Analyse, évaluation et amélioration de la performance du processus de développement de logiciels libres : une approche par la nouvelle norme ISO/CEI 29110

NORMAND SÉGUIN, CLAUDE LAPORTE ET ADÈLE LARISSA

Résumé : Les logiciels libres (F/OSS, *Free and Open-Source Software)* ont atteint une grande popularité depuis 1983 et sont devenus une alternative viable à plusieurs logiciels commerciaux grâce à certains produits tels que Linux, Apache ou encore Mozilla. Bien que certains logiciels libres soient d'une qualité élevée, comparable, parfois supérieure à leurs concurrents commerciaux [2, 3, 6], les logiciels libres font malgré tout face à de nombreux défis [3, 7] et de nombreux problèmes de qualité [3, 4, 5]. Les logiciels développés ou maintenus sous le modèle de développement de F/OSS ne suivent habituellement pas les pratiques de qualité des modèles de développement reconnus en ingénierie [1]. Malgré un nombre important de recherches sur les F/OSS, peu d'entre elles se sont penchées sur le processus de développement de F/OSS et l'amélioration de la qualité de ce processus.

Le but de cette étude a été d'évaluer la possibilité qu'une norme de développement de logicie[i], telle que la nouvelle norme ISO/IEC 291101 [12], puisse soutenir ou améliorer le processus de développement des F/OSS. À cet effet, une analyse de projets Linux, Mozilla, Apache et GNOME a été faite pour se familiariser avec le développement libre. Ce premier volet a permis de recueillir des informations sur les pratiques d'assurance qualité logicielle appliquées sous le modèle de développement des F/OSS et d'identifier les problèmes et les défis qu'ont rencontrés les F/OSS dans les années passées. Dans le second volet, le processus de développement libre extrait de cette analyse est évalué par rapport à la norme ISO/IEC 29110. Cette évaluation a donné comme résultats l'identification des différences entre le développement libre et le développement normé. Par ailleurs, cette évaluation a permis d'identifier que certains des résultats obtenus rejoignent la littérature, notamment concernant le problème de documentation des phases du développement libre. Une évaluation a aussi été réalisée pour évaluer la compatibilité du profil basique de la norme ISO/IEC 29110 avec la culture et la philosophie des F/OSS. Elle a permis d'identifier les activités du profil basique de la norme ISO/IEC 29110 qui pourraient soutenir le développement libre. Finalement, à partir des résultats de ces deux évaluations, une proposition est faite pour améliorer les performances du processus de développement libre.

Mots clés et sigles : F/OSS (*Free and Open-Source Software*), Processus de développement, logiciels libres, valeurs, principes, assurance qualité, norme du génie logiciel, ISO/IEC 29110, qualité, logiciels Open Source, développement fermé, développement classique de logiciels.

1. INTRODUCTION

La présente étude se concentre sur la qualité dans le développement des logiciels libres et open source (F/OSS) et sur l'amélioration des performances du processus de développement des F/OSS. À cet effet, des projets comme Linux, Apache, Mozilla et ►

GNOME ont été étudiés. Bien que le processus de développement des projets F/OSS ait permis d'aboutir à des logiciels d'une qualité élevée, comparable ou supérieure aux logiciels développés en entreprise [2, 3], les F/OSS font malgré tout face à un certain nombre de problèmes de qualité [2, 3, 7, 9], tels que : le manque de documentation et les problèmes dans la communication et la coordination, le problème de gestion des configurations, le code non maintenu [9] et le contrôle de maintenance à long terme [3]. Ainsi, comment procéder pour améliorer la qualité du logiciel développé dans un contexte de F/OSS ? Cette étude porte principalement sur l'apport de la nouvelle norme ISO/IEC 29110 [12] dans l'amélioration du processus de développement des F/OSS, en tenant compte de leur culture et de leur philosophie. La norme et les rapports techniques ISO/IEC 29110 ont été développés spécifiquement pour les très petits organismes (TPO) qui développent du logiciel. Les TPO sont des entreprises, des organismes, des départements ou des projets comportant 25 personnes ou moins. Le document ISO/IEC TR 29110- 5-1-2 [12] décrit un processus de gestion et un processus de mise en œuvre en tenant compte d'informations suffisantes pour qu'un TPO puisse décrire son propre processus de développement [52, 53]. La culture selon Wiegers [58] est « un ensemble de valeurs[2] partagées, des objectifs et des principes[3] qui guident les comportements, les activités, les priorités et les décisions d'un groupe de personnes travaillant vers un objectif commun ». Les caractéristiques ou valeurs des projets libres identifiées par Michlmayr, Hunt et Probert [9] sont : 1) les projets de logiciels libres sont pour la plupart du temps disséminés à travers le monde ; 2) les participants des projets de logiciels libres sont généralement des volontaires bénévoles. Par ailleurs, une autre valeur du libre est : les participants sont libres de choisir les tâches qu'ils désirent réaliser [10] en rapport à leur expertise ou leur centre d'intérêt. Le terme philosophie quant à lui en son sens le plus large, vient du mot grec *philosophia* qui consiste en l'exercice systématique de la pensée et de la réflexion. Selon Larousse [60], l'une des définitions de la philosophie consiste en : « un système d'idées qui se propose de dégager les principes fondamentaux d'une discipline ». Par ailleurs, Larousse propose une définition alternative de philosophie qui est : « la manière de voir, de comprendre, d'interpréter le monde, les choses de la vie, qui guide le comportement ». En associant ces deux définitions au développement, le terme pourrait s'appréhender à : la façon ou le comment se fait le développement et qui permet de ce fait de dégager les principes fondamentaux de ce développement. En ce sens, le terme philosophie est en rapprochement avec un modèle de développement, ce qui convient parfaitement au contexte du libre.

Une des motivations principales de cette étude est que les communautés de F/OSS utilisent un mode de développement peu classique mais efficace, offrant certains avantages économiques comparativement au développement en entreprise qualifié très souvent de développement à code source fermé. Deux modèles de développement sont rencontrés dans les F/OSS depuis leur apparition. Le premier modèle a été identifié par Raymond en 1999 [8] qui l'a qualifié de « bazar », suite à la nature non-organisée et ouverte au public du développement libre. Le second modèle de développement libre est apparu en 2007 et porte le nom de modèle en oignon et est basé sur l'aspect social des participants aux projets F/OSS [50]. Ce dernier conserve les meilleures pratiques et principes utilisés habituellement dans le développement libre depuis sa création, comme par exemple, « traitez vos utilisateurs comme des co-développeurs », ou, « distribuez tôt, mettez à jour régulièrement », ou encore, « distribuez vite et souvent, déléguez tout ce que vous pouvez déléguer et soyez ouvert jusqu'à la promiscuité » [8].

Les F/OSS sont développés par des programmeurs, distribuant le code source dans un environnement collaboratif, répartis virtuellement et géographiquement et basé sur l'Internet [1]. Cependant, selon la qualité des contributions qu'un programmeur peut avoir soumis à sa communauté de F/OSS, il gagnera en mérite et pourra se voir confier d'autres responsabilités et ainsi jouer un rôle important dans le développement du projet. On parle ainsi du processus de méritocratie[4] [6, 50] dans le développement libre. La nature du développement des projets F/OSS est associé à de nombreux défis [23] et à de nombreux problèmes de qualité [2, 3, 7, 9] qui ont comme origines le bénévolat des participants au projet [3, 9] et la nature distribuée et pas toujours bien organisée des projets.

Le bénévolat favorise la créativité chez les développeurs, les développeurs sont libres de choisir les tâches du projet qu'ils veulent réaliser en relation avec leur centre d'intérêt et leur expertise. Ils sont motivés et passionnés, et sont plus ouverts à la créativité qui est propice dans certains cas à l'innovation. Par ailleurs, le bénévolat des participants favorise la rapidité du développement ainsi qu'une détection et une correction rapides des défauts par de nombreux intervenants (utilisateurs et développeurs) [7, 8] bien avant la phase de codage [8, 51]. Malheureusement, le fait que les participants soient bénévoles entraîne dans certains cas les problèmes suivants : des échecs de projets lorsque le nombre des développeurs est inférieur à 15 [3]; du code non maintenu lié à l'abandon du projet ou d'une activité en cours par certains développeurs; et des lacunes importantes de documentation [9]. Les développeurs sont généralement attirés par des tâches liées à leur expertise et dont ils ont une réelle passion. De plus, la nature non-structurée [11] et non soutenue par des normes de développement du génie logiciel,

génère souvent des différences importantes comparativement au développement de logiciels dans les organisations (qualifié de développement fermé) [10]. Dans les entreprises, ces différences conduiraient par inadvertance à un échec du projet selon le développement traditionnel de logiciel [7], entraînant ainsi des débordements de coûts et des délais supplémentaires.

Cette étude a analysé et a évalué le processus de développement des F/OSS de quatre projets et elle a permis d'identifier les différences entre le développement libre et le développement classique de logiciels. Le développement classique fait référence aux cycles séquentiels de développement, tel que le modèle en cascade ou le modèle en V, qui sont très souvent utilisés dans les organisations, car ils sont centrés sur les livrables ou des documents, fournis à chaque étape du cycle de développement de logiciel, afin de s'assurer de la réalisation d'un processus ou d'activité, et ainsi permettre un meilleur suivi du projet. Par la suite, nous avons réalisé une étude de la compatibilité de la norme ISO/IEC 29110 avec la culture et les valeurs et la philosophie des F/OSS. Nous avons débuté par l'identification de certains critères de compatibilité avec le contexte des F/OSS. De ces critères, une évaluation de la compatibilité du profil basique de la norme ISO/IEC 29110 avec les F/OSS a été réalisée. Les résultats de cette évaluation de même que la connaissance des écarts en termes de qualité entre le développement libre et le développement classique de logiciels sont le point de départ d'une proposition pour améliorer le processus de développement libre à l'aide du profil basique de la norme ISO/IEC 29110.

La section 2 présente les problèmes de qualité rencontrés sous le modèle du développement libre. La méthode utilisée est présentée à la section 3. La section 4 présente les résultats obtenus, la section 5 présente une proposition d'amélioration du processus de développement libre. Finalement, la section 6 présente les principales conclusions tirées de cette étude.

2. Assurance qualité et problèmes de qualité sous le modèle de développement libre

L'assurance qualité est définie comme étant : « 1) un ensemble d'activités planifiées et systématiques de toutes les actions nécessaires pour fournir l'assurance suffisante qu'un élément produit est conforme aux exigences techniques établies ; 2) un ensemble d'activités destinées à évaluer le processus par lequel les produits sont développés ou fabriqués ; 3) les activités planifiées et systématiques mises en œuvre, dans le système qualité, et démontrées au besoin pour fournir l'assurance suffisante qu'une entité satisfera aux exigences de qualité (ISO/IEC/IEEE 24765 » [49].

Malgré le succès de certains produits logiciels libres et le niveau de qualité élevé, les logiciels F/OSS ne suivent habituellement pas des pratiques normées d'assurance qualité logicielle [4, 6]. Des études empiriques des activités d'assurance qualité logicielle sous le modèle de développement libre F/OSS ont été réalisées [2, 37], ainsi que des études sur les pratiques et les processus permettant aux logiciels F/OSS d'aboutir à un niveau élevé de qualité [6, 7, 8, 9, 10, 11, 26, 38, 50, 51]. Ces recherches montrent que les logiciels sont développés et maintenus par une multitude de personnes bénévoles (généralement des développeurs [8]), réparties géographiquement à travers le monde [9] et communiquant essentiellement par Internet [8, 10]. Le développement de logiciels F/OSS est asynchrone, ouvert [9], itératif [10] et incrémental [31], et se fait de manière parallèle [7]. La communauté contribue au projet par des retours sous forme de rapport d'erreurs, procède aux revues de code ou suggère de nouvelles fonctionnalités ou de nouvelles exigences [7, 8]. Elle détecte et corrige la majorité des défauts [6, 8, 11, 38]. Le code ainsi que tous les documents du projet sont disponibles pour tous les utilisateurs [9, 19]. Tous les processus du développement libre sont transparents et ouverts aux utilisateurs [9], ce qui offre une visibilité de l'état du développement pour tous les participants au projet de F/OSS ou les utilisateurs du produit logiciel. Le développement de manière collaborative par l'Internet a conduit les développeurs de logiciels F/OSS à utiliser des outils de suivis d'erreurs, de gestion de configuration et dans certains cas de construction et de tests automatiques (voir le tableau 3 en annexes). Cependant, il y a souvent un manque de documentation. Les documents de conception sont souvent absents, le code source peut rester non maintenu lorsque les développeurs quittent le projet ou lorsque des problèmes complexes surviennent dans les tests lorsque les développeurs ont un accès limité aux configurations de diverses plates-formes de développement [37]. D'autre part, il y a aussi des problèmes de coordination et de communication, des problèmes de configuration [9], de même que des problèmes de mise à jour [7, 9]. La majorité des projets F/OSS gardent généralement trace des défauts de code, mais moins des défauts dont l'origine est les exigences et la conception [37]. Il y a une pratique accrue, récurrente et itérative des revues par les pairs dans les F/OSS [3, 4, 9, 26, 50, 51] depuis la phase des exigences jusqu'à la livraison du produit logiciel, permettant de détecter et de corriger le plus tôt possible les défauts du logiciel.

La littérature sur le développement des logiciels F/OSS à succès, sur la qualité et les pratiques d'assurance qualité des logiciels F/OSS, ainsi que les informations trouvées sur les wiki et les sites web des projets Linux, Apache, Mozilla et GNOME, constituent une base riche d'informations qui est à la base de cette étude.

3. LA MÉTHODE D'ANALYSE UTILISÉE

Soutenue principalement par la nouvelle norme ISO/IEC 29110 développée pour les TPO, l'approche que nous avons utilisée pour analyser le processus de développement des logiciels F/OSS est une approche analytique, basée sur l'aspect social des développeurs de logiciels F/OSS. Les éléments analysés sont les activités[5], les rôles et les artefacts produits. Le profil basique de la norme ISO/IEC 29110 [12] a été choisi comme cadre de référence, car il décrit un minimum d'activités, de rôles et d'artefacts de développement de logiciels qui devraient se retrouver dans un projet de développement logiciel. La figure 1 illustre les principaux éléments du processus de gestion de projet et d'implémentation du profil basique de la norme ISO/IEC 29110. Également, des tableaux ont été développé (figure 2 pour le gabarit) pour évaluer les différents aspects du processus de développement des F/OSS.

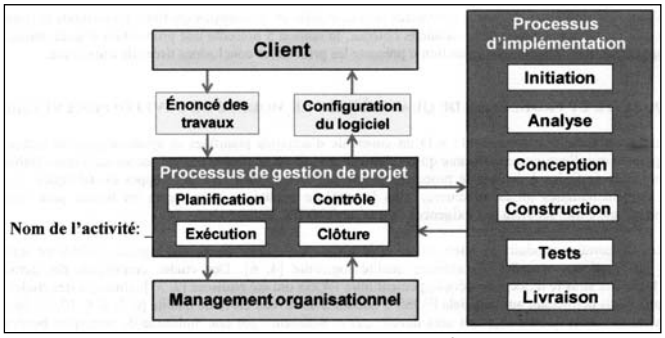

Figure 1 : Illustration des deux processus du profil basique de la norme ISO/IEC 29110 (traduit de Varkoi 2010 [65])

Pratiques définies par le profil basique de l'ISO/IEC 29110		Analyse de la réalisation de l'activité et identification des rôles et des artefacts pour chaque activité												
Référence de la tâche	Description de la tâche	Parfaitement exécutée	Particlement exécutée	Non exécutée	Aucune connaissance de l'existence de cette tâche	Artefacts ou Commentaires	CUS	AN	DES	PR	TL	PM	WT

Légende :
CUS : client
AN : analyste
DES : concepteur
PR : programmeur
TL : Leader technique
PM : gestionnaire de projet
WT : équipe de travail

☐ Tâches et sous-tâches des activités et tâches définies du profil basique de la norme ISO/IEC 29110

☐ Critères d'analyse

☐ Notes disant comment sont réalisées chaque tâche de l'activité de développement libre et quels sont les rôles associés à chacune des tâches

Figure 2 : Gabarit du tableau d'évaluation du processus de développement des F/OSS

Les lecteurs qui désirent se familiariser avec la norme ISO/IEC 29110 peuvent consulter les ouvrages suivants [66 et 67] et le site suivant : *http://profs.logti.etsmtl.ca/claporte/VSE/Groupe24-menu.html*.

Les communautés de développement F/OSS choisies pour cette étude sont les projets Linux, Apache, GNOME et Mozilla. Ces projets sont populaires, ont connu un succès et maintenu un niveau de qualité élevée dans les produits logiciels développés ou maintenus. Des recherches d'informations ont été faites sur le développement des produits logiciels libres dans ces communautés pour identifier les différents artefacts, les rôles, les tâches, ainsi que les différentes pratiques de qualité utilisées. Dans la norme ISO/IEC 29110, un élément du développement de logiciels, comme l'analyse des exigences, est appelé « activité ». Dans le cadre de cette étude, nous avons souvent utilisé le terme « processus » à la place du terme « activité ». En se basant sur le profil basique de la norme ISO/IEC 29110, les processus libres de développement de logiciels analysés sont : *l'initiation d'implémentation de logiciel, l'analyse des exigences, l'architecture et la conception, la construction logicielle, l'intégration et les tests, et la livraison de produit.*

Seul le processus d'implémentation de logiciel de la norme ISO/IEC 29110 a été analysé dans cette recherche. Bien que la gestion de projet ait un impact considérable sur la qualité du logiciel, nous n'avons pas analysé ce volet, car les projets F/OSS n'ont généralement pas un processus bien défini de gestion de projet [10]. Les projets de F/OSS ont souvent tendance à aller directement à la programmation [6].

Nous avons observé des écarts considérables dans les coûts et les délais de développement de logiciels libres par rapport à ceux du développement classique de logiciels, car habituellement dans des projets F/OSS, il n'existe pas de calendrier de révisions et dates de livrables du projet [10] nécessaires pour le respect des délais et des coûts. La plupart des projets F/OSS optent à cet effet pour la conduite de revues par les pairs récurrentes, régulières et itératives [8, 51], pour obtenir le plus tôt possibles des retours usuels. Néanmoins, certains projets, tel que GNOME [26] ont un équipe d'assurance qualité dont le rôle est de : « planifier et de coordonner l'ensemble du projet, de manière à ce que le projet suive le calendrier des révisions (revue de pairs) proposées. L'équipe d'assurance qualité est également chargée d'élaborer, en coordination avec les chefs de projets, les calendriers de livraison et de révision pour chacun des différents modules et la planification de tout le projet ».

La gestion de projet en logiciel libre telle que définie par Mockus, Fielding et Herbsleb [10] est : « l'initiation et la planification d'un projet, de manière à attirer le maximum de personnes ». Comparativement au développement de logiciel tel que prescrit par la norme ISO/IEC 29110, nous avons observé que certaines activités et tâches du processus de gestion de projet de la norme ISO/IEC 29110 peuvent se retrouver dans le processus d'implémentation du logiciel dans le développement libre, particulièrement lors de l'activité d'initiation d'implémentation de logiciel.

Au niveau de l'activité de démarrage de l'implémentation du projet libre, les activités du profil basique de la norme ISO/IEC 29110 réalisées dans le développement libre sont :

- L'activité de planification du projet. Bien qu'elles diffèrent d'un projet de logiciel F/OSS à un autre, les tâches, même partiellement réalisées, sont :
- Définir l'état du travail et les tâches qui doivent être réalisées dans le projet. Les projets sont habituellement divisés en modules pour faciliter le développement [6, 26, 51]. De plus, certains projets F/OSS, tel GNOME [26] ou Mozilla [11], planifient les dates des versions livrables. Cependant, la composition de l'équipe de développement ainsi que les rôles requis dépendront de l'intérêt du développeur puisque les participants choisissent habituellement les tâches qu'ils veulent effectuer [10].
- Définir la stratégie de contrôle de versions du logiciel. Elle diffère d'une communauté de F/OSS à une autre. Certains projets, comme Linux, utilisent des branches pour différencier les versions du système [26, 57].
- Identifier les risques qui pourraient affecter le projet. Évaluer si le logiciel attirera de nombreux participants et s'il attirera une communauté active de développeurs [10, 23, 26, 40]. La décision de la faisabilité du projet est prise par rapport aux spécifications et également par rapport à l'attraction des participants au projet [10, 26].

Il n'y a habituellement pas de processus prédéfinis ou explicite de gestion de projet dans le développement libre. La gestion de projet est habituellement implicite au développement.

La méthode d'analyse du processus de développement des logiciels F/OSS est divisée en deux volets :

1- L'évaluation du processus de développement de logiciel libre afin de déterminer les différences entre les processus du libre et les processus de développement classiques de logiciels. Les activités évaluées sont : *le démarrage de l'implémentation de logiciel, l'analyse des exigences, l'architecture et la conception, la construction logicielle, l'intégration et les tests, la livraison de produit* ;
2- L'analyse de compatibilité des activités de la norme ISO/IEC 29110 avec le développement de logiciels F/OSS afin d'identifier les activités de la norme compatibles avec le développement F/OSS qui pourraient aider à améliorer le processus de développement libre.

Nous présentons dans les sections 3.1 et 3.2 les méthodes d'évaluation et d'analyse et nous présentons les résultats dans la section 4

3.1 Méthode d'évaluation du processus de développement de logiciels libres

Basé sur l'observation des pratiques de développement des F/OSS faite à partir des informations sur les projets Linux, Apache, GNOME et Mozilla, le processus de développement des F/OSS est évalué par rapport au processus d'implémentation de la norme ISO/IEC 29110. Pour chacune des activités de développement des F/OSS, les tâches et les sous-tâches sont évaluées par rapport à la norme. Cette évaluation permet de relever les différences entre le développement libre et le développement classique de logiciel, de même que les différences entre le développement libre et le développement fermé[6] de logiciel. Les tâches des activités du développement libre ont été identifiées, de même que les rôles et les artefacts produits en sortie par chacune de ces tâches. Cette évaluation du processus de développement libre a permis d'identifier des lacunes de qualité du processus de développement des F/OSS. La figure 3 présente la méthode utilisée pour évaluer chaque tâche du processus de développement de F/OSS par rapport au processus d'implémentation de la norme ISO/IEC 29110.

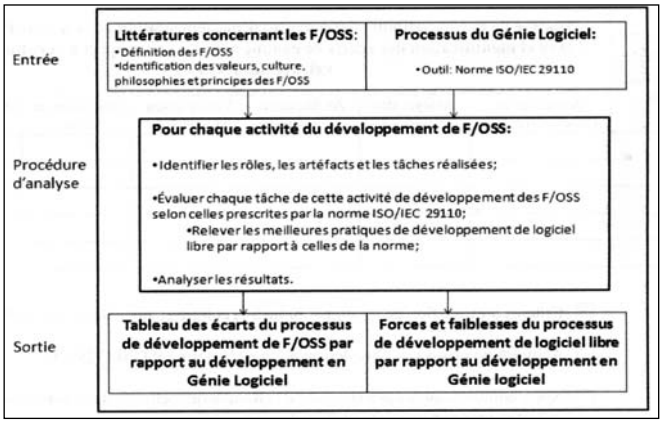

Figure 3 : Description de l'analyse du processus de développement de logiciel libre

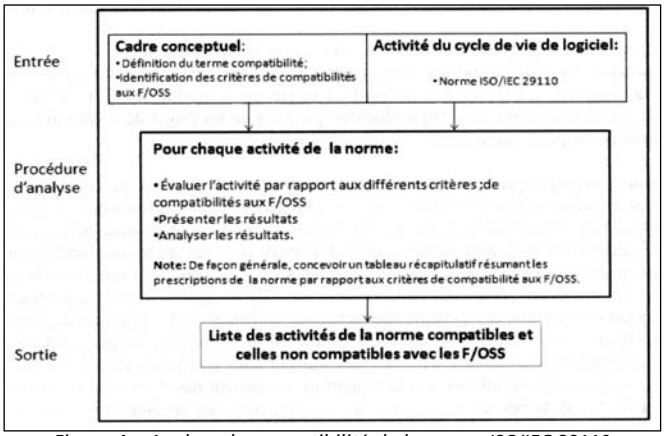

Figure 4 : Analyse de compatibilité de la norme ISO/IEC 29110 avec les logiciels libres

3.2 Méthode d'analyse de la compatibilité de la norme ISO/IEC 29110 avec le développement de logiciels libres

La figure 4 présente la méthode que nous avons utilisée pour évaluer la compatibilité de la norme ISO/IEC 29110 avec les logiciels F/OSS. Chaque activité du processus d'implémentation de la norme ISO/IEC 29110 est évaluée par rapport aux valeurs et à la philosophie de développement des F/OSS. À cet effet des tableaux d'analyse ont été conçus selon

le gabarit de la figure 5 afin d'évaluer la compatibilité de la norme ISO/IEC 29110 avec les logiciels F/OSS.

Critères de compatibilité relatifs aux F/OSS	Analyse de la compatibilité des activités d'implémentation de la norme ISO/IEC 29110 et identification des écarts de chaque activité par rapport à la culture et aux valeurs des F/OSS					
	Démarrage de l'implémentation	Analyse des exigences	Architecture et conception	Construction logicielle	Intégration et tests	Livraison du produit

Légende :

☐ Critères d'évaluation de la compatibilité à la culture et aux valeurs des F/OSS

☐ Activités du processus d'implémentation de la norme ISO/IEC 29110

☐ Notes commentant les prescriptions de chacune des activités de la norme ISO/IEC 29110 par rapport aux critères d'évaluation

Figure 5 : Gabarit d'évaluation de la compatibilité de la norme ISO/IEC 29110

3.2.1 Développement des critères de compatibilité avec les F/OSS

Le développement des critères de compatibilité avec les F/OSS s'est faite par une étude des projets Linux, Apache, Mozilla et GNOME, et particulièrement des sous-projets suivants : le noyau Linux, Apache http server, GNOME (car il est lui-même un sous-projet du projet GNU/Linux), GCC et Firefox. Le choix de ces sous-projets s'est basé sur leur popularité au sein de leurs communautés respectives. Le but recherché est de faire ressortir les similitudes des pratiques communes à chacun des sous-projets étudiés. Ces similitudes nous ont servi à développer les critères de compatibilité.

Pour identifier ces similitudes, les questions posées portent sur les aspects suivants :

- *La taille des projets.* La taille des projets de logiciels F/OSS, exprimée sous la forme du nombre de participants au projet, est-elle un obstacle à l'application au profil basique de la norme ISO/IEC 29110 aux logiciels F/OSS ? L'enquête menée par Aberdour en 2007 [6] a identifié que pour qu'un projet de F/OSS ait des chances de réussir, le nombre minimal de participants serait de 15.
- *La répartition géographique des participants au projet.* Les participants sont-ils localisés ou répartis géographiquement à travers le monde ? Quels moyens utilisent-ils pour communiquer et collaborer ? Doivent-ils se rencontrer physiquement ou doivent-ils échanger via l'Internet ? Fort heureusement, bon nombre de ces questions ont déjà une réponse dans certaines études menées sur les F/OSS [9,10]. Ces études ont identifié que les participants sont repartis géographiquement à travers le monde [9]. En 2002, Mockus *et al.* [10] ont identifié que les participants aux projets F/OSS ne se rencontrent jamais. La collaboration entre développeurs se fait via les listes de diffusion (*mailing lists*) des développeurs ou à l'aide des systèmes de partage de contenus ou des systèmes de messagerie instantanée, tels que IRC, Facebook ou encore Twitter (voir tableau 3, pour d'autres exemples). De plus, certains projets libres à l'instar d'Apache, font des réunions via Skype [56]. Mais très souvent dans les projets F/OSS, l'usage des systèmes utilisant des vidéos ou du son ne sont pas utilisés, car la plupart ne permettent pas d'archiver les communications échangées, raison pour laquelle les systèmes de messagerie, les *mailing lists* des développeurs et les systèmes de partage de contenu sont privilégiés.
- *Les pratiques de développement relatives aux projets libres retenus.* Comment les tâches à réaliser sont-elles reparties entre les différents participants au projet ? Quelles pratiques de qualité sont utilisées dans les projets libres retenus ? Quels rôles sont associés à chaque phase ou à chaque activité du développement libre ?

Des observations des projets F/OSS faites à partir des sites web de chacun des projets Linux, Apache, Mozilla, GNOME et des sous-projets libres retenus [13, 14, 15, 16, 17, 18, 19, 20, 21], de certains travaux de recherche sur les F/OSS [6, 9, 10, 11, 23, 25, 36], nous avons noté que chacun des projets Linux, Apache, Mozilla et GNOME est divisé en sous-projets qui sont à leur tour divisés en d'autres sous-projets pour faciliter leur développement. Également, des différences ont été constatées dans les projets concernant la philosophie de développement. Ces différences se retrouvent au niveau des processus, des rôles des participants, des outils utilisés dans leur infrastructure et dans la manière dont ces outils sont utilisés, ainsi que dans l'architecture de chacun de ces projets (voir le tableau 1 et le tableau 3).

Sous-projets de F/OSS	Pratiques de qualité	
	Particularités	**Similitudes**
Apache HTTP Server	• Les changements proposés sont votés à l'aide de la liste de diffusion *(mailing lists)*. 3 votes positifs et aucun vote négatif, sont requis pour soumettre un changement. • Actuellement les bogues sont rapportés en utilisant l'outil Bugzilla. • Apache a documenté chacune des versions du serveur. • Il est constitué de 6 sous-projets.	• Les bogues sont détectés et corrigés tant par les utilisateurs que par les développeurs. • Les projets sont menés par de nombreux participants répartis géographiquement à travers le monde et selon leur appartenance à un module ou à un sous-projet. • Les bogues et les problèmes de qualité peuvent être discutés sur les *mailing lists* des développeurs (cas du noyau Linux, ce sont les *mailing lists* des distributions du noyau), ou rapportés via un système de suivi de bogues, sur le portail web spécifique à chaque sous-projet (dans le cas du noyau Linux, c'est le portail web spécifique à une distribution Linux). • Le code source du projet est disponible et visible pour tous. • Chaque développeur choisit la tâche qu'il veut réaliser. Il peut s'agir des tâches de programmation, de conception, de documentation ou autres. • Chaque projet est divisé en de petits sous-projets afin de faciliter et d'améliorer son développement. • Chaque sous-projet de développement possède un petit groupe fermé de développeurs élus par les autres membres de la communauté, pour gérer, maintenir et prendre les décisions du sous-projet. • Chaque participant a le droit aux quatre libertés du libre. • Le « gel » du code est utilisé pour empêcher d'introduire des erreurs dans les versions stables du logiciel.
GCC (GNU Compiler Collection)	• Le projet requiert l'usage de standards de codage GNU ainsi que des standards de codage définis spécifiquement au projet GCC. • Utilise une suite de tests très détaillée construite autour de l'outil DejaGnu.	
Noyau Linux	• Plus de 50 listes de diffusion se concentrent sur le développement du noyau. • Il est constitué de 983 sous-projets. • Les bogues sont rapportés via Bugzilla à partir de la version 2.6. jusqu'aux versions actuelles 3.X.X • Le développement des versions est différencié par des branches et chaque version porte un numéro représenté sous le format x.y.z. • Intégration automatique du système.	
GNOME	• GNOME a documenté les guides de programmation. • Le système est testé et construit automatiquement grâce à Autoconf. • Les branches sont utilisées pour différencier les versions du logiciel. Chaque version porte un numéro représenté sous le format x.y.z	
Firefox	• Tests de portabilité forte, assistés par Tinderbox. • Builds automatiques via Tinderbox. • Les tests de régression sont effectués afin de compléter les tests unitaires.	

Tableau 1 : Résumé des pratiques des sous-projets libres retenus

Dans le concept du logiciel libre, les quatre libertés représentent les libertés auxquelles ont droit tout utilisateur d'un logiciel de type F/OSS, notamment la liberté de copier, d'exécuter, de modifier, de distribuer, d'étudier ou d'améliorer le logiciel. Elles ont été définies par Stallman [19] de la manière suivante :

- Liberté 0 : Liberté d'exécuter le programme pour tous les usages ;

- Liberté 1 : Liberté pour les utilisateurs d'étudier le programme et de l'adapter à leurs besoins. Cette liberté nécessite l'accès au code source ;
- Liberté 2 : Liberté de redistribuer les copies du programme ;
- Liberté 3 : Liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté. Cette liberté requiert également l'accès au code source.

Les critères de compatibilité développés à partir du tableau 1 sont :

- Critère 1 : les participants aux projets de logiciels libres doivent être répartis géographiquement à travers le monde [9].
- Critère 2 : les participants des projets de logiciels libres doivent être généralement des volontaires non payés [9].
- Critère 3 : les projets de logiciels F/OSS doivent être composés d'une communauté largement supérieure à 25. Les projets sur lesquels porte notre étude sont des projets à succès, possédant une très grande taille en ligne de code et dont les communautés vont d'un nombre moyen à très grand. Les projets Apache ou Mozilla sont supportés par des communautés de milliers de développeurs et de millions d'utilisateurs [61]. Apache est considéré dans le monde du libre comme un projet libre à large taille de code et à taille moyenne en nombre de participants [7]. Quant à Mozilla, il est un projet de taille moyenne de code et possédant un grand nombre de participants. Sa taille en nombre de participants est quasiment semblable à celle du noyau Linux [11]. Linux est considéré comme un projet libre à large taille de code et à très grand nombre de participants. Il est supporté par une grande communauté englobant développeurs et utilisateurs. Par exemple, il a été identifié que le noyau Linux possède une communauté de plus de 12 million d'utilisateurs et qu'elle croit de 40% par année [62] (en 2003) et il est supporté par des milliers de développeurs [62, 64] par version [63], tout en sachant qu'il y a toujours deux types de version, la stable et la non stable. Dans les projets F/OSS, plus de 25 personnes peuvent participer à la réalisation d'une activité du cycle de vie de logiciel. Cependant, la majorité des projets sont divisés en sous-projets qui constituent une petite communauté. Cet aspect est un avantage du développement des F/OSS, mais il constitue aussi la cause de la difficulté d'intégration de tous les modules du projet libre. Par ailleurs, il faut au minimum 15 développeurs de confiance pour maintenir un projet libre pour assurer sa survie [6]. D'autre part, la norme ISO/IEC 29110 est prescrite pour un TPO comportant au plus 25 personnes. Cependant, la norme souligne qu'elle n'est pas destinée à décourager ni à empêcher son utilisation dans des organisations de plus grandes tailles [12]. À cet effet, pour que cela puisse être réalisable dans les F/OSS, tout en sachant qu'un projet est souvent divisé en de plus petits projets, les gestionnaires de sous-projets de F/OSS doivent s'assurer qu'ils ne comportent pas plus de 25 personnes. Ainsi, le nombre de participants aux projets libres ne constituera plus un problème face à la norme ISO/IEC 29110.
- Critère 4 : les participants doivent eux-mêmes choisir, à partir de la liste des tâches à réaliser, les tâches qu'ils désirent réaliser selon leur expertise ou leur intérêt. Cette façon de procéder a été voulue en logiciel libre afin de préserver les libertés des participants, d'attirer le maximum de volontaires et d'augmenter les chances d'obtenir des contributions innovatrices et de qualité. À cet effet, certaines communautés de développement de logiciels F/OSS choisissent d'accepter uniquement des contributeurs ayant déjà eu à soumettre des contributions de qualité élevée. Par contre, d'autres communautés sont plus libérales dans leurs choix [7]. La présence de bons contributeurs fait parfois la différence dans le développement du projet.
- Critère 5 : le code source des logiciels libres doit être disponible pour tous les participants et pour tous les utilisateurs, compte tenu des quatre libertés [19].

Les deux premiers critères permettent de vérifier si le profil basique la norme ISO/IEC 29110 est compatible aux valeurs des logiciels libres. Le troisième critère permet de vérifier si la norme ISO/IEC 29110 peut être utilisée dans des projets libres comportant un grand nombre de participants. Quant aux critères 4 et 5, ils permettent de vérifier si la norme ISO/IEC 29110 est compatible au mode de développement de logiciels libres, principalement en ce qui concerne la transparence en termes de visibilité et d'accès des processus et des documents, ainsi que la disponibilité du code pour tous les membres du projet.

Les critères identifiés au sens du développement libre sont conformes à la culture (définition de Weigers [58] présentée en introduction) et à la philosophie de développement des logiciels F/OSS. De ce fait, ces critères sont fiables pour l'analyse de la compatibilité de la norme ISO/IEC 29110 avec le contexte particulier des logiciels F/OSS (voir figure 3), car ils respectent l'aspect social de leur développement.

L'aspect de compatibilité avec les F/OSS permettra d'introduire la façon dont la norme ISO/IEC 29110 pourrait soutenir le processus de développement des logiciels libres.

4. Présentation des résultats

Dans cette section, nous présentons les écarts du processus de développement libre avec la norme ainsi que les résultats de l'analyse de la compati-

bilité de la norme ISO/IEC 29110 avec les logiciels F/OSS

4.1 Identification des écarts du processus de développement libre

Les résultats présentés dans cette section sont essentiellement les écarts constatés entre le développement libre et le développement classique de logiciels.

Le modèle de développement des logiciels F/OSS offre de nombreux avantages. Premièrement, le code et la documentation sont disponibles pour tous les participants au projet, ainsi que pour tous les utilisateurs [19]. Cet aspect favorise une détection et une correction rapide des défauts du logiciel contribuant à atteindre un niveau de qualité élevée observé dans certains projets de logiciels F/OSS [29]. Les développeurs sont souvent eux-mêmes des utilisateurs du logiciel développé ou maintenu. Ils sont donc plus pointilleux dans leur travail, rendant ainsi le logiciel efficace et fiable [8, 30]. Cependant, la disponibilité du code est aussi une faille de sécurité du logiciel entraînant parfois des erreurs dans les versions et les mises à jour de sécurité [7, 9]. Un autre avantage du modèle de développement des logiciels F/OSS est l'usage de la modularité du code et de la réutilisation du code des composants [6, 11, 26], lesquelles sont identifiées comme des pratiques utilisées en industrie.

Le volontarisme des participants aux projets F/OSS, lesquels sont libres de choisir les tâches qu'ils désirent réaliser, contribue habituellement au manque de documentation des projets F/OSS. Il y a cependant des exceptions à la règle, par exemple, les projets Mozilla [11, 21] et GNOME [16], sont des projets prenant soin de documenter leur processus.

4.2 Résultats de compatibilité de la norme ISO/IEC 29110 avec les logiciels F/OSS

La majorité des activités du processus d'implémentation du profil basique de la norme ISO/IEC 29110 sont compatibles avec le mode de développement de logiciels libres. Ces activités sont : l'architecture et de conception, la construction logicielle, l'intégration et les tests, et enfin la livraison du produit. Cependant, les activités du démarrage de l'implémentation de logiciel ainsi que l'activité d'analyse des exigences de la norme de l'ISO ne sont pas compatibles avec le processus de développement des F/OSS.

Selon notre évaluation de la compatibilité de la norme ISO/IEC 29110 avec le modèle de développement des projets F/OSS, nous avons identifié que les critères de compatibilité numéros 4 et 5 (rappel de la section 3.2.1.) ne sont pas définis dans la description du processus du démarrage de l'implémentation du profil basique de la norme ISO/IEC 29110. Également, nous avons identifié que le critère de compatibilité avec les F/OSS numéro 4 n'est pas défini dans la description du processus d'analyse des exigences de la norme ISO/IEC 29110. Lors de notre évaluation, nous avons considéré non compatible toute activité du profil basique de la norme ISO/IEC 29110 : (1) comprenant au moins un critère de compatibilité avec les F/OSS, non défini dans les prescriptions de la norme, ou (2) qui soit conforme à au plus deux critères de compatibilité avec les F/OSS.

		Développement fermé de logiciels	Développement de logiciels F/OSS
Différences globales		• Le travail est assigné aux membres de l'équipe. • Les exigences sont définies par l'analyste. • La méthode de développement est définie et les processus sont structurés. • Les défauts sont majoritairement détectés et corrigés pendant la phase de tests et d'intégration. • La maintenance se fait à la demande des clients. • Le logiciel est intégré par les développeurs. • La composition de l'équipe de travail est effectuée par un chef.	• Les tâches sont choisies par les membres de l'équipe. • Les exigences sont définies par les développeurs. • Un processus de développement peu documenté ou souvent non défini. Le processus de développement se rapproche des méthodes agiles. • Une détection et une correction rapide des défauts avant la phase d'intégration et des tests. • La maintenance et l'évolution continue du logiciel par les développeurs. • Des revues par les pairs. • Le logiciel est intégré par les leaders techniques. • Chaque personne intéressée par le projet en cours de développement y adhère volontairement.
Particularités au niveau de chaque activité du cycle de vie du logiciel	Démarrage de l'implémentation	• L'existence d'un plan de projet	• Non existence du plan de projet. Le plan du projet prend forme tout au long de l'implémentation du logiciel.
	Analyse des exigences	• Les exigences sont définies par l'analyste. • La documentation utilisatrice du logiciel est définie par l'analyste. • Le client participe à la validation des exigences du logiciel. • Les exigences proviennent des besoins des clients. • Il y a des documents de résultats de vérification et de validation des exigences du logiciel.	• Les exigences sont définies par les programmeurs. • La documentation utilisatrice du logiciel si elle existe, est définie par le programmeur sous forme de page d'aide en ligne. • Les clients ne participent pas à la validation des exigences. • Les exigences prennent forme à travers des discussions *ad hoc* entre les développeurs. Elles sont implicites à l'implémentation et à la conception du logiciel. • Les documents de résultats de vérification et de validation des exigences du logiciel sont inexistants.
	Architecture et conception	• Le processus d'architecture et de conception de logiciels est défini et est très structuré. • L'analyste et le concepteur définissent ou mettent à jour la conception logicielle.	• Il n'y a pas de phase de conception logicielle explicite. Elle est implicite au développement. De même, l'architecture du logiciel prend forme au cours du cycle de vie du logiciel. Habituellement, la majorité des logiciels F/OSS procèdent à une traçabilité de suivi des problèmes. • Le développeur documente la conception du logiciel.
	Construction logicielle	• Un processus documenté.	• Il n'y a pas de différences. Néanmoins, en ce qui concerne la documentation du processus, c'est une tâche qui n'est pas obligatoire pour les participants au projet. Si cette documentation existe, elle est faite par les développeurs. Les artefacts peuvent être retrouvés à partir des documents et des informations se trouvant dans le dépôt de documents via le logiciel CVS.
	Intégration et tests	• Le logiciel est intégré par les développeurs. • Le processus est structuré et documenté	• Le leader technique, connu sous le terme de mainteneur de projet dans les logiciels libres, intègre le logiciel et définie les cas de tests. • Il y a des problèmes de documentation, car ce n'est pas une tâche obligatoire. Il n'existe pas de document de traçabilité et de rapport des tests.
	Livraison de produit	• La documentation de maintenance est documentée et vérifiée par le concepteur.	• La documentation de maintenance n'est pas souvent définie. Si elle existe, elle est documentée et vérifiée par le développeur et elle est disponible sous forme électronique.

Tableau 2 : Synthèse des différences entre les logiciels libres et le développement classique

Ce résultat ainsi que la connaissance des lacunes de qualité du développement libre, identifiées lors de l'analyse du processus de développement libre, a permis de proposer une amélioration au processus de développement des logiciels F/OSS à l'aide du profil basique de la norme ISO/IEC 29110.

5. Proposition d'amélioration de processus de développement des F/OSS

Suite à l'évaluation des processus de développement des logiciels F/OSS avec ceux du profil basique de la norme ISO/IEC 29110, de nombreux écarts ont été observés, tant au niveau des tâches, des rôles et des artefacts produits. Les activités du processus libre les plus problématiques sont les exigences, l'architecture et de conception, l'intégration et de tests. Notre analyse nous a permis d'identifier des écarts de documentation (les artefacts en sortie), des écarts dans la réalisation de certaines tâches telle que par exemple, l'assignation de travail aux membres, et des écarts de rôles. Selon

notre observation du résultat obtenu, le plus important sont les écarts de documentation, car un membre participant au développement libre peut réaliser plusieurs rôles lors de la réalisation d'une activité. Par ailleurs, bien que le mode de développement libre soit peu orthodoxe, que les participants soient libres de choisir les tâches sur lesquelles ils veulent travailler et que certains processus soient non définis au sens du génie logiciel, il est néanmoins possible de retrouver les différents processus d'implémentation de logiciels prescrit par la norme ISO/IEC 29110, ainsi que bon nombre de leurs tâches dans le processus de développement libre.

Nous avons ainsi identifié un manque accru de documentation dans les activités du processus du libre : l'analyse des exigences, l'architecture et de conception, et l'intégration et de tests. Habituellement, les communautés libres gardent majoritairement trace des défauts de code et moyennement trace du suivi des problèmes dans les exigences, la conception et la documentation [37]. Nous pensons que ce manque de documentation de ces activités pourrait provenir du bénévolat des participants au projet et du fait qu'ils soient libres de choisir les tâches qu'ils désirent réaliser. Les développeurs de logiciels F/OSS laissent souvent tomber plusieurs aspects de cette documentation. Une autre raison pourrait être l'absence d'une étape de vérification et de validation pour chacun des documents des logiciels F/OSS. Les outils de communication permettant le partage de contenus et documents les plus utilisés dans les F/OSS sont les listes de diffusions des développeurs et les IRC. Ces derniers sont les outils les plus privilégiés dans le développement libre, suivi des systèmes de suivi de bogues, car ils permettent d'archiver les communications échangées. Ainsi, il devient possible dans les projets libres de réaliser certaines documentations, par exemple, le document de vérification des exigences et de résultats de validation des exigences, ou encore un document de traçabilité, par simple suivi des discussions sur ces différents canaux de communication.

Étant donné que 4 des 6 activités du processus d'implémentation de la norme sont compatibles aux activités ou processus des F/OSS, il en ressort que la norme ISO/IEC 29110 serait compatible à 67% au développement de logiciels F/OSS.

Si elles sont mises en œuvre correctement, l'utilisation des normes du génie logiciel par les organisations permet, entre autres, d'améliorer la productivité du logiciel, de réduire le coût de développement et d'augmenter la qualité du logiciel développé ou maintenu. Notre hypothèse était que le suivi des activités compatibles de la norme ISO/IEC 29110 par les communautés des logiciels F/OSS, pourrait améliorer leur processus de développement. L'utilisation de la norme ISO/IEC 29110 dans les activités de développement du libre par l'approche proposée permettrait aux F/OSS d'avoir un développement plus organisé, lequel contribuerait à satisfaire aux nouvelles exigences de fiabilité et de durabilité identifiées par Weber [33]. Cet aspect pourrait inciter les organisations à utiliser de plus en plus les logiciels libres. De plus, cette approche d'application de la norme au processus de développement des logiciels F/OSS, permettra aux communautés de F/OSS d'améliorer la qualité de leur produit.

5.1 Application du profil basique de la norme ISO/IEC 29110 au développement de logiciels F/OSS

Une proposition pour l'amélioration, à court terme, des performances du processus de développement des logiciels libres serait d'appliquer initialement le profil basique la norme ISO/IEC 29110 uniquement aux activités d'architecture et conception, d'intégration et de tests des F/OSS. Une proposition à long terme serait d'appliquer la norme ISO/IEC 29110 aux activités que nous avons identifiées comme compatibles aux logiciels F/OSS.

La particularité de l'approche suggérée est qu'elle est axée sur l'aspect social du développement des logiciels F/OSS. Ainsi, elle se conforme aux valeurs, à la culture, et même à la philosophie des F/OSS. L'idée d'améliorer le développement libre n'est pas de rendre ce type de développement aussi rigoureux et contraignant que le développement classique mais d'apporter des ajustements tout en respectant et en conservant la liberté des participants des projets libres.

Il est proposé que le développement de logiciels F/OSS soit constitué en plusieurs sous-groupes de développeurs associés à des sous-projets dont le nombre serait de 25 ou moins. Les gestionnaires de projets libres devraient s'assurer que les sous-projets ne comportent pas plus de 25 développeurs. Ainsi, la prémisse de base de la norme ISO/IEC 29110 sera satisfaite. Également, nous avons choisi de supprimer la tâche consistant à assigner les tâches aux membres de l'équipe de travail selon leur rôle. Ce qui permettrait à notre approche d'être conforme aux libertés des utilisateurs et participants au projet de logiciels F/OSS.

Considérant le modèle de développement des logiciels F/OSS et le fait que les participants aux F/OSS veulent conserver leur liberté, de même que l'accessibilité et la transparence de tous les documents et de toutes les phases du processus de développement libre, les aspects de cette approche d'application de la norme aux F/OSS sont résumés par les artefacts, les rôles et les tâches pour les 4 activités de la norme ISO/IEC 29110 compatibles aux F/OSS. La description de ces aspects est la suivante :

5.1.1 Les artefacts et les rôles

L'objectif à atteindre est de permettre à un projet F/OSS de réaliser un produit logiciel de qualité élevée. Le pré-requis essentiel est que le nombre de participants au projet soit au minimum de 15 [6]. Cependant, il est suggéré de diviser le projet en sous-projets ne comportant au plus que 25 personnes. La division en sous-projets est déjà une pratique courante dans les F/OSS. Il est aussi possible que les personnes d'un sous-groupe jouent plus d'un rôle.

Par ailleurs, tel qu'identifié antérieurement, les F/OSS ont de nombreux problèmes de documentation. Ainsi, il est proposé comme solution à ce problème que, dans chaque processus de développement libre, particulièrement les activités de la norme ISO/IEC 29110 compatibles aux F/OSS, que des listes de contrôles de documents ou des artefacts soient appliquées dans les processus libres pour combler les lacunes observées.

Dans la suite de cette section, compte tenu de notre observation du développement libre et des prescriptions faites par la norme ISO/IEC 29110, nous ferons des suggestions ou recommandations d'améliorations du processus de développement libre. Particulièrement concernant des artefacts devant se retrouver dans le développement libre et que nous avons retenus pour les processus libres pour lesquels la norme est compatible. Pour chacun des artefacts retenus, nous discuterons de leur usage et de leur pertinence dans développement libre.

5.1.2 Activité d'architecture et de conception

Sachant que les F/OSS sont majoritairement développés, maintenus et gérés par des bénévoles répartis géographiquement à travers le monde [9], structurer cette activité sera très contraignant pour les participants aux F/OSS, car elle va à l'encontre de la liberté voulue dans le déroulement des processus libres. De plus, sachant que le développement des F/OSS est non seulement itératif [9, 10], incrémental [31], mais également parallèle [7, 11]; que les exigences du logiciel dans les F/OSS émanent généralement des discussions entre développeurs [32] et non pas des besoins d'un client ; il devient donc difficile d'envisager une conception globale ou détaillée du logiciel, ni même de définir une architecture du système, sans que toutes les exigences n'aient été définies correctement et complètement.

- *Artefacts :* au sens des F/OSS, les artefacts retenus sont : *le document de spécification des exigences, le document de conception logicielle, le rapport de traçabilité et les cas et les procédures de tests.* Exiger des développeurs de logiciels F/OSS la réalisation d'un document de conception semble inutile à premier abord, car ils produisent déjà du logiciel de qualité. Par ailleurs, la présence d'un tel document dans le développement libre, pourrait permettre aux F/OSS non seulement de mieux planifier et de mieux structurer leur développement, mais également de permettre d'accroître leur communauté en attirant de plus en plus de participants compte tenu de la disponibilité des ressources libres ainsi que de la transparence des processus libres. Nous recommandons qu'un des membres du groupe de développeurs des F/OSS possédant une bonne connaissance de la conception courante, réalise et documente la conception logicielle ; ceci à l'opposé de ce qui se fait habituellement dans les F/OSS. Afin de vérifier le document de conception, il peut inviter les pairs, de la liste de diffusion de développeurs, à une revue. Ainsi, les autres développeurs pourraient influencer les choix d'architecture et améliorer la qualité de la conception. Quant au rapport de traçabilité, la norme ISO/IEC 29110 prescrit qu'il devra incorporer les relations entre les exigences et les éléments de conception logicielle, les cas et les procédures de tests. À titre d'exemple, le projet Mozilla, héritant de l'architecture et de la documentation de Netscape 5 [11, 46, 47], possède déjà un document de cas de tests (rapport de test) [21], un document de conception logicielle, un document de traçabilité, un document de requête de changements [11].
- *Rôles :* La norme de l'ISO décrit, entre autres, les trois rôles suivants : le « lead » technique, l'analyste et le concepteur. Dans les F/OSS, compte tenu de notre évaluation de leur processus, les rôles de « lead » technique et de développeur seront retenus. Cependant, comme indiqué précédemment les développeurs jouent souvent les rôles d'analyste et de concepteur.

5.1.3 Activité de construction logicielle

La construction est une activité importante du développement des F/OSS [37]. La mise en œuvre de cette activité du profil basique de la norme ISO/IEC 29110 devrait permettre aux F/OSS d'améliorer la documentation du code, ainsi que de définir et/ou documenter le rapport de traçabilité. En ce qui concerne la documentation de la configuration du logiciel, elle est déjà présente en ligne pour les projets libres étudiés.

Artefacts : la recommandation porte sur les artefacts suivants :

- *Standards d'écriture de code* : cette documentation permettrait, pour chaque communauté de F/OSS, d'avoir une unique façon d'écrire le code. Ce qui faciliterait le contrôle du code source et permettrait de retenir plus rapidement des contributions importantes. Par exemple, quelques fois dans les projets F/OSS, des codes sources contrôlés par un développeur chef peuvent être rejetés faute d'un non respect du style de codage, car ce style n'est pas explicitement documenté quelque part sur le portail web de la communauté. Hors,

il se pourrait que l'un des codes sources rejetés puisse résoudre de manière efficace un problème soulevé dans le projet, et que la solution proposée soit d'un niveau élevé de qualité. Il serait ainsi dommage de passer à côté de telles contributions. Tel qu'identifié par Michlmayr [7], certains F/OSS à grand nombre de contributeurs définissent la documentation du style de codage et la documentation de soumission de code. La documentation de soumission de code décrit quand et qui peut faire des changements dans le système de contrôle de versions d'un projet libre [7]. Il est recommandé que les F/OSS se dotent de documentation comme bonne pratique de développement.

- *Cas de tests unitaires :* dans les F/OSS, chaque développeur réalisant une fonctionnalité du logiciel, effectue déjà des tests unitaires. C'est une pratique répandue dans les projets libres.
- *Rapport de traçabilité :* ce rapport devrait permettre pour les logiciels F/OSS de facilement repérer les composants logiciels qui ont été construits et de faciliter le développement et l'exécution des tests et la maintenance.
- *Rôles :* La norme décrit, entre autres, deux rôles qui sont : le « lead » technique et le programmeur. Ces rôles sont utilisés dans les logiciels F/OSS d'après notre évaluation de leur processus de développement.

5.1.4 Activité d'intégration et de tests

Le développement de logiciels F/OSS a un avantage par rapport au développement de logiciels développés en entreprise : les défauts sont détectés et corrigés par de nombreuses personnes (combinaison de développeurs et d'utilisateurs) bien avant la phase d'intégration du logiciel. Cependant, la mise en œuvre de la norme ISO/IEC 29110, selon notre proposition, permettrait aux F/OSS d'améliorer d'avantage la qualité du processus, tout en conservant aussi bien la liberté des utilisateurs que la transparence des documents et du processus.

- *Artefacts :* l'application portera sur le rapport de tests et le rapport de traçabilité dans les F/OSS. Tel que mentionné, ces documents n'existent habituellement pas dans les projets de F/OSS. Cependant, ils peuvent être reconstruits à partir des archives d'un logiciel de gestion de version, tel le CVS, sur les composants du logiciel pour le rapport de traçabilité, et à partir des discussions de traitement de erreurs se trouvant sur les listes de diffusion des développeurs et les systèmes de suivis de bogues pour le rapport de tests. Une description d'une erreur dans les F/OSS peut être partagée comme le code entre les développeurs et elle comporte certaines caractéristiques, entre autres : un numéro d'identification, le nom de l'auteur/propriétaire, les fichiers liés, la gravité et priorité de l'erreur, le statut et enfin le nombre de commentaires [11]. Quant à l'outil de gestion de version, les archives contiennent habituellement des informations sur le nom, la version, le mainteneur, la licence, la taille et autres informations sur le package ou composant logiciel, avec le plus important étant que, chaque package contient des références7 pour d'autres packages nécessaires à l'exécution [38].
- *Rapport de tests :* Il documente les résultats des tests faits pour l'intégration du logiciel.
- *Rapport de traçabilité :* le document de traçabilité sera créé s'il n'existe pas. Dans le cas contraire, il sera mis à jour par l'ajout de l'intégration des différentes relations entre les composants logiciels ayant été construits ou modifiés.

Bien que les défauts du logiciel dans les F/OSS soient détectés et corrigés par de nombreux individus [8] bien avant la phase d'intégration [6], la mise en œuvre du profil basique de la norme ISO/IEC 29110 par l'approche proposée, permettra aux F/OSS d'améliorer la qualité de ce processus libre tout en conservant tant les libertés des utilisateurs que la transparence des documents et celle du processus. Ainsi, chaque sous-groupe d'un module sera constitué d'au plus 25 développeurs. Ils coordonneront et intégreront toutes les contributions des développeurs externes à leur module [10, 11, 23, 48]. De plus, nous recommandons que ce groupe rédige le rapport de tests et le rapport de traçabilité pour l'intégration de leur composant. Le groupe pourra procéder par vote entre ses membres pour désigner la personne responsable de cette tâche. Une fois rédigés, ces documents seront soumis aux autres développeurs de la communauté (via les listes de diffusion des développeurs) pour révision. Ainsi, toute personne voulant participer à ce processus peut soumettre des suggestions pour la vérification de ces artefacts ou en réalisant les tests. Ainsi, cette recommandation permet de conserver non seulement les bonnes pratiques du développement libre, mais également de les améliorer par l'ajout des documents de traçabilité et du rapport de tests d'intégration.

- *Rôles :* La norme décrit, entre autres, quatre rôles qui sont : le « lead » technique, le programmeur, le client et l'analyste. Selon les résultats de notre évaluation, les développeurs de logiciels libres portent aussi le chapeau d'analyste et souvent celui de client. Les clients étant les utilisateurs du logiciel libre. Par utilisateurs, nous entendons, des individus, des programmeurs ou des organisations.

5.1.5 Activité de livraison de produit

Cette activité consiste à fournir un logiciel intégré ainsi que toute la documentation nécessaire à l'installation, à la compréhension et à l'utilisation du logiciel. Les projets F/OSS peuvent aussi livrer le logiciel sous forme de version empaquetée. Une version empaquetée du logiciel est une version emballée pour la vente au détail. En général dans les projets F/OSS, le format zip ou le gzip sont utilisés

à cet effet. La différence entre la livraison du produit logiciel dans les F/OSS et celle en entreprise se fait ressentir au niveau des documents qui sont inclus avec le logiciel. En entreprise, les logiciels sont livrés avec tous les documents relatifs à l'installation, à la configuration, à l'utilisation. Par contre, selon notre observation des projets F/OSS, la livraison du produit consiste à fournir une version empaquetée du logiciel (connu sous le terme anglais Package) et sans documentation. Concernant les logiciels F/OSS documentés, la documentation est habituellement présente sous forme de pages d'aide en ligne. Dans certains projets F/OSS, il est possible de retrouver en ligne, la documentation sur l'installation et l'utilisation du logiciel fourni.

- *Artefacts :* les différents artefacts retenus pour la personnalisation sont la documentation de maintenance et la configuration logicielle.
- *Rôles* : La norme décrit, entre autres, trois rôles qui sont : le « lead » technique, l'équipe de travail et le concepteur. Pour appliquer la norme dans les développements F/OSS, les rôles suivants sont retenus : le « lead » technique et le développeur (ce dernier jouera également, pour le processus de livraison de produit, le rôle de concepteur).

5.1.6 Les tâches de livraison du produit :
La tâche « l'affectation des tâches aux membres de l'équipe de travail selon leur rôle » a été supprimée afin de respecter le mode développement des F/OSS. Les autres tâches décrites dans le profil basique la norme ISO/IEC 29110 peuvent être suivies par le processus de développement des F/OSS.

Considérant l'aspect de la structure informelle des F/OSS et de leur mode de développement à la fois ouvert, asynchrone et évolutif [9], les tâches de documentation sont plutôt difficiles à respecter. Plusieurs projets libres connaissent des problèmes de documentation identifiés en 2005 par Michlmayr, Hunt et Probert [9]. Cependant, il est possible de retrouver dans d'autres projets libres de la documentation pour le support des utilisateurs finaux ou des développeurs [11, 36]. Ainsi, face au problème de documentation du processus des F/OSS, il est recommandé que les responsables de sous-projets se chargent de pallier ces lacunes. En plus des tâches déjà identifiées pour les responsables de sous-projets dans les F/OSS [23], pour que la documentation soit réalisée, nous proposons :

- Que les membres du groupe de mainteneurs de projets, pour un module donné, élisent une personne responsable de la documentation. Exception faite de Mozilla pour qui chaque auteur d'un module du projet est également le propriétaire du module et représente l'autorité de prise de décision concernant ce module [11]. Dans ce cas, en plus des tâches qui lui sont allouées, l'auteur devra définir ou mettre à jour la documentation relative à son module;
- Qu'une fois la documentation faite, que celle-ci soit transmise aux autres membres du sous-projet pour révision. Dans le cas échéant, l'un des membres du groupe de mainteneurs, désigné par ses confrères, sera chargé de vérifier que la documentation relative à un processus donné soit définie.

Cette façon de procéder sera conforme aux pratiques des F/OSS et n'entraverait pas la liberté des développeurs concernant leur choix des tâches. Par ailleurs, l'approche suggérée permet de s'assurer que la documentation soit faite.

6. Conclusion

Bien qu'elle soit une étude analytique sur l'application du profil basique de la norme ISO/IEC 29110 au processus de développement des logiciels F/OSS, cette étude propose que les projets F/OSS pourraient profiter de cette norme pour raffiner leur processus et améliorer la qualité du produit développé. Suite à la définition de critères de compatibilité avec les F/OSS, l'analyse suggérée de la compatibilité de la norme ISO/IEC 29110 avec les F/OSS est avantageuse, car elle respecte les valeurs, la culture et la philosophie de développement des F/OSS. De ce fait, elle représente une approche intéressante au contexte particulier des F/OSS. Elle a permis de démontrer que le profil basique de la norme ISO/IEC 29110 peut soutenir le développement libre, car 4 des 6 activités d'implémentation sont compatibles aux valeurs, à la culture, de même qu'à la philosophie de développement des F/OSS. Une proposition d'amélioration de la performance du processus de développement libre a été suggérée. Les améliorations proposées se concentrent principalement au niveau des activités d'architecture et de conception, d'intégration et de tests ainsi que celles relatives à la livraison du produit.

La mise en place des suggestions permettront au processus de développement libre de se raffiner en termes de définition et/ou d'ajout de documentation au niveau des phases de développement telles que : l'architecture et conception, la construction du logiciel, l'intégration et les tests, et la livraison du produit. Nous recommandons à cet effet que les développeurs de confiance qui sont les *leads* ou les développeurs chefs, les experts en leur domaine, communément connus sous le terme de mainteneurs de projets dans les F/OSS appliquent une liste de contrôle de documents et effectuent eux-mêmes les tâches de documentation afin d'assurer la réalisation de la documentation souvent absente dans le processus de développement libre. Nous sommes conscients que certaines de nos suggestions pourraient être reçues avec une certaine réticence par les communautés libres. Ainsi, comme travail futur, nous pourrions effectuer une phase de validation des nos suggestions d'améliorations

du processus de développement libre en effectuant un enquête auprès des participants à certains projets F/OSS.

La mise en œuvre du profil basique de la norme ISO/IEC 29110 dans les processus libres tel que proposé, permettra aux développeurs de logiciels F/OSS d'avoir un développement plus organisé, plus formel, lequel contribuera à satisfaire aux nouvelles exigences de fiabilité et de durabilité identifiées par Weber [33]. De plus, ceci permettra aux communautés des F/OSS de maintenir ou d'améliorer la qualité, déjà présente dans certains produits de F/OSS.

Ce travail peut constituer une base de recherche pour l'évaluation de la compatibilité des normes sur le développement du génie logiciel par rapport au modèle de développement libre par le biais des critères de compatibilité avec le développement de logiciels F/OSS décrits dans cette étude.

NOTES

1 Son acronyme en français est ISO/CEI 29110. ISO signifie : Organisation Internationale de Normalisation et CEI signifie : Commission Électrotechnique Internationale.

2 Les valeurs sont les règles qui régissent le comportement d'un ensemble de personnes. Elles sont implicites à la culture de chaque organisation. Pour plus de détails, voir la définition de Wikipédia : *http://fr.wikipedia.org/wiki/Valeur_%28personnelle_et_culturelle%29*.

3 Un principe est : « une proposition fondamentale de la discipline formulée sous forme prescriptive (règle), à la source des actions, pouvant être vérifiée dans ses conséquences et par l'expérience » [59].

4 C'est un processus dans le développement libre qui consiste à donner du mérite aux participants ayant soumis des contributions importantes. Plus des participants aux projets de F/OSS fournissent des contributions signifiantes, plus ils gagnent des responsabilités au sein de la communauté. Pour les nouveaux adhérents au projet, ils obtiennent le statut actif, et peuvent ainsi continuer de grimper en échelon, et gagnent un accès direct au dépôt de code.

5 Dans le cadre de l'étude, le terme « processus » est souvent utilisé à la place du terme « activité » pour désigner une phase du processus de développement de logiciel.

6 Le développement fermé représente le développement utilisé dans les entreprises commerciales. Il est très souvent procédural et le logiciel est sous des licences protégeant la propriété intellectuelle.

7 Les références dont il est question consistent en des informations de dépendances portant les statuts suivants : « requis », « recommandé » ou « suggéré » [38].

7. RÉFÉRENCES

[1] E. S. Raymond : *Linux and open-source success* ; IEEE Software, vol. 16, no 1, 1999, p. 85–89, cité par T. Otte, R. Moreton et al., « Applied Quality Assurance Methods under the Open Source Development Model », in the 32nd Annual IEEE International Conference on Computer Software and Application, no 0730-3157, 2008, p. 1247-1252, Turku.

[2] T. J. Halloran et W. L. Scherlis : *High quality and open source software practices* ; in Proceedings of the 2nd Workshop on Open Source Software Engineering, Orlando, FL, États-Unis, ICSE, 2002, États-Unis.

[3] D. C. Schmidt et A. Porter : *Leveraging open-source communities to improve the quality & performance of open-source software* ; in Proceedings of the 1st Workshop on Open Source Software Engineering, 2001, Toronto, Canada.

[4] Anas Tawileh et Omer Rana : *Free and Open Source Software Quality Assurance* ; IEEE, 2006

[5] Martin Michlmayr : *Quality improvement in volunteers free software projects: Exploring the impact of release management* ; in Proceedings of the First International Conference on Open Source Systems, juillet 2005, p. 309-310, Genève, Suisse.

[6] Mark Aberdour : *Achieving quality in open source software ;* IEEE Computer Society, IEEE Software, no 0740-7459, 2007, États-Unis.

[7] Martin Michlmayr : *Quality improvement in volunteer free and open source software projects : Exploring the impact of release management ;* Centre de la gestion de technologie, A dissertation submitted to the University of Cambridge for the Degree of Doctor of Philosophy, 2007, Angleterre, *http://www.cyrius.com/publications/michlmayr-phd.pdf*, visité le 5 octobre 2011.

[8] E. S. Raymond : *The Cathedral and the Bazaar ;* Sebastopol, CA: O'Reilly & Associates, 1999, Canada, *http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/*, visité le 5 octobre 2011.

[9] Martin Michlmayr et al. : *Quality practices and problems in free software projects* ; in Proceedings of the First International Conference on Open Source Systems, juillet 2005, p. 24-28, Genève, Suisse.

[10] Audris Mockus, Roy T. Fielding et al. : *Two case studies of open source software development: Apache and Mozilla ;* ACM Transactions on Software Engineering and Methodology, vol. 11, no 3, 2002, p. 309-346, États-Unis.

[11] Christian R. Reis et Renata P.d.M. Fortes : *An overview of the software engineering pro-*

cess and tools in the Mozilla project : CiteSeerX, 2002, Brésil, *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.6127*, visité le 22 octobre 2011.

[12] ISO/IEC TR 29110-5-1-2:2011- Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs) - Part 5-1-2: Management and engineering guide - Generic profile group: Basic profile. Organisation internationale de normalisation/Commission électrotechnique internationale: Genève, Suisse. Disponible gratuitement sur l'ISO: h*ttp://standards.iso.org/ittf/PubliclyAvailableStandards/c051153_ISO_IEC_TR_29110-5-1_2011.zip*

[13] Apache HTTP Server, *http://httpd.apache.org/, http://www.apache.org/dev/infrastructure.html*, visité le 5 octobre 2011.

[14] Mozilla Firefox, *http://www.mozilla.com/fr/firefox/*, visité le 5 octobre 2011.

[15] Noyau Linux, *http://www.kernel.org/*, visité le 5 octobre 2011.

[16] GNOME, *http://www.gnome.org/, http://directory.fsf.org/project/gnome/*, visité le 5 octobre 2011.

[17] GCC, *http://gcc.gnu.org/, http://directory.fsf.org/project/gcc/*, visité le 5 octobre 2011.

[18] Debian GNU/Linux, *www.debian.org*, visité le 5 octobre 2011.

[19] Projet GNU, *http://www.gnu.org/philosophy/free-sw.fr.html*, visité le 17 septembre 2011.

[20] Apache, *www.apache.org*, visité le 5 octobre 2011.

[21] Mozilla, *www.mozilla.org*, visité le 5 octobre 2011.

[22] Roy T. Fielding : *Shared leadership in the Apache project ;* Communications of ACM, vol. 42, no 4,1999, p. 42-43, États-Unis.

[23] Josh Lerner et Jean Triole : *The simple economics of open source ;* National Bureau of Economic Research: Cambridge, no 7600, 2000, MA, *http://papers.nber.org/*, visité le 5 octobre 2011.

[24] Justin R. Erenkrantz : *Release management within open source projects* ; 3rd Workshop on Open Source Software Engineering, 2003, États-Unis.

[25] Christian R. Reis et Renata P.d.M. Fortes : *An overview of the software engineering process and tools in the Mozilla project* ; CiteSeerX, 2002, Brésil, *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.6127*, visité le 22 janvier 2011.

[26] Daniel M. German : *The GNOME Project: A case study of open source global software development ;* Software Process Improvement and Practice, vol. 8, no 8, 2003, p. 201-215, Canada.

[27] Roy T. Fieldind et G. Kaiser : *The Apache HTTP Server Project ;* IEEE Internet Computing, vol. 1, no 4, 1997, p. 88 -90.

[28] Apache Software Foundation Bugs System: *https://issues.apache.org/bugzilla/*, visité le 5 octobre 2011.

[29] Michel Dumais : *Avantages et contraintes du logiciel libre* ; logiquelibre.com, avril 2007, *http://www.logiquelibre.com/modules/documentation/item.php?itemid=3*, visité le 5 octobre 2011.

[30] Ioannis Samoladas et Ioannis Stamelos : *Assessing Free/Open Source Software Quality* ; Computer and Information Science, Aristotle University of Informatics, Grèce nd, 2003, p. 1-36, États-Unis, *http://kb.cospa-project.org/retrieve/2768/samoladasstamelos.pdf* , visité le 5 octobre 2011.

[31] Niels Jørgensen : *Putting it all in the trunk: Incremental software engineering in the FreeBSD open source project* ; Information Systems Journal, vol. 11, no 4, 2001, p. 321–336.

[32] Walt Scacchi : *Free and Open Source Development Practices in Game Community* ; IEEE Computer Society, no 0740-7459, 2004, p. 59-66, États-Unis.

[33] S. Weber (2005) : *The Success of Open Source ;* Cambridge MA: Harvard University Press, no 10.10170S0008423907070266, 2004, publié en ligne dans le Canadian Journal of Political Science, vol. 40, no 1, 2007.

[34] Stefan Koch et Georg Schneider : R*esults from software engineering research into open source development projects using public data ;* Diskussionspapiere zum 166 Tätigkeitsfeld Informationsverarbeitung und Informationswirtschaft, 22. Institut für Informationsverarbeitung und Informationswirtschaft, WU Vienna University of Economics and Business, Vienne, Autriche, 2000, *http://epub.wu.ac.at/494/*, visité le 10 octobre 2011.

[35] Brian Behlendorf, Scott Bradner et al. : *Tribune Libre : Ténors de l'Informatique Libre »*, O'Reilly & Associates, Inc., Adaptation française du livre intitulé « Open Sources: Voices from the Open Source Revolution », 1999, États-Unis, *http://www.velic.com/publications/tribunelibre/index.html*, visité le 5 octobre 2011.

[36] Walt Sacchi : *Understanding the requirements for developing open source software systems* ; IEEE Proceedings-Software, vol. 149, no 1, 2002, p. 24-39, États-Unis.

[37] T. Otte, R. Moreton, et al. : *Applied quality assurance methods under the open source development model* ; 32nd Annual IEEE International Conference on Computer Software and Application, no 0730-3157, 2008, p. 1247-1252, Turku.

[38] Sebastian Spaeth, Matthias Stuermer et al. : *Sampling in open source software development: the case for using the Debian*

GNU/Linux Distribution ; Proceedings of the 40th Annual Hawaii International Conference on System Sciences, IEEE Computer Society, no 0-7695-2755-8, 2007.

[39] Ulf Asklund et Lars Bendix : *A study of configuration management in the open source software projects ;* IEEE Proceedings Software, vol. 149, no 1, 2002, p. 40-46.

[40] Eclipse, *www.eclipse.org*, visité le 5 octobre 2011.

[41] Dépôt de documents GNOME, *http://live.gnome.org/Git*, visité le 5 octobre 2011.

[42] Apache Software Foundation Bugs System: *https://issues.apache.org/bugzilla/*, visité le 20 janvier 2011.

[43] Test automatique : *http://quality.mozilla.org/teams/automation/*, visité le 20 janvier 2011.

[44] Listes de liste de diffusion Mozilla : *https://lists.mozilla.org/listinfo*, visité le 20 janvier 2011.

[45] Listes de liste de diffusion Eclipse : *https://dev.eclipse.org/mailman/listinfo*, visité le 20 janvier 2011.

[46] Jacques Longchamp : *Open source software development process modeling* ; Springer-Link, The Kluwer International Series in Software Engineering, vol. 10, no 0-387-24262-7_2, 2005, p. 29-64, France.

[47] Chris Llias : *Understanding the relationship and history between Mozilla and Netscape* ; Mozilla, *http://ilias.ca/MozillaNetscape Relationship*, visité le 23 mars 2011.

[48] Daniel M. German : *The evolution of the GNOME Project ;* Proceedings of the 2nd Workshop on Open Source Software Engineering, 2002, Boston, MA, États-Unis.

[49] ISO/IEC 24765:2010 : *Systems and Software Engineering Vocabulary* ; Organisation internationale de normalisation/Commission électrotechnique internationale, Genève, 2010, en ligne : www.computer.org/sevocab

[50] Kevin Crowston, Hala Annabi, James Howison et al. : *Effective Work Practices for Software Engineering:* Free/Libre Open Source Software Development ; ACM, in proceeding of WISER, NFS Grants, no 03-41475 et 04-14468, 2004, États-Unis.

[51] Peter C. Rigby, Daniel M. German et Magaret-Anne Storey : *Open source software peer review practices: A case study of the Apache server ;* ACM, no 978-1-60558-079, 2008, Allemagne.

[52] C. Y. Laporte et E. Palza Vargas : *The development of international standards to facilitate process improvements for very small enterprises* ; Cité dans « Software Process Improvement and Management: Approaches and Tools for Practical Development », IGI Global Publisher, 2012, p. 34-61, États-Unis. Disponible à: *http://profs.etsmtl.ca/claporte/ Publications/Publications/IGI_Chapter_SPI_in_VSEs.pdf*

[53] Site internet sur la norme ISO/IEC 29110: *http://profs.etsmtl.ca/claporte/VSE/Groupe24-menu.html*

[54] *« The Open Source Definition »*, Open Source Initiative, *http://www.opensource.org/docs/osd*, visité le 10 août 2010.

[55] La Free Software Fondation, *www.fsf.org*, visité le 14 août 2010.

[56] « Apache Karaf 2nd Meeting (2012-01-04) », Apache, *https://cwiki.apache.org/KARAF/karaf-2nd-meeting-2012-01-04.html*, visité le 4 février 2012

[57] Mohamed F. Ahmed et Swapna S. Gokhale : Linux bugs: *Life cycle and resolution analysis ;* IEEE Computer Society, no 150-6002, 2008, États-Unis.

[58] Karl Weigers : Creating a software engineering culture ; Dorset House Publishing, 1996.

[59] Normand Séguin : Inventaire, analyse et consolidation des principes fondamentaux du génie logiciel ; École de technologies supérieure, no 9780494208748, 2006, Montréal, CA, *http://proquest.umi.com/pqdweb?did=1251894611& sid=13& Fmt=2& clientId=46962& RQT=309& VName=PQD*, visité le 22 janvier 2011.

[60] Philosophie, Larousse, *http://www.larousse.fr/dictionnaires/francais/philosophie*

[61] C. Gacek et B. Arief : *The many meanings of Open Source* ; IEEE Software 21, p. 34–40, 2004 cité par Flore Barcillini, Francoise Détienne et Jean-Marie Burkhardt, « User and developer mediation in an Open Source Software community: Boundary spanning through cross participation in online discussions », International Journal of Human-Computer Studies, vol. 66, no 7, 2008, p. 558 – 570

[62] Gwendolyn K. Lee et Robert E. Cole : *From a Firm-Based to a Community-Based Model of Knowledge Creation: The Case of the Linux Kernel Development ;* Organization Science, vol. 14, no 6, 2003, p. 633-649

[63] Baruch Siach : Linux Kernel Participation How to» ; 2011, *http://www.slideshare.net/benavrhm/linux-kernel-participation-howto*, visité le 16 mars 2012

[64] Guido Hertel, Sven Niedner et Stefanie Herrmann : *Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel* ; Open Source Software Development, vol. 32, no 7, 2003, p. 1159–1177

[65] T. Varkoi : *Process assessment in very small entities* ; 7th International Conference on the Quality of Information and Communications Technology, Oporto, Portugal, 29 septembre – 2 octobre, 2010.

[66] C. Y. Laporte : *Contributions to software engineering and the development and deployment of international software engineering standards for very small entities* : PhD thesis

of the Université de Bretagne Occidentale, Brest (2009), *http://tel.archives-ouvertes.fr/tel-00483255/fr/*

[67] C. Y. Laporte, S. Alexandre et R. O'Connor : *A software engineering lifecycle standard for very small enterprises* ; in: R. O'Connor et al. (Coord.) Proceedings of EuroSPI. CCIS, vol. 16, pp. 129–141. Springer, Heidelberg (2008)