellement la même problématique spécifiée dans mon projet.....	3
6.1.1.	1 <sup>ère</sup> tentative .....	3
6.1.2.	2 <sup>ème</sup> tentative .....	4
7.	Sélection et identification des logiciels libres les plus populaires utilisés aujourd'hui pour chaque activité du cycle de vie (ISO 12207) .....	5
7.1.	Méthode d'évaluation utilisée pour identifier quels sont les logiciels libres les plus populaires utilisés pour chaque activité de cycle de vie (ISO12207) .....	5
7.2.	Logiciels libres qui supportent des activités de génie logiciel .....	5
7.3.	Méthode utilisée pour la recherche des logiciels libres.....	5
7.4.	Logiciels libres et commerciales identifiés et suggérés lors d'un sondage réalisé au moyen d'un questionnaire électronique sur SurveyMonkey .....	6
7.4.1.	Questionnaire électronique sur SurveyMonkey .....	6
7.4.2.	Recherche de logiciels libres et commerciaux identifiés lors d'un sondage réalisé au moyen d'un questionnaire électronique sur SurveyMonkey .....	6
7.5.	Recherche de logiciels libres et commerciales utilisés par les étudiants et le personnel académique dans les cours de programme de premier cycle en Génie logiciel/TI et identifiés dans la figure de l'article "Integrating Tools and frameworks in undergraduate software engineering curriculum" présente en annexe 3. ....	9
7.5.1.	Sélection des logiciels libres (inventaire).....	9
7.6.	Recherche des logiciels libres de cycle de vie du répertoire "Sourceforge.net" .....	11
7.7.	Inventaire globale des projets de logiciels libres les plus populaires qui supportent des activités de génie logiciel .....	11
7.8.	Méthodologie de sélection à partir de l'inventaire globale .....	12
8.	Les critères dévaluation et de sélection qui vont nous permettre de choisir un logiciel libre d'une activité de génie logiciel plutôt qu'un autre .....	12
8.1.	Méthode d'évaluation utilisée pour repérage des logiciels libres.....	12
8.2.	Evaluation de l'utilisabilité de logiciels libres - Métriques d'utilisabilité .....	13

8.3.	La grille d'évaluation de Wheeler [3].....	13
8.4.	Le modèle de maturité Open Source OSMM de Capgimini [3].....	14
8.5.	Le modèle de maturité Open Source OSMM de Golden [3] .....	15
8.6.	Sélection des logiciels libres matures .....	15
8.7.	Sélection finale - Tests des logiciels libres matures .....	15
9.	Intégration des logiciels libres de cycle de vie dans une machine virtuelle unique afin de supporter un spécialiste afin de ne pas nécessiter trop de formation/difficultés d'apprentissage .....	16
9.1.1.	Défis.....	16
9.1.1.1.	Les défis techniques .....	16
9.1.1.2.	Le défi d'apprentissage.....	16
	Annexe.....	18

## 1. Introduction

Le concept du logiciel libre date du milieu des années 80, lorsque Richard Stallman, travaillant sur le système d'exploitation Unix, s'exaspère de la fermeture de plus en plus du code source du logiciel au profit d'une licence d'utilisation (méthode popularisée par Microsoft). Richard Stallman crée alors la Free Software Foundation et développe la licence GPL pour définir le logiciel libre. Il existe une communauté de logiciels libres qui distribue des produits de qualité dont le code source est publié et diffusé ouvertement. Les logiciels libres sont couverts par des licences publiques qui garantissent les droits à l'utilisateur d'exécuter, de modifier et de redistribuer le logiciel. [2]

Le logiciel libre publié et diffusé ouvertement doit respecter les 4 libertés [2]:

- La liberté d'exécuter le programme, pour tous les usages (liberté 0).
- La liberté d'étudier le fonctionnement du programme, et de l'adapter au besoin.
- La liberté de redistribuer des copies, donc d'aider son voisin, (liberté 2).
- La liberté d'améliorer le programme et de publier ses améliorations, pour en faire profiter la communauté Open source (liberté 3).

Historiquement, internet, en général a été construite sur la base des logiciels libres avec le serveur d'infrastructure Linux, le serveur Apache et le SGBDR MySQL. Les grands acteurs du monde des TI tels que IBM avec Redhat Linux, Novell avec Suse ont intégré les logiciels libres dans leurs plateformes de systèmes logiciels dans leur organisation. Yahoo et Google ont tous deux réussi dans le monde du Web parce qu'ils ont pu intégrer des logiciels libres dans leur architecture. [2]

## 2. Acronymes et définitions

**OSMM:** Open source Maturity model.

**ETS :** École de Technologie Supérieure.

**OOD:** OO Design.

**OOA:** OO analysis & design.

## 3. Avantages du libre pour les utilisateurs

Parmi les avantages du logiciel libre sur les logiciels propriétaires ou commerciales [2]:

- Absence de coût de licence.
  - ✓ coût de licence d'utilisation réduit;
  - ✓ coût de contrat d'entretien nul;
  - ✓ coût minimal des risques de passage à un logiciel alternatif.
- Accès au code source et droit de le modifier.
  - ✓ Permet de faire des adaptations;
  - ✓ Garantie de disponibilité du logiciel;
  - ✓ Contrôle de passage aux nouvelles versions;
- Liberté de choisir des fournisseurs.
  - ✓ Bénéficier de la concurrence et des avantages de la compétition;
  - ✓ Permet de changer de fournisseur en cas de problèmes;
  - ✓ Risque minimal en cas de changement de fournisseur ou en cas de passage à une nouvelle version;
- Code source accessible par tout le monde.

## 4. Contexte de projet

Notre projet vise à identifier des logiciels libres qui supportent des activités de génie logiciel. Par exemple lors de la définition d'exigences il serait intéressant d'utiliser un outil pour décrire les exigences des clients d'une manière graphique. Il s'agit d'établir des critères de comparaison, comparer les logiciels disponibles, en identifier un candidat et ensuite configurer une machine virtuelle avec le logiciel en question à des fins d'utilisation gratuite. On itère sur quelques logiciels clés de manière à aider les utilisateurs car ils auraient une machine virtuelle pré-configurée avec de bons outils gratuits.

## 5. Définition du champ de l'étude

Pour les besoins de notre projet, nous allons nous limiter aux outils de développement qui supportent les activités de cycle de vie de génie logiciel :

- Éditeurs, interpréteur, compilateur, et système de contrôle de version;
- Les logiciels de bureautique : traitement de texte, chiffrier, etc.
- Les outils d'analyse des besoins et d'élaboration des spécifications;
- Les outils de conception;
- Les outils de construction/ développement;
- Les outils de tests;
- Les outils de livraison;
- Les outils "Database & persistence";
- Les outils "Debugger/Monitor/Simulator";
- Les outils de l'environnement de développement;
- Les outils "Graphics API";
- Les outils "Library/API";
- Les outils "Modeling and design";
- Les outils "Project management";
- Les outils "QA/Maintenance";
- Les outils "Web development";

## 6. Travaux (articles et publications) existants et visant la même problématique de notre projet

### Recherche d'articles et publication

Pour rechercher les articles et publications existants et visant la même problématique, c'est-à-dire, portant sur les mêmes sujets qu'aborde notre projet, nous avons utilisé les sources de recherches bibliographiques suivantes :

#### Digital Library

- ACM digital Library;
- IEEE Xplore DIGITAL LIBRARY.

#### Mémoires et thèses électroniques

- ProQuest Dissertations and These;
- Espace ETS.

## **Google Scholar**

- Google Scholar.

### **6.1. Quelques sources bibliographiques et travaux ayant traité partiellement la même problématique spécifiée dans mon projet**

Il y a eu quelques tentatives qui traitent partiellement la problématique spécifiée dans notre projet. Parmi celles-ci, nous avons sélectionné deux :

#### **6.1.1.1<sup>ère</sup> tentative**

**Titre de l'étude :** Évaluation et expérimentation de logiciels libres pour la petite et moyenne entreprise.

Michèle Sauvé, Mario Boutin, Robert Gérin-Lajoie, Isabelle Therrien

**Source:** Google Scholar; 2005; <http://www.cirano.qc.ca/pdf/publication/2005RP-17.pdf>

#### **Résumé**

Cette étude vise à identifier en premier lieu les meilleurs logiciels d'affaires libres parmi un très grand nombre de logiciels d'affaires libres. Il s'agit en premier lieu de définir la liste des différents logiciels d'affaires. Les auteurs se sont inspirés des méthodologies de sélection de Capgemini et de Wheeler pour construire la liste des logiciels d'affaires libres les plus prometteurs.

Par la suite, parmi la liste, les auteurs ont identifié trois logiciels d'affaires diffusés sous licence libre pour une évaluation plus détaillée. Le but de l'évaluation détaillée est de pouvoir identifier des logiciels approchant la maturité. En finale, les logiciels identifiés approchant la maturité peuvent être proposés pour faire l'objet d'un projet pilote d'implantation en entreprise.

#### **Problématique traitée**

L'utilisation des logiciels d'affaires propriétaires en PME Canadienne entraîne souvent des coûts de licences d'utilisation et des contrats d'entretien importants, ce qui les rend inaccessibles aux PME. Vu le faible coût que peut engendrer le logiciel d'affaires libre aux PME, il semble logique de penser à acquérir le meilleur logiciel d'affaires libre qui remplit les besoins stratégiques et professionnels de l'entreprise. Pour ce faire, en premier lieu, il s'agit d'établir des critères de comparaison pour comparer les logiciels d'affaires disponibles, en identifier des candidats et en sélectionner le meilleur candidat parmi la liste des logiciels d'affaires libres.

#### **But de l'étude**

L'objectif de ce projet est d'identifier des candidats de logiciels libres d'affaires, en sélectionner le meilleur candidat qui remplit les besoins stratégiques et professionnels de l'entreprise parmi la liste des logiciels d'affaires libres. En finale, le logiciel d'affaires sélectionné doit procurer un avantage concurrentiel à la PME par rapport aux logiciels d'affaires propriétaires.

## Solution proposée

Pour trouver le logiciel d'affaire le plus adapté, il s'agit d'évaluer les produits logiciels d'affaire selon les besoins réels des utilisateurs. Pour ce faire, les auteurs ont utilisé une méthode d'évaluation pertinente comme la grille d'évaluation du modèle de maturité proposé par Capgemini nommé Open source Maturity model OSMM.

## Moyens utilisés

Pour atteindre ce but, les auteurs ont procédé globalement comme suit [3]:

- Appliquer le processus de selection proposé par Wheeler sous forme de grille de selection complète pour compléter notre courte liste de candidats. Ce processus comporte quatre étapes:
  - ✓ Identifier
  - ✓ Évaluer
  - ✓ Comparer
  - ✓ Analyser
- Définir une typologie de logiciels d'affaires;
- Développer une grille d'évaluation des logiciels d'affaire;
- Faire la selection de trois logiciels;
- Évaluer l'ergonomie et la facilité d'installation des logiciels de la liste.

### 6.1.2.2<sup>ème</sup> tentative

**Titre de l'étude :** Méthodologie d'évaluation d'utilisabilité de logiciel  
Perez Berro, Miriam

**Source :** ProQuest Dissertations and Theses; 2008; ProQuest Dissertations & Theses (PQDT)

## Résumé

### Problématique traitée

La problématique traitée dans cette étude c'est qu'il n'y a pas un outil d'évaluation générique que l'on peut appliquer à tous types de logiciel afin de permettre aux développeurs d'évaluer la facilité d'utilisation de leurs produits, et de vérifier la pertinence des fonctionnalités proposées en rapport avec les besoins. Ceci dit, qu'ils n'arrivent pas à s'assurer de la qualité d'utilisation de leurs produits logiciels. Comme résultats, les utilisateurs n'utilisent pas de manière efficace toutes les fonctionnalités du logiciel faute des erreurs de conception qu'ont commis les développeurs.

Pour pallier à cela, les auteurs ont proposé une méthodologie simple et légère pour évaluer l'utilisabilité des logiciels en question.

### But de l'étude

L'objectif de ce projet est de générer un modèle d'évaluation de l'utilisabilité des produits logiciels. La méthodologie suivie permet d'identifier les problèmes d'utilisabilité de produits logiciels en tenant compte de l'environnement de développement des produits et des contraintes de coûts, de temps et de performance.

## **Solution proposée**

Développer une méthodologie d'évaluation adaptée aux utilisateurs en prenant en considération les tâches de leur utilisation. Les auteurs se sont basés sur l'analyse des données du terrain et des prototypes d'évaluation pour pouvoir mettre en œuvre cette méthodologie d'évaluation d'utilisabilité.

La méthodologie d'évaluation d'utilisabilité proposée est supporté par les normes de qualité ISO/IEC 9126 et le guide de l'utilisabilité de l'ISO 9241.

## **7. Sélection et identification des logiciels libres les plus populaires utilisés aujourd'hui pour chaque activité du cycle de vie (ISO 12207)**

### **7.1. Méthode d'évaluation utilisée pour identifier quels sont les logiciels libres les plus populaires utilisés pour chaque activité de cycle de vie (ISO12207)**

Pour identifier les logiciels libres qui supportent des activités de génie logiciel (ISO 12207), il faut se doter d'une méthode de recherche qui permet d'identifier les principaux logiciels libres arrivés à maturité, puis d'une méthode d'évaluation afin de vérifier la pertinence des fonctionnalités proposées en rapport avec les besoins d'évaluation et de sélection.

### **7.2. Logiciels libres qui supportent des activités de génie logiciel**

Le génie logiciel étudie les méthodes de travail et les bonnes pratiques de praticiens qui développent des logiciels. Ces praticiens effectuent un certain ensemble d'activités de génie logiciel pour offrir de produits logiciels avec de bonnes performances aux attentes de clients tout en ayant un coût d'entretien réduit et en respectant les délais et les coûts de construction. Parmi ces activités de cycle de vie de logiciel, on retrouve l'analyse des besoins, l'élaboration des spécifications, l'architecture/conception, la programmation (construction/développement), les tests, la validation, la maintenance, la gestion de projet, la gestion de la qualité et la gestion de configuration. [1]

### **7.3. Méthode utilisée pour la recherche des logiciels libres**

Dans la littérature, et selon la communauté Open source, le répertoire de logiciels libres le plus important est Sourceforge. Cependant dans le cas de notre projet, pour identifier les logiciels libres qui supportent des activités de génie logiciel (ISO 12207), on va identifier en plus des logiciels libres du répertoire "Sourceforge.net", les logiciels libres et commerciales utilisés par les étudiants et le personnel académique dans les cours de programme de premier cycle en Génie logiciel/TI identifiés dans la figure1 présente dans ce document en annexe, et aussi dans la figure 2 dans l'article "Integrating Tools and frameworks in undergraduate software engineering curriculum" d'une part, et d'autre part, les logiciels libres identifiés, suggérés et utilisés pour chaque étape du cycle de vie logiciel par les professeurs et le personnel académique du département de génie de l'ETS identifiés au moyen d'un questionnaire /sondage réalisé sur SurveyMonkey en format électronique.

## **7.4. Logiciels libres et commerciales identifiés et suggérés lors d'un sondage réalisé au moyen d'un questionnaire électronique sur SurveyMonkey**

### **7.4.1. Questionnaire électronique sur SurveyMonkey**

À cette fin, on avait établi une version de questionnaire sur SurveyMonkey en format électronique et administrée en ligne sur le site de SurveyMonkey (<http://www.surveymonkey.com/s/5HWTC56>) qui identifie une technique de sondage d'opinion afin d'identifier les logiciels libres suggérés et utilisés pour chaque étape du cycle de vie logiciel par les professeurs et les étudiants du département Génie logiciel/TI.

Un message a été transmis à ce sujet par courriel aux enseignants pour solliciter leur participation à ce sondage.

### **7.4.2. Recherche de logiciels libres et commerciaux identifiés lors d'un sondage réalisé au moyen d'un questionnaire électronique sur SurveyMonkey**

#### **Phase d'analyse (fonctionnelle) ou de conception**

- **Analyse des besoins / Élaboration des spécifications**

**Q1.** Lors de l'analyse des besoins et de la définition d'exigences, quels outils/logiciels libres utiliserez-vous pour analyser, récolter des informations détaillées et éliciter des spécifications concernant l'éventail de fonctions logicielles que devra offrir le produit logiciel final?

#### **Réponses :**

- Gabarits OpenUP (OPEN Unified Process): licence publique;
- Word, Excel, PowerPoint : licence commerciale;
- La suite Adobe : licence publique;
- Open office: licence publique. Cette suite se compose de six outils :
  - ✓ Outil de traitement de texte (Writer);
  - ✓ Outil pour faire des présentations (Impress);
  - ✓ Tableur (Calc);
  - ✓ Outil pour faire du dessin (Draw);
  - ✓ Un gestionnaire de base de données(Base);
- Un éditeur de texte vim (un logiciel semblable à vi) : licence publique.

- **Conception**

**Q2.** Lors de la conception, quels outils/logiciels libres de modélisation utiliserez-vous pour déterminer et schématiser les mécanismes qui devront être programmés en vue d'obtenir des fonctions logicielles que devra offrir le produit final?

#### **Réponses :**

- UML (Visio) : licence commerciale;
- UML RUP : licence commerciale;
- Visual Paradigm for UML : licence publique;
- LucidChart (pour les diagrammes UML): {Basic, Pro, Team : licence commerciale};

- LucidChart (pour les diagrammes UML): licence publique;
- USE (pour OCL) : A UML-based Specification Environment: licence publique;
- UPPAAL (pour TimedAutomaton) : licence publique;
- CPN AMI (pour Petri nets) : licence publique;
- Romeo (pour Petri nets) : licence publique;
- TINA (pour Petri nets) : licence publique;
- Un éditeur de texte vim (un logiciel semblable à vi) : licence publique.

**Q3.** Lors de la modélisation, quels outils/logiciels libres utiliserez-vous pour créer et décrire vos bases de données?

**Réponses :**

- UML (Visio) : licence commerciale;
- UML RUP : licence commerciale.

**Q4.** Lors de la modélisation, quels outils/logiciels libres utiliserez-vous pour décrire votre architecture logicielle?

**Réponses :**

- PowerPoint: licence commerciale;
- UML (Visio) : licence commerciale;
- UML (Visio) avec Stencil UML : licence commerciale;
- UML (Visio) : licence commerciale;
- UML RUP : licence commerciale;
- Astah UM Logiciel de modélisation Astah : licence publique;
- Un éditeur de texte vim (un logiciel semblable à vi) : licence publique;

### **Phase de réalisation ou de programmation (écriture et tests des programmes)**

#### **• Construction/ développement**

**Q5.** Lors de la modélisation, quels outils/logiciels libres utiliserez-vous pour écrire vos programmes de code source en suivant les plan établis et la méthodologie et pratiques de développement choisies (itératif, Cascade, en spirale, agile (Extreme, Scrum, etc.)?)

**Réponses :**

- Eclipse édition JEE plus plugin Maven : licence publique;
- Eclipse IDE for Java Developers : licence publique;
- Eclipse RCP : licence publique;
- Nokia Qt: licence publique;
- C++: licence publique;
- Java: licence publique;
- Java3D: licence publique;
- NLTK with Python: Natural language toolkit with Python: licence publique;
- JESS Python: licence publique;

- Eclipse, plugins : licence publique;
- J2SE 1.5-6-7 : licence publique;
- Protégé, JessTab : licence publique;
- MS Visual Studio .NET : logiciel propriétaire;
- Netbeans : licence publique;
- TrollTech QT : licence publique.

**Q6.** Lors de la rédaction du code source quel outils/logiciels libres utiliserez-vous comme éditeur de texte?

**Réponses :**

- JEdit : licence publique;
- Qt Creator: licence publique;
- Eclipse: licence publique;
- Notepad++: licence publique;
- Java: licence publique;
- Un éditeur de texte vim (un logiciel semblable à vi) : licence publique.

**Q7.** Lors du débogage, quel outils/logiciels libres utiliserez-vous pour contrôler le déroulement de l'exécution du logiciel?

**Réponses :**

- Eclipse : licence publique;
- gdb (GNU debugger) : licence publique;
- Packet Tracer (Cisco) : licence publique;
- Eclipse pour le débogage : licence publique.

**Q8.** Lors de la gestion de versions quels outils/logiciels libres utiliserez-vous pour enregistrer des fichiers en conservant l'historique de toutes les modifications effectuées?

**Réponses :**

- CVS, SVN: licence publique;
- Wireshark (packet Analyzer) : licence publique;
- SVN et CVS pour la gestion des versions : licence publique;
- Révision Control System (RCS (un outil UNIX/linux qui prédate CVS)) : licence publique.

• **Tests**

**Q9.**Lors des tests unitaires, quels outils/logiciels libres utiliserez-vous pour vérifier l'exécution de vos programmes afin de déceler des bugs/défauts de programmation qui peuvent engendrer des résultats erronés?

**Réponses :**

- JUnit (framework de test unitaire) : licence publique;

- uispec4j (functional and/or unit testing library for Swing-based Java applications): licence publique;
- packet tracer (Cisco): Packet Tracer (Cisco) : licence publique;
- Wireshark packet Analyzer: licence publique;
- Make (outil de test): licence publique;
- Diff: outil de comparaison entre 2 fichiers seulement : licence publique;
- Python: licence publique.

- **Livraison**

**Q10.** Lors de la phase livraison, quels outils/logiciels libres utiliserez-vous pour la documentation logicielle?

**Réponses :**

- MS Word: licence commerciale;
- HTML (HyperText Markup Language): licence publique;
- fichier ASCII: Un éditeur de texte (documentation sous forme de fichiers ASCII): licence publique.

- **Maintenance**

**Q11.** Une fois que le logiciel a été mis à disposition des utilisateurs et durant lesquelles le logiciel subit des transformations, des corrections ou des améliorations, quels outils/logiciels libres utiliserez-vous pour effectuer des opérations de maintenance (analyse, programmation et test)?

**7.5. Recherche de logiciels libres et commerciales utilisés par les étudiants et le personnel académique dans les cours de programme de premier cycle en Génie logiciel/TI et identifiés dans la figure de l'article "Integrating Tools and frameworks in undergraduate software engineering curriculum" présente en annexe 3.**

La Figure présente dans ce document en annexe de l'article "Integrating Tools and frameworks in undergraduate software engineering curriculum" montre comment les différents types de technologies sont intégrés dans les cours du programme de premier cycle. Les cours sont présentés dans le tableau par ordre chronologique selon le programme de premier cycle en génie logiciel. Cette figure montre aussi que les technologies utilisées dans l'industrie sont adressés aux étudiants dès la première année.

**7.5.1. Sélection des logiciels libres (inventaire)**

- **Database & persistence**

- Derby : licence publique;
- Hibernate : licence publique;
- MyBatis : licence publique;
- Oracle 11g : licence commerciale;

- PostgreSQL : licence publique.
- **Debugger/Monitor/Simulator**
  - OPNET : licence publique;
  - Wireshark (packet Analyzer) : logiciel libre (licence publique).
- **Développement Environment**
  - Eclipse, plugins : licence publique;
  - J2SE 1.5-6-7 : licence publique;
  - Protégé, JessTab : licence publique;
  - MS Visual Studio .NET : licence propriétaire;
  - Netbeans : licence publique;
  - TrollTech QT : licence publique.
- **Graphics API**
  - OpenGL : licence publique;
  - Glut : licence propriétaire;
  - LibQGLViewer : licence publique pour logiciel libre/licence propriétaire pour logiciel commercial.
- **Library/API**
  - HTK Toolkit : licence propriétaire;
  - JacORB : licence publique;
  - Java SDK 3D, JMF : licence publique;
  - Spring Framework : licence publique;
  - MPI, pthreads : licence publique;
  - SAS : licence propriétaire;
  - Weka : licence publique.
- **Modeling and design**
  - IBM Rational Rose : licence propriétaire;
  - IBM Rational SDL Suite : licence propriétaire;
  - Lattix LDM : licence propriétaire.
- **Project management**
  - Apache Ant : licence propriétaire;
  - CVS/SVN client : licence publique;
  - IBM RUP : licence propriétaire;
  - Maven : licence publique;
  - Microsoft Project : licence propriétaire;
  - Trac : licence publique.
- **QA/Maintenance**

- Bugzilla : licence publique;
- CheckStyle, PMD : licence publique;
- CodeCover : licence publique;
- IBM Rational Logiscope : licence propriétaire;
- John the Ripper : licence publique;
- JUnit : licence publique;
- QALab : licence publique;
- uispec4j : licence publique;
- USE : licence publique.

- **Web development**

- Apache Tomcat : licence publique;
- Jetty : licence publique (licence Apache);
- WebLogic : licence propriétaire (licence Oracle).

## **7.6. Recherche des logiciels libres de cycle de vie du répertoire "Sourceforge.net"**

Dans le cadre de notre projet, nous allons définir une partie de la liste des logiciels libres à partir de l'annuaire et plateforme du répertoire "Sourceforge.net", le répertoire le plus important de logiciels libres de la communauté Open source. Cependant, la recherche se focalise uniquement pour les logiciels libres qui supportent des activités de génie logiciel (ISO 12207).

Une première sélection sera faite à partir du répertoire "Sourceforge.net", dont nous allons repérer une bonne vingtaine de projets libres, toute activité de cycle de vie confondue. Ces projets correspondront à des logiciels libres de la communauté open source qui supportent différentes activités de cycle de vie de génie logiciel. Le repérage des logiciels candidats sera assisté par l'application des modèles de maturité et d'évaluation du libre de Capgemini, de Golden et la grille d'évaluation de Wheeler.

## **7.7. Inventaire globale des projets de logiciels libres les plus populaires qui supportent des activités de génie logiciel**

Durant cette phase, nous allons regrouper l'ensemble des logiciels libres sélectionnés lors des trois stades d'inventaire. Le nouvel inventaire global inclura les logiciels libres sélectionnés lors des périodes de recherche suivantes :

- Recherche de logiciels libres identifiés lors d'un sondage réalisé au moyen d'un questionnaire électronique sur SurveyMonkey;
- Recherche des logiciels libres du répertoire "Sourceforge.net";
- Recherche de logiciels libres utilisés par les étudiants et le personnel académique dans les cours de programme de premier cycle en génie logiciel/TI. Ces logiciels sont identifiés dans la figure2 de l'article "Integrating Tools and frameworks in undergraduate software engineering curriculum" et ici présente en annexe.

## 7.8. Méthodologie de sélection à partir de l'inventaire globale

Dans le cas de notre projet, vu le nombre important des logiciels libres pour chaque activité de génie logiciel présents dans l'inventaire globale composé :

- des logiciels libres identifiés lors du sondage électronique sur SurveyMonkey;
- des logiciels libres utilisés par les étudiants et le personnel académique dans les cours de programme de premier cycle en Génie logiciel/TI identifiés dans la figure2 de l'article "Integrating Tools and frameworks in undergraduate software engineering curriculum" et présente ici en annexe.
- et aussi, par les logiciels libres du grand répertoire de la communauté open source "Sourceforge.net".

il est impossible d'évaluer la liste complète de tous ces logiciels. Pour résoudre ce problème, on va appliquer une suite de cribles aux différents logiciels libres de la liste pour éliminer les logiciels non pertinents afin d'alléger la liste et pour mieux évaluer pour chaque activité de génie logiciel des logiciels qui restent.

Une première sélection sera faite selon les critères suivants:

- Recherche de logiciels libres stables (version 1.0 ou supérieure);
- Recherche de logiciels libres ayant pour version beta (< 1.0) ne présentant pas de bug fatal à l'utilisation;
- Recherche de logiciels libres ayant des fonctionnalités indispensables;
- Les logiciels libres qui ne fonctionnent que sur Linux seront éliminés vu le nombre non important d'utilisateurs qui utilisent ces systèmes.

Dans la deuxième sélection du processus de sélection, on va utiliser les méthodologies d'évaluation et de sélection adaptés au libre (Wheeler, Golden, Capgemini) [3].

Crible1 - Choisir les logiciels pertinents et actifs : éliminer les logiciels inactifs ou non pertinents.  
Crible2 (application du OSMM de Golden) - Popularité et activité (nombre de références dans Google, nombre de téléchargements): utiliser des statistiques des projets pour éliminer les projets inactifs.

Crible3 – Analyser les fonctionnalités et services professionnels : installation, tests identiques pour tous les logiciels sélectionnés, bilan pour chaque logiciel, comparaison entre eux.

Sélection finale : établir un choix objectif entre les logiciels restants (logiciels libres ayant des avantages concurrentiels par rapport aux logiciels propriétaires).

## 8. Les critères dévaluation et de sélection qui vont nous permettre de choisir un logiciel libre d'une activité de génie logiciel plutôt qu'un autre

### 8.1. Méthode d'évaluation utilisée pour repérage des logiciels libres

Une fois la liste globale des logiciels libres de génie logiciel établie, nous allons utiliser une méthode d'évaluation pertinente pour le présent projet. Pour ce faire, nous allons utiliser une grille d'évaluation pour apporter une justification rationnelle, structurée et quantitative qui permet de justifier une décision à faire une sélection d'un logiciel de cycle de vie en particulier

dans un cadre formel. Cette grille d'évaluation va nous permettre d'établir des critères de comparaison pour chaque logiciel libre correspondant à chaque activité de cycle de vie de génie logiciel, les comparer les uns par rapport aux autres, et finalement en identifier un candidat par activité.

Dans la littérature, pour définir les critères de succès et d'échec d'un logiciel libre afin de trouver le produit logiciel open source le plus adapté, on doit appliquer certaines méthodes de sélection basées sur des grilles d'évaluation proposées dans le monde du logiciel libre. Parmi ces méthodes de sélection, on retrouve entre autres, le modèle de maturité et d'évaluation OSMM de Capgemini, la grille d'évaluation de Wheeler, et le modèle de maturité et d'évaluation OSMM de Golden.

## 8.2. Evaluation de l'utilisabilité de logiciels libres - Métriques d'utilisabilité

Une évaluation de l'utilisabilité est une analyse qualitative et quantitative de l'utilisabilité d'un logiciel libre d'une activité de cycle de vie.

Les métriques d'utilisabilité sont des mesures qui aident à déterminer le niveau d'utilisabilité d'un produit libre. Le processus d'évaluation sera supporté par l'ISO/IEC 14598-5. De même que la gestion des métriques sera supporté par les caractéristiques générales de la qualité de l'ISO/IEC 9126. Ces métriques mesurables comprennent [5]:

- Le temps qu'a pris un utilisateur pour réaliser une tâche;
- Le taux d'erreur commis par l'utilisateur (productivité);
- La période d'apprentissage et de formation nécessaire à l'utilisateur pour pouvoir réaliser les tâches;
- Le niveau de satisfaction des utilisateurs;
- Fonctionnalité : étude de la fiabilité de point de vue utilisateur;
- Conception visuelle : étude de la densité des éléments à l'écran;
- Efficience : pouvoir de réalisation d'une tâche dans un laps de temps déterminé;
- Simplicité d'apprentissage : nécessité ou non de la lecture de la documentation, interface simple;
- Navigation : facilité de déplacement dans le logiciel libre;
- Consistance : conventions conforme à l'utilisation;
- Rétroaction : rétroaction convenable et utile à l'utilisateur (aide en ligne, instructions d'utilisation claires, volet d'apprentissage);
- Accessibilité : adaptabilité à différents contexte d'utilisation.

## 8.3. La grille d'évaluation de Wheeler [3]

La grille d'évaluation de Wheeler comporte quatre étapes pour déterminer les critères qui permettront de choisir un logiciel libre plutôt qu'un autre. Ces étapes sont :

- Identifier;
- Évaluer;
- Comparer;
- Analyser.

### ✓ Identification

Wheeler suggère six sources d'informations pour identifier les critères de comparaison pour comparer les logiciels disponibles afin de pouvoir identifier un candidat qui pourra combler les besoins exprimés.

- **Les amis et collègues** : puisque les logiciels libres font peu de publicité, il faut compter entre autres sur les réseaux informels pour obtenir de l'information sur ces logiciels;
- **Les moteurs de recherche** : permettent de trouver des articles, des sites, des commentaires dans les forums avec un choix minutieux des termes et mots clés;
- **Les sites spécialisés** : SourceForge.net, FreshMeat.net;
- **Les distributions de Linux** : contiennent souvent de nombreux logiciels libres;
- **La liste de logiciels libres que la communauté reconnaisse comme matures**, c'est à dire, utiles, fiables, largement utilisés et largement supportés.

✓ **Evaluation**

Rechercher les indicateurs de popularité relative aux projets sélectionnés. Parmi ces indicateurs on énumère le nombre de hits dans les moteurs de recherche et le nombre de liens vers le site du projet libre en question.

✓ **Analyse**

Pour chaque logiciel libre sélectionné, nous allons étudier certains des aspects suivants à partir des informations disponibles sur leurs sites respectifs. Parmi ces aspects, on trouve : le code source, les fonctionnalités, le coût, la part de marché, le support, l'ergonomie, la sécurité, la facilité d'adaptation, l'intégration et la licence.

✓ **Résultat final**

En finale, une courte liste de candidats doit faire l'objet d'une analyse détaillée des critères sus-cités.

#### **8.4. Le modèle de maturité Open Source OSMM de Caggimini [3]**

La grille d'évaluation du modèle de maturité OSMM de Caggimini propose d'évaluer le logiciel libre selon 4 aspects, eux-mêmes subdivisées en 12 catégories pour déterminer les critères qui permettront de choisir un logiciel libre plutôt qu'un autre. Les 4 aspects et 12 catégories correspondantes sont :

- **Produit** : âge, hiérarchie du projet, originalité, licence, communauté de développement;
- **Intégration** : interopérabilité, modularité;
- **Utilisation** : facilité de déploiement, support;
- **Acceptation** : communauté d'utilisateurs, part de marché.

✓ **Evaluation**

En tant qu'évaluateur, pour chaque sous-catégorie des 4 aspects, nous allons accorder une note de 1 à 5. De la même façon, nous allons attribuer une valeur de 1 à 5 pour chaque caractéristiques environnementales suivantes : interface, robustesse, sécurité, absence de

dépendance à une plateforme, absence de dépendance à un fournisseur, support technique, disponibilité des services professionnels (conseil, formation, intégration).

### 8.5. Le modèle de maturité Open Source OSMM de Golden [3]

La grille d'évaluation du modèle de maturité OSMM de Golden, PDG de Navicasoft firme d'intégrateurs, propose d'analyser le produit et d'évaluer le logiciel libre selon 6 axes, évalué de 0 à 10 et pondéré pour un total de 100 points. Les 6 axes sont :

- **Produit** : fonctionnalités, qualité de code source (évaluer le temps moyen requis par la communauté de développeurs pour solutionner les problèmes et combien de problèmes sont ouverts), fiabilité, facilité d'utilisation;
- **Support** : support à l'utilisation, résolution des pannes;
- **Documentation** : livre, sites web;
- **Formation** : en classe, en ligne;
- **Intégration** : dans la pile d'applications et le portfolio de la compagnie;
- **Services professionnels** : disponibilité et qualité.

### 8.6. Sélection des logiciels libres matures

Après avoir défini la courte liste de différents logiciels libres candidats de chaque activité de génie logiciel, par la suite, nous allons procéder à établir une analyse détaillée des critères de comparaison, en utilisant une méthodologie de sélection pour ne déduire en finale qu'une petite liste des logiciels libres les plus prometteurs.

Le but de cette étape est d'identifier au plus deux ou trois logiciels libres approchant la maturité pour chaque activité de génie logiciel. Cette étape sera suivie des tests pour sélectionner qu'un seul, le meilleur candidat pour chaque activité de génie logiciel.

### 8.7. Sélection finale - Tests des logiciels libres matures

Après avoir identifié au plus deux ou trois logiciels libres approchant la maturité pour chaque activité de génie logiciel, et dans le but de recommander l'un de ces logiciels (pour chaque activité) pour une installation pilote (machine virtuelle pré-configurée), nous allons évaluer la liste réduite des projets libres de chaque activité sur 5 points. Le but est d'en sélectionner qu'un seul, le meilleur candidat pour chaque activité de génie logiciel : [3]

- **Communauté et part de marché** : détail sur la taille de la communauté, la structure de l'équipe de développement et le nombre d'utilisateurs connus;
- **Facilité d'installation**;
- **Fonctionnalité** : exécution d'un scénario de test couvrant les fonctionnalités de base pour vérifier ces fonctionnalités;
- **Ergonomie** : suivre une évaluation heuristique. Examen systématique de l'interface humain-machine pour en déterminer la conformité avec un ensemble de règles de l'art empiriques ou heuristiques;
- **Extension/personnalisation** : évaluer la facilité d'adaptation du logiciel aux besoins des développeurs.

## **9. Intégration des logiciels libres de cycle de vie dans une machine virtuelle unique afin de supporter un spécialiste afin de ne pas nécessiter trop de formation/difficultés d'apprentissage**

Notre projet vise à chercher sur internet des logiciels libres pour le génie logiciel, comme pour la spécification des exigences ou la gestion de projet etc. Donc j'ai à chercher des logiciels open source ou des équivalents en logiciels libres et gratuits qui font la même chose que MS Project par exemple (entre autres ganttProject, open Workbench, Openproj & TaskJuggler, etc...). Il s'agit donc de les étudier selon différents critères de comparaison, et par la suite je vais proposer le meilleur. En finale, il s'agit de l'installer et le tester dans une machine virtuelle comme VMware par exemple qui sera utilisée par la suite par les usagers d'une manière gratuite. En d'autres termes, celui qui va l'utiliser n'aura même pas à installer et le configurer mais juste à copier la machine virtuelle.

### **9.1.1. Défis**

#### **9.1.1.1. Les défis techniques**

Vu le nombre important des technologies proposées aux étudiants des cycles supérieurs dans les travaux pratiques associées au cours offerts dans les programmes de premier cycle en génie logiciel, le département de l'ETS opte pour certaines techniques au niveau du choix technologique pour le type de déploiement au niveau des laboratoires. Parmi les techniques utilisées pour le déploiement au niveau des postes de travail, on retrouve entre autres, la technique du multiboot permettant d'installer plusieurs systèmes d'exploitation Windows ou Linux sur un même ordinateur, et la technique utilisée par les postes MAC utilisant des machines virtuelles Windows pour mettre en œuvre du multitâches afin de permettre à un seul ordinateur de réaliser plusieurs opérations simultanées au service de plusieurs utilisateurs.[4]

#### **9.1.1.2. Le défi d'apprentissage**

Pour surmonter les défis de difficultés d'apprentissage et de formation, à mon avis, on doit organiser des tutoriels et didacticiels d'outils open-source pour permettre aux étudiants de se familiariser seules avec les fonctionnalités importantes des nouvelles technologies. Le but, est de permettre à ces étudiants d'acquérir de nouvelles expertises et de nouvelles expériences pour accomplir des exercices pratiques et objectifs fixés lors des séances de laboratoire.

## Références et Bibliographie

[1][http://fr.wikipedia.org/wiki/Richard\\_Stallman](http://fr.wikipedia.org/wiki/Richard_Stallman)

[2][http://scholar.googleusercontent.com/scholar?q=cache:5F4XjEvj\\_QYJ:scholar.google.com/&hl=en&as\\_sdt=0,5](http://scholar.googleusercontent.com/scholar?q=cache:5F4XjEvj_QYJ:scholar.google.com/&hl=en&as_sdt=0,5)

[3]Evaluation et expérimentation de logiciels libres pour la PME, Michel Sauvé, Mario Boutin, Robert Gérin Lajoie, Isabelle Therrien, 2005, Google Scholar  
<http://www.cirano.qc.ca/pdf/publication/2005RP-17.pdf>

[4]C. Fuhrman, R. Champagne, A. April, "Integrating Tools and frameworks in undergraduate software engineering curriculum", Department of Software and IT Engineering, ETS, University of Québec, Montréal, Canada.

[5]Méthodologie d'évaluation d'utilisabilité de logiciel, Perez Berro, Miriam, ProQuest Dissertations and Theses; 2008; ProQuest Dissertations & Theses (PQDT), pg. n/a.

## Annexe

URL du questionnaire du sondage électronique : (<http://www.surveymonkey.com/s/5HWTC56>)

### Phase d'analyse (fonctionnelle) ou de conception

- **Analyse des besoins / Élaboration des spécifications**

**Q1.** Lors de l'analyse des besoins et de la définition d'exigences, quels outils/logiciels libres utiliserez-vous pour analyser, récolter des informations détaillées et éliciter des spécifications concernant l'éventail de fonctions logicielles que devra offrir le produit logiciel final?

- **Conception**

**Q2.** Lors de la conception, quels outils/logiciels libres de modélisation utiliserez-vous pour déterminer et schématiser les mécanismes qui devront être programmés en vue d'obtenir des fonctions logicielles que devra offrir le produit final?

**Q3.** Lors de la modélisation, quels outils/logiciels libres utiliserez-vous pour créer et décrire vos bases de données?

**Q4.** Lors de la modélisation, quels outils/logiciels libres utiliserez-vous pour décrire votre architecture logicielle?

### Phase de réalisation ou de programmation (écriture et tests des programmes)

- **Construction/ développement**

**Q5.** Lors de la modélisation, quels outils/logiciels libres utiliserez-vous pour écrire vos programmes de code source en suivant les plan établis et la méthodologie et pratiques de développement choisie (itératif, Cascade, en spirale, agile (extreme, Scrum, etc.)?)

**Q6.** Lors de la rédaction du code source quel outils/logiciels libres utiliserez-vous comme éditeur de texte?

**Q7.** Lors du débogage, quel outils/logiciels libres utiliserez-vous pour contrôler le déroulement de l'exécution du logiciel?

**Q8.** Lors de la gestion de versions quels outils/logiciels libres utiliserez-vous pour enregistrer des fichiers en conservant l'historique de toutes les modifications effectuées?

**Q9.** Lors de l'optimisation du code quels outils/logiciels libres utiliserez-vous?

- **Tests**

**Q10.** Lors des tests unitaires, quels outils/logiciels libres utiliserez-vous pour vérifier l'exécution de vos programmes afin de déceler des bugs/défauts de programmation qui peuvent engendrer des résultats erronés?

- **Livraison**

**Q11.** Lors de la phase livraison, quels outils/logiciels libres utiliserez-vous pour la documentation logicielle?

- **Maintenance**

**Q12.** Une fois que le logiciel a été mis à disposition des utilisateurs et durant lesquelles le logiciel subit des transformations, des corrections ou des améliorations, quels outils/logiciels libres utiliserez-vous pour effectuer des opérations de maintenance (analyse, programmation et test)?