

# LA PROFESSIONNALISATION DES ETUDIANTS EN INFORMATIQUE PAR LES PROJETS LIBRES

Morgan Magnin<sup>1</sup>, Guillaume Moreau<sup>2</sup>

<sup>1</sup> *École Centrale de Nantes, IRCCyN, Nantes, France*

<sup>2</sup> *École Centrale de Nantes, CERMA, Nantes, France*

eat-tice@ec-nantes.fr

## Résumé

L'apprentissage par projets permet de développer les compétences techniques et humaines des apprenants. L'originalité de notre contribution réside dans le lien mis en place, dans le cadre de l'enseignement de l'informatique à l'École Centrale de Nantes, entre la communauté des logiciels libres et les apprenants. Nous avons mis en place une dynamique de collaboration des étudiants à des logiciels libres, dans le cadre de projets réalisés en autonomie. Le travail des élèves bénéficie ainsi non seulement à eux-mêmes (par les connaissances et compétences acquises, utiles pour leur insertion professionnelle), mais également à la communauté – au sens large – d'usage de ces programmes. Ce dispositif a créé un cercle vertueux entre utilisateurs et contributeurs des logiciels libres.

## Mots-clés

Projets, compétences, pratiques pédagogiques, étudiants, enseignants.

## I. INTRODUCTION

L'un des objectifs des formations professionnalisantes telles que le cursus ingénieur dispensé à l'École Centrale de Nantes est de préparer les étudiants à l'emploi par l'acquisition de compétences tant scientifiques et technologiques qu'humaines (capacité à s'intégrer, communiquer, partager). Il importe donc de favoriser les contacts directs des étudiants avec le milieu professionnel durant leurs années d'étude. Cette action prend évidemment place dans le cadre des stages. Mais elle peut également se décliner à l'occasion d'études de cas et de projets partagés avec des professionnels, pour une prise de conscience de l'intérêt de l'ensemble de la démarche projet enseignée en cours, mais qui ne peut pas être justifiée à l'échelle d'un TP, même long. La difficulté est alors d'identifier des sujets qui ne soient ni anecdotiques, ni critiques (en termes de fonctionnalités ou de confidentialité). De plus, le travail accompli par les étudiants est généralement cédé gracieusement au partenaire extérieur, alors qu'il aurait pu être utile à une plus large communauté.

L'informatique a ceci de particulier qu'elle se développe tout autant selon une philosophie dite « propriétaire » (l'utilisation, la redistribution ou la modification d'un logiciel propriétaire sont interdites) que « libre » (les libertés d'utilisation, d'accès et de modification du code sont garanties [GNU, 1996]). Le choix entre les deux repose généralement sur l'histoire et le modèle économique du logiciel. L'approche libre a l'avantage de faire profiter tous les utilisateurs des améliorations portées par chacun. De plus, il est aujourd'hui acquis que les cycles de développement des grands logiciels libres sont similaires à ceux des logiciels propriétaires. C'est pourquoi nous avons souhaité évaluer l'intérêt et la pertinence de projets étudiants menés en lien avec des communautés de logiciels libre. Nous avons mis en œuvre un cadre favorable à la succession de plusieurs projets, permettant ainsi des améliorations itératives du code produit par les étudiants. Nous montrerons notamment comment, en s'insérant dans ces cycles de développement, nos étudiants ont eu la possibilité de participer à des projet d'envergure (ce qu'ils ne peuvent pas faire avec des logiciels « propriétaires » faute de disposer des sources) et de dépasser le seuil des travaux pratiques classiques qui limitent la portée de l'apprentissage à une dimension technique, avec une pérennité observée, au-delà de nos objectifs.

Dans la suite de cet article, nous aborderons d'abord les origines de ce dispositif pédagogique, nous décrirons ensuite comment nous le mettons en œuvre dans l'option informatique de l'Ecole Centrale de Nantes. Nous proposerons quelques recommandations pour cette mise en œuvre avant de dresser un bilan de l'expérience que nous menons depuis deux ans et d'envisager quelques perspectives. L'article sera émaillé de citations de Olivier Girardot, diplômé 2009 devenu développeur OpenOffice.org pendant sa scolarité, Nelle Varoquaux, diplômée 2010 qui a collaboré pendant un an sur des projets en lien avec MarkUs, et Benjamin Vialle, étudiant de la promotion 2011 qui contribue depuis une dizaine de mois à MarkUs. Lorsqu'ils étaient étudiants, ces trois personnes étaient les chefs de projet des groupes qui ont travaillé sur OpenOffice.org et MarkUs.

## II. ORIGINES DU PROJET

À l'heure actuelle, la diversité et la grande variété des logiciels libres déjà existants permet de couvrir de nombreux besoins. Toutefois ces logiciels manquent parfois de fonctionnalités, ou nécessitent un travail d'intégration préalable afin de répondre correctement aux demandes des utilisateurs finaux. C'est dans ce cadre qu'il peut être particulièrement fructueux de mettre en place une collaboration entre une équipe pédagogique et une communauté libre. Nous avons identifié deux besoins importants :

1. la conception et le développement de fonctions d'annotations dans OpenOffice.org, permettant l'utilisation de ce logiciel sur des ordinateurs tactiles (tels que des Tablets PC).
2. la mise à disposition d'un outil de correction en ligne des travaux pratiques en informatique. Nous avons identifié un logiciel libre prometteur, MarkUs, initié à l'Université de Toronto, qui devait être adapté à notre système universitaire.

Nous avons dès lors proposé aux étudiants de l'option informatique de l'École Centrale de Nantes de participer à différents projets itératifs autour d'un module de développement d'annotation pour OpenOffice.org d'une part, de l'intégration de MarkUs au système académique français d'autre part. L'originalité de notre approche réside, à notre sens, dans la confrontation au développement professionnel d'applications que nous donnons à nos étudiants, tout en maintenant leur activité projet : les développements qu'ils ont à effectuer ne sont pas considérables en volume mais ils doivent s'insérer dans un cycle extérieur. Cette professionnalisation s'effectue donc à temps constant. La valorisation pour les étudiants, pour le projet libre comme pour l'établissement est immédiate.

### **III. CARACTERISTIQUES DU DISPOSITIF PEDAGOGIQUE**

#### **III.1 Les communautés logicielles en jeu**

Le code du logiciel OpenOffice.org est connu pour sa complexité. Participer au développement d'OpenOffice.org requiert ainsi d'importantes compétences de génie logiciel et de programmation. OpenOffice.org favorise heureusement la participation d'étudiants au code par l'intermédiaire de son projet Education mené par Eric Bachard. Le projet OpenOffice.org Education vise à faire le pont avec le monde de l'enseignement, en accueillant les nouveaux venus, en organisant hebdomadairement des cours sur le développement au sein du logiciel et en répondant directement aux questions des élèves, notamment par IRC et par une liste de diffusion dédiée.

MarkUs, quant à lui, a vu le jour à l'Université de Toronto en 2005, via plusieurs projets d'étudiants. Une quinzaine de personnes y contribuent fréquemment. Depuis l'été 2009, MarkUs fait l'objet d'une collaboration entre cette Université et l'École Centrale de Nantes, à travers la participation d'étudiants au développement du logiciel (stages, projets, etc.).

#### **III.2 Modalités pédagogiques**

Plusieurs projets, rassemblant à chaque fois entre 2 et 7 étudiants, se sont succédés, de septembre 2008 à mars 2011. Cette continuité est rassurante pour les étudiants au moment de s'investir dans ces projets : en effet, ils peuvent profiter des connaissances synthétisées et rassemblées par leurs aînés dans les rapports de projet, ou sur le wiki du projet Education d'OpenOffice.org [OOo, 2006], quand il ne s'agit pas simplement de demander de l'aide à leurs prédécesseurs, désormais diplômés, mais qui continuent de s'investir sur le logiciel. La complexité de ces projets, loin de décourager les étudiants, les a motivés à s'investir pleinement. Jusqu'à dépasser le cadre de leur scolarité puisque certains sont maintenant des développeurs OpenOffice.org et MarkUs, élément perçu très favorablement par leurs recruteurs.

### **III.3 Éléments de professionnalisation**

#### **III.3.1 Compétences techniques - Savoir-faire**

Grâce à leur investissement dans ces projets libres, nos étudiants ont acquis des compétences nouvelles et utiles dans leur future vie professionnelle. Parmi elles, la capacité à analyser un projet informatique d'ampleur, comprendre l'articulation des différents modules et être capable de se repérer dans le code existant. Citons également la familiarisation avec des processus de revue de code, c'est-à-dire que le code écrit par les étudiants dans le cadre de ces projets est systématiquement contrôlé par des membres de l'équipe de développement du logiciel. Ainsi, comme l'explique Nelle Varoquaux, les étudiants développent des compétences telles qu'une « *immersion rapide dans une base de code* » et le fonctionnement d' « *une assurance qualité* ». Ce que confirme Olivier Girardot, en précisant que la participation à des projets libres « *donne une maturité et une expérience en terme de programmation qu'il est difficile d'acquérir par ailleurs* ».

#### **III.3.2 Compétences humaines - Savoir-être**

Tout comme les logiciels propriétaires, les logiciels libres existent grâce à l'implication d'une communauté. Celle-ci est particulièrement active, et est souvent constituée de professionnels. Il importe donc que les élèves s'immergent dans la communauté, comme ils le feraient dans une entreprise.

## **IV. RECOMMANDATIONS**

Outre la première évidence d'un besoin de rédiger tous les documents en anglais (les logiciels libres fédèrent des personnes de tous horizons, ils sont mondiaux par nature), nous allons établir les principales recommandations utiles à quiconque souhaiterait mener des expériences similaires.

### **IV.1 Assurer une communication fréquente au sein de la communauté académique et de la communauté du logiciel libre**

Nous recommandons l'utilisation régulière et soutenue de médias, tels que des informations sur les sites officiels des communautés en jeu [OOo, 2006], les blogs [ECN, 2008] [ECN, 2010], les conférences institutionnelles [Bachard et Girardot, 2009] [Magnin et Moreau, 2010], etc. En effet, de tels projets transversaux suscitent la curiosité. En donnant une bonne visibilité à ces collaborations, il est possible de sensibiliser de futurs contributeurs et de recruter ainsi des beta-testeurs, des développeurs, ou simplement des personnes qui peuvent aider les étudiants.

#### **IV.2 S'assurer que l'équipe de développement est ouverte aux étudiants**

Afin de mettre en place une collaboration fructueuse, il est primordial de vérifier a priori que la communauté autour du logiciel libre concerné voie d'un bon œil les contributions d'étudiants.

#### **IV.3 Identifier des étudiants intéressés et maintenir leur motivation**

Les tuteurs de tels projets doivent trouver des étudiants avec de bonnes connaissances théoriques, prêts à vivre une première expérience professionnelle de développement, potentiellement intéressés pour continuer à contribuer au-delà du projet (afin de renforcer le transfert de connaissance et d'assurer l'intégration effective des ajouts dans le logiciel, processus souvent assez long). La rigueur et la continuité de la motivation doivent aussi être surveillées de près.

#### **IV.4 Suivre régulièrement les avancées du projet**

Il est fortement souhaitable que ces projets soient encadrés par deux tuteurs : l'un, issu de l'établissement d'enseignement, est le « garant » théorique et pédagogique du projet ; l'autre, choisi dans l'équipe de développement du logiciel, joue le rôle de « *mentor* » auquel les étudiants pourront se référer pour toute question technique liée au développement. La mise en place d'un tel binôme aux connaissances et compétences complémentaires permet de faciliter l'adaptation des étudiants au contexte du logiciel, tout en assurant que le projet évolue conformément aux attentes académiques. Il convient d'organiser des réunions de travail régulières, en présentiel ou à distance, entre les étudiants et leurs tuteurs. Ces réunions donnent lieu à des rapports d'avancement gardant trace des échanges et des solutions envisagées, ce qui peut être utile pour les projets menés les années suivantes.

#### **IV.5 Être conscient des difficultés**

Pour la bonne réussite du projet, le tuteur académique doit adopter un principe de transparence, c'est-à-dire ne pas mentir quant aux difficultés techniques du projet et tenir compte, dans son évaluation, de la complexité de l'environnement (de nombreux outils rarement abordés dans le contexte académique sont souvent mis en œuvre) dans lequel évoluent les étudiants. Qui plus est, il n'est pas toujours simple d'identifier le « bon » projet libre, c'est-à-dire la branche dont la pérennité est assurée, d'où le rôle important de l'enseignant.

### **V. CONCLUSION ET PERSPECTIVES**

Dans ce papier, nous avons présenté les collaborations que nous avons établies avec deux communautés de logiciels libres. À ces occasions, les étudiants ont acquis

des compétences techniques et humaines clairement identifiées... et appréciées par les recruteurs (pour les stages et les offres d'emplois). Benjamin Vialle d'enfoncer le clou : « *en entretien, tout recruteur est plus intéressé par notre contribution à un logiciel libre plutôt que par la réalisation d'un Puissance 4* ». Nelle Varoquaux rajoute que sa participation à OpenOffice.org « *a joué en [sa] faveur* ». Selon elle, ce n'est pas parce « *OpenOffice.org est un projet libre, mais parce que le code est universellement connu pour être complexe* ». Enfin, Olivier Girardot parle également de l'intérêt des professionnels pour la « *maturité* » acquise via ces projets.

Les collaborations avec l'équipe d'OpenOffice.org et de MarkUs ont donné naissance à des cercles vertueux durables. Nos étudiants ont produit du code désormais visible de l'extérieur, dans les sources des logiciels. Ils contribuent ainsi à des projets structurants et professionnalisants, et d'autres étudiants en tirent directement bénéfice. Cela attise leur curiosité et nous avons désormais des étudiants qui, avant même d'arriver dans notre filière « informatique », nous font part de leur souhait de pouvoir y contribuer les années suivantes. Pour approfondir et prolonger ces expériences, nous jugeons utile de maintenir une veille sur le thème des logiciels libres utilisables - notamment - en pédagogie. Le croisement d'enjeux ambitieux (en termes de pédagogie par projets) avec des projets libres aboutit à une relation gagnant-gagnant pour chacune des parties : enseignants, étudiants, établissement d'enseignement supérieur et communauté du logiciel.

## REFERENCES

- Bachard, E. et Girardot, O. "Contribution à OpenOffice.org de projets d'étudiants". Conférence aux 10e Rencontres Mondiales du Logiciel Libre, Université de Nantes, juillet 2009. Nantes, France.
- ECN (École Centrale de Nantes) (2008), Tablets PC à Centrale Nantes, <https://pedagogie.ec-nantes.fr/tablet-pc/> (page visitée en mars 2011).
- ECN (École Centrale de Nantes) (2010), Enseigner et Apprendre avec les Technologies, <http://eat-tice.ec-nantes.fr/> (page visitée en mars 2011)
- GNU (GNU's Not Unix) (1996), Définition d'un logiciel libre, <http://www.gnu.org/philosophy/free-sw.fr.html> (page visitée en décembre 2010).
- Magnin, M. et Moreau, G. (2010). "Retour sur l'usage de Tablet PC à l'École Centrale de Nantes : autonomie et initiatives". Actes du 7ème colloque international "Technologies de l'Information et de la Communication pour l'Enseignement", Universités Nancy-Metz, 6-8 décembre 2010. Nancy, France.
- MarkUs (2009), MarkUs Project, <http://markusproject.org/> (page visitée en mars 2011).
- OOo (OpenOffice.org) (2006). OpenOffice.org Education project, [http://wiki.services.openoffice.org/wiki/Education\\_Project](http://wiki.services.openoffice.org/wiki/Education_Project) (page visitée en mars 2011).