

 <p>LIBERTÉ · ÉGALITÉ · FRATERNITÉ RÉPUBLIQUE FRANÇAISE</p> <p>MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE</p>	<p>Direction générale de la recherche et de l'innovation</p>	<p>Service de la stratégie de la recherche et de l'innovation</p>	<p>Secteur mathématiques, physique, nanosciences et STIC</p>
	<p><b>Groupe de Concertation Thématique</b></p> <p><b>Mathématiques-STIC</b></p> <p style="text-align: right;">Groupe de travail: Industrie du logiciel Code GT: IL</p>		

## L'industrie du logiciel

### Résumé exécutif

L'informatique et les TIC (Technologies de l'Information et de la Communication) sont des technologies diffusantes qui se retrouvent dans l'ensemble des objets, services et échanges de la vie moderne. Au cœur de ces technologies, le logiciel est une composante essentielle de toute innovation.

Le Web unifie l'accès à toutes les données et tous les services et définit les nouvelles normes de fait d'interfaces homme-machine. Il se trouve de plus en plus doté de dispositifs d'entrée / sortie très divers : caméras, capteurs de mouvement, écrans de toutes tailles 2D ou 3D, sons de haute qualité, etc.

Le logiciel embarqué est stratégique pour des secteurs comme le transport, les processus de fabrication industrielle, les télécommunications et tout ce qui touche à la sphère domestique. Cette omniprésence du logiciel au cœur des artefacts de notre vie quotidienne fait surgir des contraintes critiques de sécurité et de fiabilité.

L'arrivée à maturité de technologies de virtualisation ouvre les portes à l'utilisation *flexible* et à la *demande* de ressources informatiques importantes et possiblement délocalisées: c'est le cœur du *Cloud Computing*, qui soulève des questions importantes en termes de programmation à grande échelle et de confidentialité des données.

La généralisation de l'accès au réseau, mobile et très haut débit, couplée avec l'explosion des *terminaux intelligents* comme les mobiles et les tablettes, et au *Cloud Computing*, permettront de proposer au grand public dans des conditions économiques favorables, des applications traditionnellement très gourmandes en ressources de calcul, par exemple les jeux immersifs ou les traductions automatiques audio en temps réel.

Les solutions envisagées aux grands défis sociétaux, santé, aide aux personnes âgées ou handicapées, fluidité des transports urbains, économie d'énergie et diminution de l'empreinte carbone de nos activités, etc. reposent **toutes** sur des innovations logicielles encore à venir.

Dans ce contexte de croissance, il est essentiel que l'industrie du logiciel en France se développe significativement, et ne se réduise pas à un simple réseau de distribution de solutions venant de l'extérieur.

Mais la France représente moins d'un pour cent de la population mondiale, et moins de quatre pour cent du PIB de la planète. Elle ne peut être efficacement présente sur tous les secteurs, et doit choisir intelligemment où concentrer l'effort public en tenant compte des caractéristiques de son industrie.

Étant donnée la difficulté de prévoir l'évolution de la technologie et du marché sur un secteur dont l'évolution est aussi rapide, les recommandations de ce groupe de travail se résument comme suit :

**Améliorer l'image et l'attractivité des métiers de l'informatique.** Cela passe principalement par un effort important de formation et d'information des jeunes générations, dès le collège et surtout au lycée.

**Rendre multidisciplinaire la formation des informaticiens,** en encourageant la diversification des formations des spécialistes, et en assouplissant les frontières entre disciplines dans le système universitaire.

**Encourager les enseignants-chercheurs à s'investir dans des projets d'innovation** en reconnaissant pleinement la valeur et l'utilité de leurs contributions.

**Adapter les mécanismes de financement et d'incitation de l'innovation aux spécificités du logiciel,** notamment par la mise à l'étude de mesures fiscalement favorable pour les entreprises souhaitant contribuer à de grands projets de logiciels libres.

**Construire des indicateurs pertinents** permettant de mieux appréhender la dynamique du secteur de l'innovation logicielle.

**Mettre en place une infrastructure technique mutualisée,** support de grands projets coopératifs de développement logiciel associant les établissements de recherche, les universités, les entreprises de toutes tailles et les pôles de compétitivité qui les fédèrent.

**Concentrer les investissements de la politique industrielle publique sur les domaines à haute valeur ajoutée** pour lesquels on dispose de compétences fortes et dans lesquels il n'y a pas encore d'acteur mondial dominant.

**Maintenir les efforts budgétaires pour recruter des enseignants-chercheurs en informatique, et accroître l'attractivité des carrières d'enseignants-chercheurs pour les chercheurs confirmés français et étrangers.** Les capacités de formation de jeunes techniciens et ingénieurs en informatique en France sont insuffisantes pour répondre aux besoins des industries du secteur et des industries utilisatrices.

<u>Responsable du document :</u> Roberto Di Cosmo, Université Paris Diderot	<u>Statut du document :</u>  Livrable final
<u>Contributeurs :</u> Voir liste dans le corps du document	<u>Destinataires :</u> Membres du groupe de travail : oui Membres du GCT: oui Pilote : oui Autres : oui
Référence document : GT Industrie du logiciel Version finale 2.1 16/06/2011	

## Table des matières

Chapitre 1	<a href="#">Résumé.....</a>	1
Chapitre 2	<a href="#">Contexte, objectifs, organisation et méthode du groupe de travail.....</a>	4
Chapitre 3	<a href="#">L'industrie du logiciel.....</a>	6
3.1	<a href="#">L'édition logicielle.....</a>	6
3.2	<a href="#">Les services.....</a>	7
3.3	<a href="#">Édition logicielle versus service.....</a>	7
3.4	<a href="#">Des changements majeurs.....</a>	7
Chapitre 4	<a href="#">La nature du logiciel et le rôle de la programmation.....</a>	12
Chapitre 5	<a href="#">Attirer et former les acteurs.....</a>	15
5.1	<a href="#">Généraliser la formation à l'Informatique.....</a>	15
5.2	<a href="#">Rénover la formation des Informaticiens.....</a>	16
Chapitre 6	<a href="#">Favoriser l'innovation : une vision globale.....</a>	18
6.1	<a href="#">Soutenir l'innovation logicielle.....</a>	18
6.2	<a href="#">Les limitations du financement et du capital-risque.....</a>	20
6.3	<a href="#">Des modes de financement des infrastructures de recherche peu adaptés au logiciel.....</a>	22
6.4	<a href="#">Un cadre à clarifier pour les activités "valorisation" des enseignants-chercheurs.....</a>	23
Chapitre 7	<a href="#">Résumé des propositions d'actions.....</a>	25
7.1	<a href="#">Améliorer l'image et l'attractivité des métiers de l'informatique.....</a>	26
7.2	<a href="#">Améliorer la formation "multidisciplinaire" des informaticiens.....</a>	26
7.3	<a href="#">Encourager les enseignants-chercheurs à participer à des projets d'innovation.....</a>	26
7.4	<a href="#">Améliorer les mécanismes de financement de l'innovation logicielle.....</a>	27
7.5	<a href="#">Mettre en place une infrastructure technique mutualisée.....</a>	27
7.6	<a href="#">Construire des indicateurs pertinents.....</a>	28
7.7	<a href="#">Investir dans les domaines scientifiques et techniques des TIC à fort potentiel.....</a>	28
Chapitre 8	<a href="#">Conclusion.....</a>	30

## 2- Contexte, objectifs, organisation et méthode du groupe de travail

Presque toutes les innovations modernes contiennent une part importante de logiciel. L'informatique et les TIC sont omniprésentes. On assiste à une montée en puissance significative du logiciel embarqué, ce qui rend essentielle à terme la maîtrise de la sûreté et la qualité. L'externalisation et la virtualisation des ressources de calcul, rendues possibles par le *Cloud Computing*, modifient la dichotomie traditionnelle entre licences et services, et posent des réels problèmes de sécurité des données et de qualité de service. La part de valeur relative du logiciel dans les produits des marchés grand public se développe : c'est vrai de l'automobile, de la domotique, des objets nomades, de l'assistance au 3ème âge, de la télémédecine. On assiste aussi à une véritable explosion du logiciel libre, qui en quelques années a pénétré pratiquement tous les domaines de la production logicielle. Au départ nouvelle forme de *distribution* du logiciel, il induit ensuite une rupture dans les formes d'organisation de la *production* du logiciel, permettant le développement rapide d'applications innovantes, et offrant l'opportunité de prendre des positions fortes sur des marchés jusqu'alors dominés par des acteurs traditionnels.

Renoncer au logiciel serait perdre une place majeure dans l'innovation. Ce rôle clé des TIC est reconnu à sa juste importance dans la stratégie nationale de recherche et d'innovation (SNRI). Dans le cadre de la réflexion sur l'actualisation de la SNRI menée par le ministère de la Recherche et de l'Enseignement Supérieur, un groupe de travail a reçu pour mission de réfléchir à ce qui peut être fait pour accompagner et favoriser le développement de l'industrie du logiciel. D'emblée, il est apparu aux membres du groupe qu'il serait restrictif de limiter la réflexion et l'action simplement au domaine de la recherche : dans le triptyque "invention – innovation – diffusion", la recherche est souvent associée au premier volet, sa finalité étant de découvrir des nouvelles connaissances, mais c'est l'innovation qui, en mettant en pratique ces nouvelles connaissances, crée les nouveaux produits, et c'est la diffusion qui permet de construire de la valeur économique pouvant conduire à la création d'emplois. Maints exemples confirment que les premiers à inventer ne sont pas forcément ceux qui transforment cette invention en produit innovant, les "innovateurs" eux-mêmes n'étant pas forcément ceux qui arriveront à diffuser ce produit et à en tirer des bénéfices.

L'objectif fixé à ce groupe de travail doit donc se comprendre comme une réflexion sur l'ensemble de l'écosystème de l'innovation dans le domaine du logiciel, même si par suite du choix de ce périmètre large, certaines des recommandations formulées par le groupe de travail dépassent les limites des attributions du seul MESR.

Cette vision globale explique aussi pourquoi une place importante a été faite dans le groupe de travail aux acteurs du monde industriel, et en particulier aux PME innovantes ayant des liens forts avec la recherche. Le groupe de travail, dont la constitution est donnée ci-dessous, s'est réuni à plusieurs reprises, identifiant des éléments de réflexion utilisés pour constituer une liste de questions, elles-mêmes soumises à des personnalités extérieures choisies en fonction de leur expérience complémentaire à celle des membres du groupe de travail. Des contributions ont aussi été demandées à divers acteurs de la recherche et de l'enseignement supérieur liés au logiciel, comme Specif, le GDR GPL, et l'alliance Allistene.

### ***Composition du groupe de travail***

Le groupe de travail était constitué des personnes suivantes :

François Bancilhon	Data Publica
Éric Bantegnie	Esterel Technologies
Gérard Berry	INRIA
Patrice Bertrand	Smile
Jacques Blanc-Talon	DGA
Fabrice Derepas	CEA
Roberto Di Cosmo	Université Paris Diderot
Franco Gasperoni	AdaCore
Gaétan Hains	Université Paris-Est Créteil
Laurent Julliard	Minalogic
Gérard Ladier	Aerospace Valley
Alain Michard	MESR

### ***Remerciements***

Nous remercions les personnes suivantes pour les fructueuses discussions autour de ce rapport :

M. Romain Berrendonner	AdaCore
M. Jacques Crémer	Institut d'Économie Industrielle de Toulouse
M. Colin de la Higuera	Laboratoire d'Informatique de Nantes Atlantique
M. Frédéric Duflot	Juriste en propriété intellectuelle
Mme. Catherine Garbay	Laboratoire d'Informatique de Grenoble
M. Jean-Marie Lapeyre	General Motors
M. Yves Ledru	directeur du GDR Génie de la Programmation et du Logiciel
M. Hervé Martin	Laboratoire d'Informatique de Grenoble
Me Édouard Mille	Avocat à la Cour à Paris
M. Ludovic Noël	Imaginove
M. Mathieu Poujol	Cabinet Pierre Audoin Consultants
M. Emmanuel Rondeau	Imaginove

## 3- L'industrie du logiciel

L'industrie du logiciel développe, maintient, déploie, adapte et fait évoluer les logiciels que l'on retrouve aujourd'hui dans presque tous les objets qui nous entourent, allant des avions aux voitures, aux téléphones portables, aux téléviseurs et aux rasoirs électriques, et qui sont utilisés pour gérer tous les aspects de notre vie, des achats dans un supermarché, aux démarches administratives, et à la participation à des réseaux sociaux. Par sa nature, l'industrie du logiciel a un impact économique qui va bien au-delà du marché du logiciel : les innovations réussies dans les systèmes logiciels améliorent directement la productivité dans tous les autres secteurs industriels, et les erreurs et les échecs peuvent produire un coût économique bien supérieur à celui du seul logiciel. Traditionnellement, on tend à séparer l'activité d'édition de logiciel de celles des services informatiques. Nous verrons toutefois que les frontières entre ces domaines sont floues et sont en cours de redéfinition du fait de certaines évolutions du secteur. Nous rappelons ici quelques éléments et chiffres clés de ces domaines.

### 3.1 L'édition logicielle

L'édition logicielle vise au développement d'un produit logiciel répondant aux besoins d'une population d'utilisateurs. En cas de réussite, elle acquiert rapidement une dimension mondiale qui la confronte à une concurrence très rude, mais dont la rentabilité économique peut être élevée. Son modèle économique traditionnel est celui de la vente de licences PUF (« Paid-Up-Front » ou payable d'avance) où l'éditeur perçoit le droit d'usage du logiciel au moment de la vente et y adjoint une partie récurrente plus faible basée sur un service de maintenance à faible valeur ajoutée. Sur les produits grand public, cette seconde part est souvent omise.

Avec l'essor des éditeurs et distributeurs de logiciels libres, de nouveaux modèles se sont mis en place. Ces éditeurs de logiciel libre consentent aux utilisateurs des droits étendus et fusionnent l'offre produit à un ensemble de services à forte valeur ajoutée autour du ou basés sur le produit (conseils d'utilisation, d'intégration, résolution de problèmes, formation, accès à la roadmap du produit, etc.).

On assiste depuis un certain temps dans le domaine de l'édition à un mouvement de consolidation très poussé : selon les chiffres récents de l'étude PAC commanditée par l'AFDEL<sup>1</sup>, plus du tiers du marché mondial de l'édition de logiciel est concentré entre les mains de quatre acteurs majeurs. En France, la concentration du secteur est aussi une réalité, le premier éditeur représentant à lui seul 31 % du chiffre d'affaires logiciels du top 100, et les dix premiers 61 % du total<sup>2</sup>.

Il est naturel devant ces données de se poser la question du rôle que l'industrie française de l'édition logicielle peut bien jouer à niveau mondial. La volonté politique de pousser à la concentration des acteurs existants dans l'espoir de faire émerger des poids lourds mondiaux est compréhensible, mais une telle politique trouve vite ses limites pour deux raisons. La première est que l'édition française est composée d'acteurs sectoriels spécialisés qui se placent sur des marchés différents. Leur consolidation au sein d'un groupe n'est donc pas une garantie de croissance de part de marché pour chacun de leurs produits. La seconde limite vient du fait que certains acteurs qui occupent une place significative du marché du logiciel aujourd'hui ne sont pas le résultat de fusions-acquisitions d'acteurs pré-existants, mais de la création d'offres innovantes répondant aux attentes d'un large marché.

<sup>1</sup><http://www.afdel.fr/communiqu00012fbd.asp>

<sup>2</sup>EuroSoftware, Top 100 des éditeurs français 2010, [http://www.afdel.fr/iso\\_album/eurosoftware100\\_edition\\_2010\\_-\\_top\\_100\\_editeurs\\_france.pdf](http://www.afdel.fr/iso_album/eurosoftware100_edition_2010_-_top_100_editeurs_france.pdf)

## 3.2 Les services

Les industries de service dans le secteur du logiciel ont des activités variées: gestion de processus d'entreprises externalisés (BPO), sécurité des systèmes d'information, détachement de personnel technique en régie, ingénierie informatique, etc. La France a réussi à se positionner à niveau mondial grâce aux sociétés de services en ingénierie informatique (SSII), qui assurent un rôle d'intégrateur au service de grands comptes.

Ce service répond aux besoins spécifiques de clients qui ne peuvent pas se satisfaire de la configuration standard d'un ou plusieurs logiciels d'éditeur. À différence de l'activité d'édition, le service est souvent vu comme une activité locale en raison du besoin de proximité entre le client final et le fournisseur de service. Les SSII sont proches du client non seulement géographiquement, mais aussi et surtout dans la chaîne de création de valeur : elles peuvent jouer un rôle important comme levier pour la diffusion et le déploiement des logiciels, mais elles sont peu enclines à prendre des risques quand il s'agit de très grands acteurs.

Les SSII jouent un rôle modeste dans la *création* des technologies logicielles innovantes. En revanche, elles peuvent jouer un rôle précieux dans le troisième volet de l'innovation, celui des la *diffusion* de ces technologies. Dans cet esprit, la formation continue des ingénieurs et la veille technologique peuvent aider les grandes SSII à établir un contact avec la recherche et avec les PME innovantes, leur permettant *in fine* de maintenir leurs compétences techniques au niveau de l'état de l'art, contribuant ainsi à la compétitivité de leur offre.

## 3.3 Édition logicielle versus service

En théorie économique, et jusqu'à un certain point en pratique, le modèle traditionnel de l'édition logicielle est plus rentable que le service. Dans son incarnation la plus pure, le modèle économique de l'édition consiste précisément à construire des revenus croissants à partir d'une structure de coût fixe : chaque vente supplémentaire d'un logiciel « fermé » rapporte le même montant, alors que le coût pour produire une copie du logiciel est nul. A contrario, dans l'industrie du service, pour chaque nouveau client il est nécessaire de fournir des consultants ou des programmeurs dont les coûts se cumulent. Ce phénomène est particulièrement visible dans le cadre du logiciel embarqué dans les appareils électroniques de notre vie quotidienne : la taille d'une équipe qui met au point un logiciel embarqué spécifique reste la même si ce logiciel est embarqué dans 100.000 appareils, ou dans 100 millions, alors que les revenus de la vente basée sur des « royalties » du logiciel changent énormément. C'est bien pour cela que l'on retrouve de plus en plus de composants logiciels libres génériques dans l'embarqué, les fabricants de matériel étant très réticents à payer des royalties sur des millions d'appareils. Même si l'on observe des contrats entre fabricants et fournisseurs de service en logiciel libre qui font apparaître des « prix de licence », ces prix sont basés sur la valeur du service rendu et correspondent à une stratégie de partage de risque, plus qu'à une licence PUF (Pay Up Front) traditionnelle. Les conclusions à tirer de ces observations sont différentes selon les acteurs. La puissance publique pourrait vouloir soutenir des initiatives orientées services, porteuses de plus de potentiel de créations d'emplois.

## 3.4 Des changements majeurs

Cette vision traditionnelle est en train d'évoluer sous la pression de trois facteurs :

- la croissance très forte du secteur du *logiciel embarqué*, et l'importance que cela donne à la *qualité* du logiciel, qui inclut les dimensions fiabilité, sûreté de fonctionnement, sécurité et adaptation aux besoins des utilisateurs ;
- la virtualisation des plateformes physiques (*cloud computing*) qui conduit à rendre économiquement rentables l'hébergement sur réseau et la mutualisation des applications logicielles ;

- la montée en puissance d'entreprises centrées sur le *logiciel libre*, qui réalisaient en 2008 un chiffre d'affaires représentant 3,6% (soit 1105 millions d'euros) de la demande en logiciels et services en France<sup>3</sup>.

### ***Le logiciel embarqué***

Le logiciel est devenu un élément essentiel d'un grand nombre d'objets dont le bon fonctionnement est critique pour la vie (ou au moins pour le confort) d'un grand nombre de personnes: véhicules terrestres ou aériens, réseaux de transports collectifs, systèmes de production et de distribution d'énergie, etc. Embarqué dans du matériel spécialisé, il fournit l'intelligence qui contrôle des phénomènes physiques, et les erreurs éventuelles dans ces logiciels ont un impact potentiellement bien plus important que des erreurs dans des logiciels de bureautique : un avion peut tomber, une voiture peut être immobilisée ou pire ne plus s'arrêter, un *pace-maker* peut être dérégulé.

On est donc prêts à investir beaucoup d'efforts dans la validation de ces logiciels embarqués, ce qui renvoie à des défis scientifiques majeurs pour trouver les méthodes rigoureuses et industriellement exploitables permettant de prouver et de vérifier la sûreté et la conformité de ces logiciels.

Les normes strictes utilisées dans le monde de l'avionique ne sont pas encore obligatoires dans les autres domaines industriels, mais on peut prévoir que la situation changera dans les prochaines années, surtout si des avancées scientifiques permettant d'abaisser le coût de la validation du logiciel. Les domaines des transports individuels et collectifs, des grilles de distribution d'énergie et de la santé sont particulièrement concernés.

La France dispose des véritables atouts dans ce domaine très concurrentiel et pas encore stabilisé, pour lequel il faut aider à émerger et consolider des acteurs de taille mondiale. Ce point est traité spécifiquement dans le rapport " Briques Génériques du Logiciel Embarqué" paru en Octobre 2010<sup>4</sup>.

### ***Réseau, Mobile, Web, Données et Cloud Computing***

Un autre changement majeur dans le monde des TIC est dû à la disponibilité de réseaux à haut débit et la virtualisation du matériel qui ont permis de construire des offres de ressources flexibles et à la demande, communément appelées *Cloud Computing*. Sur cette base technologique se construisent des innovations très diverses. On y trouve les applications d'hébergement de contenu du Web 2.0 qui l'utilisent pour pouvoir absorber à moindre coût les pics ponctuels de demande. On voit aussi des offres de plateformes de programmation verticalisées conçues pour le web, comme *SalesForce.com*, et plus généralement se multiplient les offres de services en ligne, qui tendent à gommer la frontière entre édition et service, le modèle économique basé sur l'abonnement mensuel couvrant l'ensemble.

La généralisation du *Cloud Computing* a plusieurs conséquences : d'un côté, on observe l'importance de l'effort d'investissement, récurrent, pour les grands fournisseurs d'infrastructure, afin d'offrir un service de qualité suffisante ; de l'autre côté, on voit diminuer très fortement le coût de l'innovation basée sur les services en ligne en raison de la possibilité de (re)dimensionner facilement les ressources en fonction des besoins, avec des coûts d'entrée très faibles (la plupart des jeunes entreprises du web utilisent le *Cloud* pour leur infrastructure).

Les acteurs traditionnels qui ont jusque-là joué la carte des solutions propriétaires se retrouvent coincés entre le logiciel libre, qui vient les concurrencer sur leur terrain traditionnel, et les services hébergés en ligne, qui offrent nettement plus qu'une simple mise à disposition de licence PUF. C'est pour cela qu'on voit tous les grands acteurs de l'édition essayer de se positionner rapidement sur le *Cloud Computing*, au moins en termes de marketing.

On observe aussi la mise à disposition sur le réseau d'une masse de données croissante, venant en

---

<sup>3</sup>Étude sur les métiers du logiciel libre, FAFIEC 2008

<sup>4</sup><http://www.ladocumentationfrancaise.fr/rapports-publics/104000528/index.shtml>



partie des données rendues publiques par les internautes eux-mêmes, et en partie par la mise à disposition de données publiques à travers des initiatives comme *Open Data*<sup>5</sup> : cela ouvre les portes pour des applications innovantes, d'autant plus utiles qu'elles peuvent désormais compter sur une multitude de dispositifs d'accès : ordinateur traditionnel, netbook, tablette internet, téléphone mobile.

La disponibilité de réseaux à très haut débit et de ressources flexibles fournies par le Cloud ouvre aussi la porte à une évolution dans le monde du jeu vidéo, où les consoles de jeu pourraient être remplacées dans un futur proche par des terminaux bien plus simples, comme les divers "box" qui donnent l'accès à Internet, l'essentiel du calcul lourd étant effectué dans le Cloud. Cela permettrait à cette industrie de passer d'un modèle de vente de licence PUF à un modèle d'abonnement, et de réduire significativement les copies de jeux non autorisées.

Ces changements bouleversent la séparation traditionnelle entre grands acteurs de l'édition et du service, qui se rapprochent à la fois à travers le modèle de développement en logiciel libre et l'essor de la fourniture de services logiciels à la demande à travers le Cloud, avec l'évolution des modèles économiques que cela implique.

En même temps, la montée en puissance des *AppStores*, plateformes de vente en ligne d'applications pour mobiles, mais aussi les plateformes de jeux en ligne sans consoles, comme celles proposées par Zinga ou Facebook, redonnent des couleurs à la petite édition individuelle qu'on avait connue au début des années 1980 avec les PC, avec une prime à la créativité et à la rapidité.

### ***Le logiciel libre et le développement collaboratif***

On appelle logiciel libre un logiciel dont la licence donne à l'utilisateur le droit de l'utiliser, de le modifier et de le redistribuer, y compris avec ses modifications<sup>6</sup>. Le logiciel libre est apparu voici plusieurs décennies dans la communauté des chercheurs grâce au support fourni par les premiers systèmes Unix à la fin des années 70, mais son irruption explosive dans le monde économique est plus récente. C'est un phénomène de rupture sur plusieurs plans :

- il a un impact direct sur la technologie informatique, puisqu'il change la façon de développer, distribuer et commercialiser les logiciels ;
- il modifie en profondeur le secteur industriel de l'informatique, en redistribuant les cartes et ouvrant des marchés qui étaient considérés comme fermés et insaisissables, des systèmes d'exploitation aux bases de données et à la business intelligence ;
- il a un impact sur l'ensemble des secteurs industriels qui voient le coût d'entrée de technologies informatiques essentielles baisser considérablement, ainsi qu'une opportunité de reprendre la maîtrise de leur système d'information ;
- il contribue au développement de la société en construisant un patrimoine technologique commun et universel, et en permettant aux citoyens de s'approprier les technologies essentielles ;
- il provoque, accélère, ou facilite des évolutions profondes dans la manière de travailler des informaticiens, depuis les développeurs des technologies de base jusqu'aux intégrateurs des SSII. Cette évolution importante et méconnue du public est décrite ci-après, car elle induit de nouveaux besoins en matière de formation des informaticiens, besoins encore mal pris en compte en France.

---

<sup>5</sup><http://www.regardscitoyens.org/open-data-en-france/>

<sup>6</sup>Voir <http://www.gnu.org/philosophy/free-sw.html> et <http://www.opensource.org/> pour plus de détails.

## ***Développement Collaboratif***

Une des caractéristiques les plus intéressantes des logiciels libres qui ont du succès est le développement distribué et collaboratif dont ils font l'objet. Même si chaque communauté de développeurs a son organisation propre, on retrouve régulièrement un certain nombre d'outils pour supporter le développement distribué, qui vont des systèmes de contrôle de version aux forges et aux listes ou forums de discussion.

On remarque aussi dans les logiciels libres qui ont du succès la présence d'une architecture logicielle particulièrement adaptée pour la collaboration. Tout développement logiciel libre commence par une phase initiale qui voit intervenir un petit groupe de développeurs. Pour atteindre une large diffusion, ce logiciel doit rapidement fédérer une communauté beaucoup plus large d'acteurs qui contribuent au développement, à la correction, à la documentation, à la promotion de l'utilisation de ce logiciel. Pour que ceci soit possible, il est essentiel que le prototype original permette aux nouveaux venants d'apporter des contributions sans avoir à faire un effort trop important, et que la politique d'acceptation des contributions soit claire et pas trop restrictive<sup>7</sup>.

De tels projets nécessitent donc de fortes compétences en architecture logicielle, mais aussi en gestion de communautés. Ces compétences complémentaires sont cruciales, et on les retrouve dans tous les grands projets à succès comme Linux, Eclipse, Apache ou Debian.

On peut dire que la clé du succès d'un logiciel libre réside dans *le système sociotechnique de collaboration* qui le supporte, et les leçons à tirer de ces succès peuvent être transposées à quelques nuances près à l'ensemble de l'industrie du logiciel.

## ***Le Logiciel Libre comme pont entre édition et service***

Les projets de logiciel libre fonctionnent de façon très transparente puisque tout le code source est accessible à tous et que chaque participant peut suivre les évolutions du produit à travers les versions et les correctifs successifs. Les ingénieurs des SSII qui sont ou seront appelés à utiliser la technologie développée comme composant des solutions vendues à leurs propres clients peuvent donc suivre de près le ou les projets de développement qui les concerne. Ce suivi, même s'il ne donne pas lieu à une contribution en code source, est extrêmement formateur pour l'ingénieur qui peut comprendre l'architecture du logiciel, les raisons des choix techniques, appréhender les limitations fonctionnelles et avoir une bonne vision de la "roadmap", c'est à dire des modifications et améliorations à attendre des versions futures. Cet ingénieur utilisateur peut également influencer sur les choix techniques pour tenter de les orienter pour mieux répondre à ses propres besoins d'intégrateur.

Dans certains cas (logiciels de la fondation Apache notamment), les développeurs sont pour la plupart des ingénieurs rétribués par des SSII utilisatrices, qui consacrent une part significative de leur temps de travail à contribuer activement au projet de logiciel libre. De grandes SSII par exemple IBM sont bien connues comme des contributrices "en nature" aux projets de logiciels libres. Cette implication est particulièrement bien vue par les "grands comptes" clients finaux de la SSII qui acceptent d'autant plus facilement des solutions bâties sur des technologies "libres", qu'ils savent que leur intégrateur entretient des liens étroits avec la communauté de développement de celles-ci.

## ***Une opportunité pour la France***

L'importance de la mise à disposition d'un ensemble de briques génériques pour le logiciel embarqué est clairement reconnue par le rapport "Briques Génériques du Logiciel Embarqué" déjà cité, et l'étude PAC pour la commission européenne de 2010 montre aussi l'opportunité de la création d'acteurs européens forts sur le créneau des intergiciels (*middleware*).

---

<sup>7</sup>Anthony Senyard and Martin Michlmayr, "How to Have a Successful Free Software Project", APSEC 2004

Selon une étude RedHat - GeorgiaTech publiée en 2009<sup>8</sup>, la France est numéro un en terme d'activité de développement et d'intégration basée sur le logiciel libre au niveau de la commande publique et numéro trois au niveau de la communauté éducative (enseignement et recherche). En revanche, l'adoption dans le secteur privé est beaucoup plus faible, la France ne s'octroyant que la 25ème place, derrière la Finlande, la Norvège, le Danemark, les Pays Bas, l'Estonie ou l'Australie. En terme d'« environnement », la France se classe au 15ème rang, loin derrière la Suède – numéro un –, mais surtout les États-Unis, qui prennent la deuxième place de ce classement prospectif, fondé sur des critères éducatifs, de pénétration des technologies, ou encore socio-économiques. Si l'on en croit les conclusions de cette étude, le logiciel libre a bien décollé en France grâce au soutien du secteur public, mais ce décollage ne se transformera en réelle opportunité économique que si l'écosystème global de l'industrie du logiciel évolue favorablement.

Nous reviendrons plus loin sur cette notion d'écosystème de l'industrie du logiciel, mais il est important de noter déjà quelques éléments factuels sur la nature des entreprises actives en France dans le domaine du Logiciel Libre.

Selon l'étude publiée par le Conseil National du Logiciel Libre en 2010, ces entreprises sont plutôt de petite taille, mais ont des grands comptes comme clients et ont très bien résisté à la crise. Elles font beaucoup de développement et d'intégration, avec un investissement important en recherche et développement. La majorité d'entre elles contribuent à au moins un projet en logiciel libre, ce qui rend plus **floue la frontière entre édition et service**. Elles cherchent à recruter des bons programmeurs ayant une connaissance avancée du logiciel libre, et rencontrent des difficultés pour ces recrutements.

Les modèles économiques utilisés par ces acteurs sont multiples: on trouve en général un mélange de service, d'abonnement, de formation, de développement à façon, et d'offre de services d'édition. On note aussi qu'une partie importante du chiffre d'affaires est faite à l'exportation, et que la petite taille de ces entreprises ne leur empêche pas d'être bien positionnées sur des marchés importants, en raison de la qualité de leur offre.

Dans un contexte international hautement compétitif, la France peut viser une place de premier plan sur le logiciel libre. Pour cela, le premier défi à relever est celui de la formation des informaticiens développeurs, inadéquate en France aussi bien au plan quantitatif que qualitatif. Nous revenons plus longuement sur ce point dans les chapitres 4 et 5.

---

<sup>8</sup>Voir l'étude RedHat à l'adresse [http://www.redhat.com/f/pdf/OSSI\\_Research.pdf](http://www.redhat.com/f/pdf/OSSI_Research.pdf) et l'analyse publiée par LeMagIT : <http://www.lemagit.fr/article/france-redhat-open-source-index/3135/1/la-france-championne-open-source-aux-pieds-argile/>

## 4- La nature du logiciel et le rôle de la programmation

Le logiciel étant au cœur de toutes les innovations majeures d'aujourd'hui, on pourrait s'attendre à une ruée vers les métiers de l'informatique, et une allocation massive de ressources publiques pour la formation à cette discipline depuis le plus jeune âge. Cela n'est pas le cas aujourd'hui, et l'on observe une certaine incompréhension chez la plupart des publics de ce qu'est réellement l'informatique comme science du développement des logiciels ou des artefacts intégrant matériel et logiciel. On entend encore souvent le terme de « nouvelles technologies de l'information et de la communication ». Or, ce terme est devenu vraiment obsolète, puisque les jeunes d'aujourd'hui n'ont pas connu le monde d'avant !

### Le grand public

associe l'informatique à des outils divers et variés, comme les tableurs, le traitement de texte, les navigateurs, et autres applications particulières, ainsi que bien évidemment, aux virus et bogues qui polluent leur vie. Les efforts, par ailleurs louables, comme le B2I et le C2I qui ont été faits pour familiariser les nouvelles générations avec certains de ces outils ne donnent pas de l'informatique une image très différente. On reste au niveau de l'usage d'applications prises comme des boîtes noires, et dont on ne comprend ni le fonctionnement ni les limitations ou les défauts. On ne doit donc pas s'étonner si le modèle mental dominant ne place pas l'informatique parmi les *sciences* dignes de ce nom.

### Les spécialistes des disciplines traditionnelles

comme les mathématiques, la physique et bien d'autres ont parfois tendance à assimiler l'informatique à un ingénieur support ou à un chercheur de seconde catégorie : après tout, le mathématicien ou le physicien écrivent eux aussi du code logiciel, au service de leur savoir faire scientifique premier, résoudre des équations différentielles ou modéliser des phénomènes physiques. Les informaticiens écrivent aussi du code, mais ne savent pas résoudre des équations différentielles, pour un regard superficiel, leur savoir est donc limité au développement d'outils au service des "vrais chercheurs". Ce même regard superficiel pourrait expliquer le fait que l'on trouve bien moins d'ingénieurs de recherche dans les laboratoires d'informatique qu'ailleurs, les demandes des laboratoires étant rarement satisfaites : si un informaticien est quelqu'un qui écrit du code, on peut se dire qu'on ne voit pas pourquoi il aurait besoin d'un ingénieur de recherche qui fait la même chose.

### Les institutions,

dont le rôle devrait être de promouvoir cette science plus tellement jeune qu'est l'informatique se montrent très lentes dans leur compréhension de la discipline. La composition des conseils fait peu ou pas de place aux informaticiens. Les structures de valorisation de la recherche, si l'on en juge par les campagnes régulières de formation offertes aux enseignants et chercheurs qui mettent systématiquement en avant les brevets comme instrument de valorisation, ne paraissent connaître que très peu le monde du logiciel. Les mécanismes d'évaluation de la recherche ne prennent pas non plus en compte formellement l'activité de développement de logiciels dans l'activité des informaticiens. Même s'il est vrai que les instances universitaires françaises donnent plus de poids à l'activité de développement qu'en Italie ou en Espagne, grâce notamment à des prises de position de certains membres de la CNU ou à la réflexion des instances d'évaluation de l'INRIA<sup>9</sup>, cela ne se retrouve pas dans les guides

<sup>9</sup>David Margery, Jean-Pierre Merlet, Cordelia Schmid, Agnès Sulem, Paul Zimmermann (rapporteur) *Évaluation des*

d'évaluation de l'AERES. Il n'est guère surprenant si après, des nombreux chercheurs français en informatique considèrent qu'il est plus payant pour la reconnaissance de leurs pairs et pour leur carrière de démontrer un théorème que de développer un code quelque peu complexe.

## Les architectes

Pendant des décennies, l'activité d'intégration au service de marchés clients verticalisés qui a fait le succès des SSII a reposé sur des architectes qui conçoivent les grandes lignes des systèmes d'information, pour la plupart à partir de briques logicielles qu'ils ne réalisent pas, et qu'ils sont obligés de traiter comme des boîtes noires : quarante ans d'édition logicielle « fermée », en empêchant de voir le code source des logiciels, ont fini par effacer l'intuition même de la nature de ces logiciels de la culture technique des décideurs. On trouve donc largement répandue l'idée que, une fois dessinés quelques diagrammes conceptuels décrivant l'architecture de l'application, tout le reste n'est qu'un « détail d'implémentation », qui peut être sous-traité en Inde ou en Chine à moindre coût. Cette pratique, très connue sous le terme anglais de « *outsourcing* », a bien montré ses limites : le gain apparent en coût de main-d'œuvre est souvent compensé par les pertes liées aux défauts qui n'apparaissent qu'à la réception, tantôt dues à des spécifications ambiguës ou incomplètes, tantôt dues simplement au fait que les spécifications ont évolué après que le contrat d'outsourcing ait été finalisé, tantôt dues au fait que le sous-contractant se borne à réaliser ce qu'on lui a demandé le plus vite possible, sans trop de souci de qualité.

Cette vision est présente chez un grand nombre d'acteurs: quand il s'agit de développer des applications, innovantes ou non, la partie « noble » résiderait toujours dans l'idée scientifique ou commerciale, le reste n'étant que des « détails d'implémentation », à déléguer à des sous-traitants. Bien des collègues académiques sollicités pour contribuer à ce rapport ont reçu des sollicitations qu'on pourrait résumer dans une phrase assez crue comme « *tu n'aurais pas un étudiant qui s'y connaît en développement web pour réaliser dans le cadre de son stage l'application révolutionnaire que j'ai imaginé et qui va me valoir gloire et notoriété ?* ».

Pourtant le logiciel, complémentaire du matériel, est l'objet essentiel nécessaire pour rendre effectives toutes les innovations informatiques : qu'on réalise une simulation avec des équations sophistiquées ou de la compression d'images ou de la recherche d'informations avec des algorithmes avancés, il faut à un certain moment coder les équations et les algorithmes dans un langage de programmation, et produire un logiciel qui les met en pratique. Or écrire un logiciel concis, correct et efficace est une activité difficile : une page de code est bien plus dense qu'une page de texte, et c'est bien celui qui sait réaliser les logiciels qui a, en définitive, le pouvoir de transformer en réalité les grandes idées algorithmiques ou architecturales, qui resteraient autrement lettre morte. C'est d'ailleurs souvent celui qui a eu les idées qui les code en premier et en assure le développement. Dans un film dédié à l'histoire de Facebook sorti récemment en salle, on trouve un passage édifiant : accusé d'avoir « volé » l'idée de Facebook, Marc Zuckerberg répond à ses accusateurs à peu près comme suit: « si c'était tellement facile à faire, vous l'auriez fait vous-mêmes ».

Le terme « Programmation » recouvre l'ensemble des méthodes et connaissances nécessaires pour réaliser correctement les logiciels, les faire évoluer et les transformer en code exécutable par un ordinateur ; c'est un des axes fondamentaux de l'informatique, comme l'algorithmique, les réseaux, ou l'architecture du matériel. Il reste beaucoup d'efforts à faire en France pour faire admettre cette place essentielle de la programmation dans la culture scientifique et technique des jeunes ingénieurs et chercheurs informaticiens.

## **Conséquences**

Cet état de fait a des conséquences négatives sur l'innovation que l'on ne doit pas sous-estimer : l'échange entre industrie et recherche, qu'il s'agisse de créer une entreprise innovante ou de contribuer à un projet industriel, pourrait être particulièrement efficace en informatique en utilisant le logiciel comme support du transfert de connaissances et de compétence, mais la mauvaise image du développement logiciel et la non-prise en compte des activités de développement dans la carrière des enseignants-chercheurs la limitent inutilement.

**Proposition n°1 :** Prendre en compte explicitement la production logicielle de qualité des enseignants-chercheurs dans l'évaluation de leur activité scientifique.

**Proposition n°1 bis** Des ingénieurs de développement et de recherche en nombre suffisant doivent travailler avec les chercheurs pour accélérer l'ingénierie des prototypes de recherche et maintenir les plateformes et outils de développement.

## 5- Attirer et former les acteurs

### 5.1 Généraliser la formation à l'Informatique

Le moyen le plus efficace sur le long terme pour faire comprendre la pleine valeur intellectuelle et économique du développement logiciel est la mise en place d'une stratégie qui aide l'*ensemble* (ou tout au moins une large part) de la population à se constituer un modèle mental correct de ce qu'est l'informatique. Cela passe, comme déjà souligné dans le document actuel de la SNRI, par l'introduction de l'informatique dans l'enseignement secondaire, et la revalorisation de l'enseignement de l'informatique dans le supérieur.

Cela permettra d'avoir à terme des décideurs à tous les niveaux mieux informés, plus de vocations vers les métiers de l'informatique, une meilleure formation à l'informatique parmi les étudiants d'autres disciplines, et plus stratégiquement, une population *mieux préparée* pour la société de la connaissance qui se construit en ce moment. Ce dernier point est probablement la motivation la plus profonde pour introduire rapidement de l'enseignement de l'informatique dans les programmes, et un récent rapport en Grande-Bretagne fait la même analyse<sup>10</sup>.

Aux États Unis, des initiatives ont vu le jour depuis plusieurs années diffuser largement cette idée et un large débat anime le monde éducatif sur les différentes approches possibles (voir par exemple les débats sur le *Computational Thinking*<sup>11 12</sup>). Quelles que soient l'acuité et la pertinence de ce débat, le besoin est maintenant unanimement reconnu.

Une première étape dans cette direction se met en place en France depuis 2010 par la formation à l'informatique de formateurs qui vont eux-mêmes mettre à niveau des enseignants de lycées venant d'autres disciplines, en vue de l'ouverture d'une option informatique dans certains lycées volontaires. Pour louable que puisse être cette initiative, elle semble timorée au regard des enjeux. Pour aller plus loin, il faudra non seulement introduire l'informatique comme *discipline* à part entière au moins dans les sections scientifiques des lycées, au même titre que par exemple la physique ou la chimie, y affecter des enseignants informaticiens convenablement formés, et surtout mettre en place la structure de formation permanente nécessaire pour garantir une mise à niveau régulière des enseignants dans cette discipline qui évolue extrêmement rapidement.

**Proposition n°2 :** Faire de l'informatique une discipline à part entière dans l'enseignement secondaire, et créer les corps d'enseignants, les concours de recrutement et les corps d'inspection correspondants.

Un autre axe d'action, très important, consiste à compléter les cursus d'enseignement supérieur non informatiques par des formations complémentaires dans le domaine des TIC. L'enjeu est de favoriser à tous les niveaux l'hybridation des compétences TIC et non TIC.

**Proposition n°3 :** Introduire de manière systématique l'enseignement de l'informatique dans les cursus d'enseignement supérieur non informatiques.

Pour aller plus loin, on peut envisager de soutenir des concours de programmation à différents niveaux, au collège et au lycée, pour un public plus large que celui visé par le concours de l'ACM. Doté de prix attractifs (micro-ordinateurs, etc.) ces concours peuvent faire connaître la discipline à un jeune public.

Il est aussi souhaitable de relayer dans les universités et grandes écoles françaises des initiatives

<sup>10</sup><http://www.itpro.co.uk/610078/students-need-more-it-challenges>

<sup>11</sup>*Computational Thinking*, Wikipedia, [http://en.wikipedia.org/wiki/Computational\\_thinking](http://en.wikipedia.org/wiki/Computational_thinking)

<sup>12</sup>*Computational Thinking - A problem-solving tool for every classroom*  
<http://education.sdsc.edu/resources/CompThinking.pdf>

comme le *Summer of Code* de Google<sup>13</sup>, centrées sur du logiciel libre et avec une attention particulière aux projets français. Ces actions permettent aux étudiants d'entrer en contact directement avec de vrais projets industriels et de découvrir des projets de développement collaboratifs internationaux.

**Proposition n°4 :** Favoriser et accompagner les initiatives de « main à la pâte » dans l'Informatique, sous la forme de concours de programmation à tous les niveaux.

Tout ceci demande des moyens considérables : les 3.500 enseignants chercheurs de la 27ème section sont loin d'être en nombre suffisant pour assumer la formation des maîtres en informatique, la formation des étudiants universitaires non informaticiens à l'informatique, la formation des étudiants spécialistes de la discipline, répondre aux besoins de l'interdisciplinarité et contribuer à l'innovation industrielle.

## 5.2 Rénover la formation des Informaticiens

Une étude récente de l'OPIIEC (Observatoire paritaire des métiers de l'Ingénierie, de l'Informatique, des Études et du Conseil, association paritaire de branche regroupant les syndicats d'employeurs et de salariés), fait état d'un manque réel de compétences, surtout au niveau "chef de projet", ressenti par les entreprises, et en particulier pour ce qui concerne le développement collaboratif, tant pour ce qui concerne le logiciel libre que d'autres grands projets de développement. Le rapport mentionne notamment parmi ses recommandations<sup>14</sup> :

Investir dans la formation initiale

Des investissements sont nécessaires à ce niveau, car il manque de professeurs pour former des informaticiens de haut niveau capables de faire de la R&D, qui est une activité à forte valeur ajoutée. Dans cette optique, le logiciel libre est important, car il permet une bonne compréhension du fonctionnement d'un logiciel et d'un système informatique, puisqu'on a accès à toutes les parties du logiciel ou du système. On peut ainsi former des spécialistes informatiques qui en plus sont familiers avec le fonctionnement des communautés et des projets en logiciel libre, ce qui permet à la France de conserver son avance sur le sujet. De plus, la communauté est comme une formation continue, basée sur l'émulation. Le logiciel libre est un facteur d'attraction pour les jeunes étudiants vers les carrières scientifiques. Néanmoins, il est important d'étudier les deux facettes du logiciel, le logiciel fermé et le logiciel libre.

Sur le contenu de la formation nécessaire, ce même rapport souligne qu'il ne suffit pas de former des "codeurs" de faible niveau, mais des personnes capables d'inscrire leur travail dans un projet de grande envergure :

Le logiciel libre demande aussi des compétences en gestion de projet, en anglais et en capacité à travailler avec des équipes qui sont mondiales, farouchement indépendantes et qui travaillent d'une manière très différente de ce que l'on rencontre en entreprise. ***Ce développement communautaire et planétaire préfigure aussi les changements majeurs dans l'organisation informatique des entreprises.***

Les programmes de formation actuels en informatique n'intègrent pas de façon suffisante la dimension de développement collaboratif, ni les potentialités offertes par le logiciel libre : pour participer efficacement à de grands projets de développement modernes, l'informaticien doit non seulement avoir de bonnes bases scientifiques et techniques, mais il doit en outre apprendre à insérer son travail dans un ensemble complexe, produit par une large communauté.

L'éditorial de mars 2006 dans *Communications of the ACM*, écrit par David A. Patterson, à l'époque président de l'ACM, porte le titre *Computer Science Education in the 21st century*. Ses analyses et propositions peuvent se résumer que quelques mots simples, mais percutants: pour avoir

<sup>13</sup><http://code.google.com/intl/fr/soc/>

<sup>14</sup>OPIIEC - Étude sur les métiers du logiciel libre en France - Septembre 2008



une forte valeur ajoutée aujourd'hui, il faut enseigner l'informatique *autrement*, en profitant de l'opportunité unique de "*mettre la main à la pâte*" offerte coté matériel par la banalisation des circuits programmables (composants des systèmes embarqués) et cotés logiciel par l'ouverture du code dans les communautés (souvent industrielles) du logiciel libre.

Mais il ne s'agit pas seulement de former à l'informatique autrement: il s'agit aussi de transmettre une vision plus globale, où le logiciel est un élément parmi d'autres, permettant de réaliser des objets et services innovants. Pour cela, il faut favoriser l'émergence de cursus mixtes dans lesquels l'étudiant découvre que l'informatique au sens strict n'est qu'une compétence parmi d'autres, toutes indispensables à la réussite d'un projet d'innovation ambitieux : la microélectronique, l'architecture, le design, l'économie, l'ergonomie, la didactique technique, le management d'équipes et de projets, la connaissance des techniques de validation, de certification, bref d'industrialisation du logiciel, contribuent également à la réussite globale.

En d'autres termes, on ne peut plus penser au logiciel de façon séparée de son environnement, qui est composé à la fois du matériel dans lequel le logiciel peut être embarqué, des usages auxquels le logiciel fourni le support, et du modèle économique qui va le faire vivre : tout ceci devient rapidement un mélange indissociable, et il faut des compétences nouvelles pour réussir.

Aux États-Unis, certaines universités ont mis en place des cursus associant la formation technologique à une ou plusieurs des disciplines des "*liberal arts*" évoquées ci-dessus. En France, la logique "disciplinaire" semble pour l'instant la plus forte.

<p><b>Proposition n°5 :</b> Modifier les cursus spécialisés en informatique pour y intégrer des approches multidisciplinaires, qui tiennent compte des apports de disciplines connexes, comme l'ergonomie, l'économie, les mathématiques appliquées, la gestion de projet, les techniques de validation et de certification du logiciel, de contrôle qualité, etc.</p>
--

## 6- Favoriser l'innovation : une vision globale

La formation, la recherche, l'innovation et le transfert forment un continuum qui relie universités, grandes écoles, instituts publics de recherche, centres privés de recherche, sociétés de technologies, industriels intégrateurs et utilisateurs, investisseurs et pouvoirs publics.

D'innombrables ouvrages et rapports ont été consacrés à analyser le processus de l'innovation, à comprendre comment chaque acteur peut et doit interagir avec les autres dans un "écosystème vertueux" qui favorise l'émergence des idées neuves et leur transformation en produits innovants trouvant leur cible de marché.

Depuis plus de vingt ans, la France n'a pas ménagé ses efforts pour tenter de dynamiser cet écosystème, de faciliter les relations entre les différents acteurs. De très nombreux dispositifs ont été imaginés à ces fins : concours national des entreprises innovantes, possibilité pour des chercheurs d'être détachés dans de jeunes entreprises innovantes en création, incubateurs publics, mécanismes de subventions ou d'avance remboursable gérée d'abord par l'ANVAR puis par OSEO, crédit d'impôt recherche, statut fiscal de Jeune Entreprise Innovante, incitations fiscales à l'investissement dans les PME, investissement direct de l'État dans des PME innovantes ("investissements d'avenir"), création des pôles de compétitivité visant à faciliter les coopérations entre laboratoires publics et PME et entre les PME elles-mêmes, accès des PME innovantes aux marchés publics, subventions à des projets de R&D associant des laboratoires et des entreprises, tant au niveau national qu'à celui de l'Union Européenne.

Tous ces dispositifs jouent leur rôle, avec plus ou moins d'efficacité et de complexité et en visant différentes étapes du cycle d'innovation.

Dans cette section, nous résumons un consensus remarquable qui a émergé entre les experts interrogés sur les aspects atypiques de la valorisation dans l'industrie du logiciel, dont la compréhension est importante dans la détermination des verrous à lever pour permettre l'essor de la créativité et de l'innovation dans le secteur du logiciel.

### 6.1 Soutenir l'innovation logicielle

#### *Le brevet, un instrument inadapté au soutien de l'innovation logicielle*

Dans des domaines scientifiques et techniques pour lesquels le cycle de développement d'un produit est très long et où les innovations se réduisent à quelques éléments clefs facilement identifiables (la formule chimique d'un médicament et son procédé de fabrication, par exemple), l'instrument du brevet est utilisé comme pilier de la stratégie de valorisation : en garantissant un monopole de durée importante sur des procédés qui sont coûteux à mettre à point, il permet de centraliser les revenus qui peuvent être importants si le brevet est effectivement exploité, et de les répartir sur la base d'accords de licence qui prévoient des pourcentages fixés à l'avance, entre les chercheurs, les organismes de recherche et les industriels.

Dans le domaine du logiciel, la notion de brevet est très loin de recueillir un consensus. D'un côté, le législateur a semblé exclure clairement du champ de la brevetabilité ceux portant sur les logiciels. Cependant, les tentatives répétées de plusieurs acteurs industriels de changer la donne ont conduit en Europe à un état de grande confusion et d'insécurité juridique, personne ne sachant exactement sous quelles conditions il est possible d'obtenir un brevet logiciel, ni ce que vaut vraiment un brevet logiciel obtenu.

De l'autre côté, il est fréquemment observé que dans l'industrie du logiciel les brevets sont utilisés beaucoup plus comme armes de guerre économique que comme instruments de valorisation de la

recherche : c'est le cas des grands portefeuilles de brevets, traditionnellement aux mains de grandes entreprises américaines, mais qui se retrouvent aujourd'hui aussi dans celles de cabinets spécialisés qualifiés de "*patent trolls*", dont la seule finalité est de gagner de l'argent en brandissant des conflits de brevets avec les vrais acteurs du domaine, et dont l'utilité économique est fort douteuse. Au vu de la répartition des portefeuilles de brevets logiciels dans le monde, il n'y a aucun intérêt économique à ouvrir les portes en Europe aux brevets logiciels : cela reviendrait à soumettre l'ensemble des petits acteurs européens du logiciel, qui constituent l'essentiel du tissu économique dans le domaine, aux détenteurs de ces grands portefeuilles.

Déjà en 1991, un Bill Gates très lucide disait:

If people had understood how patents would be granted when most of today's ideas were invented and had taken out patents, the industry would be at a complete standstill today...A future start-up with no patents of its own will be forced to pay whatever price the giants choose to impose."

si les gens avaient compris le système des brevets à l'époque où les idées d'aujourd'hui ont été inventées, l'industrie serait complètement immobilisée maintenant. Une start-up de demain sans brevet sera forcée de payer n'importe quel prix que les géants choisiront de lui imposer.

Des pans entiers de l'industrie logicielle ont des modèles économiques qui font complètement abstraction du brevet: à ses tout débuts, Microsoft n'a pas breveté MS-DOS qui allait être le piédestal de sa future fortune. Plus récemment, la plupart des grands acteurs de l'Internet, comme Amazon, e-Bay, Google, ou Facebook ne se sont pas développés grâce à des brevets<sup>15</sup>, mais grâce à des innovations rapidement mises sur le marché. De même, l'industrie des jeux vidéo (pour la partie logicielle) construit ses succès sur la créativité et la vitesse de développement et ne mise pas du tout sur les brevets.

En revanche, dans le domaine des systèmes embarqués pour lequel les innovations sont des *objets* qui mêlent étroitement une architecture matérielle et du logiciel embarqué conçu spécifiquement pour celle-ci (même si ce logiciel est bâti avec des briques génériques ouvertes et standard), la protection par brevet peut être beaucoup plus pertinente, voire indispensable.

Quoi qu'il en soit, il paraît sage de rééquilibrer le poids donné aux brevets dans l'évaluation de la recherche en informatique en mettant beaucoup plus l'accent sur l'exploitation que sur le dépôt. Un portefeuille de brevets contenant peu ou pas de brevets exploités sur une durée significative, n'est probablement pas un indicateur d'excellence scientifique.

**Proposition n°6 :** Toutes les fois où le brevet est utilisé comme mesure de l'innovation ou comme instrument de valorisation, seuls devraient compter les brevets réellement exploités, et ce en fonction de l'importance économique de cette exploitation.

Par ailleurs, dans les formations dispensées aux informaticiens et aux économistes d'entreprise, il serait bon de montrer à partir d'étude de cas que l'innovation réussie en logiciel ne se fait presque jamais via des brevets.

**Proposition n°7 :** Améliorer les formations sur la valorisation dispensées aux scientifiques, aux économistes et aux hauts fonctionnaires, en traitant spécifiquement du cas du logiciel.

---

<sup>15</sup>Même si Google a déposé dès sa création un brevet défensif sur la technique de *ranking* des résultats de son moteur de recherche, les observateurs s'accordent à dire que ce brevet n'a joué aucun rôle réel dans le succès de l'entreprise dû surtout à la rapidité d'action dans le déploiement d'une architecture virtualisée "cloud" qui elle n'était pas brevetée !

### ***Rendre le droit d'auteur aussi incitatif que le brevet***

Dès lors que le droit d'auteur est le meilleur outil pour protéger l'innovation dans l'industrie du logiciel, il est nécessaire que son statut, fiscal notamment, soit aussi favorable que celui applicable aux brevets.

**Proposition n°8** : Faire une étude comparative des régimes fiscaux du droit d'auteur portant sur le logiciel et du brevet, afin de rendre le droit d'auteur aussi attractif et aussi incitatif que le brevet.

Une proposition dans ce domaine peut être immédiatement formulée, à titre d'exemple.

L'article 244 quater B du code général des impôts, qui détermine l'assiette et les modalités de calcul du crédit d'impôt en faveur de la recherche, permet d'inclure dans cette assiette les dépenses « *de prise et de maintenance de brevets* » (I (e)) ainsi que « *les frais de défense de brevets [...], ainsi que dans la limite de 60 000 euros par an, les primes et cotisations ou la part des primes et cotisations afférentes à des contrats d'assurance de protection juridique prévoyant la prise en charge des dépenses exposées, à l'exclusion de celles procédant d'une condamnation éventuelle, dans le cadre de litiges portant sur un brevet [...] dont l'entreprise est titulaire ;* » (I (e bis)). Si le paragraphe I alinéa (e) n'est évidemment pas transposable en matière de droit d'auteur, le paragraphe I alinéa (e bis) l'est totalement. Ceci permettrait aux éditeurs de logiciel français de déduire d'une partie de leurs impôts le coût de certaines assurances dommage professionnelles spécialisées pour leur secteur d'activité, comme celle que le Syntec propose à ses adhérents. Cela leur permettrait également de voir d'un œil plus serein la prise en charge des frais juridiques dans des pays connus pour leur caractère élevé, comme les États-Unis ou le Royaume-Uni. À ce titre, une telle mesure favoriserait l'expansion à l'étranger, et notamment sur le marché américain, des entreprises françaises.

## **6.2 Les limitations du financement et du capital-risque**

La simplicité apparente du brevet est telle que le nombre de brevets déposés (et non pas, très étonnamment, celle des brevets exploités) est aujourd'hui inclus dans les indicateurs de production scientifique des chercheurs et des instituts de recherche. Il est aussi possible de valoriser dans les comptes d'une entreprise un brevet comme un actif, même s'il n'est pas exploité et qu'il génère des coûts récurrents de maintien non négligeables. Tout naturellement, certains investisseurs demandent donc à trouver des brevets dans les dossiers de financement comme éléments concrets de valorisation de la start-up.

Comme le brevet n'est pas particulièrement adapté au logiciel (au moins dans la plupart des cas), le fait que le capital-risque français comprenne peu la valorisation en dehors des brevets est une des raisons pointées en particulier par la filière du jeu vidéo, où tout le monde reconnaît que la vitesse est cruciale et le brevet ne joue aucun rôle, pour expliquer en partie les difficultés de financement rencontrées.

Plusieurs autres limitations sérieuses du financement actuel de l'innovation dans l'industrie du logiciel ont été pointées à plusieurs reprises par les experts :

- le volume de capital-risque disponible pour une start-up en France est nettement inférieur par rapport à celui qu'une start-up américaine peut lever ; cela s'explique en partie comme une conséquence de la taille relative des deux pays et donc des marchés cibles primaires de la jeune entreprise, mais on fait aussi observer que des pays bien plus petits que la France, comme Israël, ont réussi à démultiplier le capital-risque par une politique de prise de participations publiques spécialement étudiée pour avoir un effet de levier significatif ;

- la réticence à prendre des risques : le capital-risque américain accepte de financer des startups en s'attendant à un taux de succès de 1 sur 10. Le capital-risque européen semble (sauf exception) beaucoup plus frileux. Une des conséquences de cette frilosité, pointée lors des entretiens, est la tendance récurrente des investisseurs institutionnels à chercher à se rassurer en finançant des doublons de succès qui ont déjà fait leur preuve ailleurs : Google ou Facebook n'auraient probablement pas levé en France le capital nécessaire pour se développer quand personne ne les connaissait, mais on compte un certain nombre de financements octroyés pour développer, par exemple, "le Google européen", tentatives naïves évidemment vouées à l'échec.
- La complexité du mécanisme de financement de la R&D de l'innovation : la faiblesse du financement privé de l'innovation en France étant bien connue, l'effort d'innovation et de recherche est soutenu en Europe en général et en France en particulier majoritairement par des financements publics. Or, le financement public a, en principe, des contraintes bien plus strictes qu'un fonds de capital-risque : il est soumis à l'encadrement communautaire des aides publiques aux entreprises, l'argent public se doit d'être "bien dépensé" et ne peut prendre de risque injustifié, d'où le besoin d'expertises multiples et, par exemple, des contraintes sur les fonds propres des startups qui participent à des projets collaboratifs. Si ces contraintes sont naturelles quand on veut réduire le risque, elles peuvent avoir pour conséquence de rendre inéligibles des très petites entreprises porteuses d'innovations de rupture dans le logiciel qui démarrent avec des fonds propres insignifiants. Or, on est souvent dans le domaine du « Winner take it all » où être le premier à installer une technologie innovante est crucial (voir les moteurs de recherches, les réseaux sociaux, etc.).

**Proposition n°9 :** Étudier la possibilité de moduler les contraintes sur les fonds propres imposées aux jeunes pousses pour accéder au financement public de l'innovation, éventuellement en contrepartie d'engagements permettant d'assurer que le logiciel développé et l'expertise afférente ne seront pas perdus en cas de défaillance de l'entreprise.

- un « trou » dans le soutien de la croissance des startups: au démarrage, le financement public est très attractif pour les startups qui arrivent à réunir suffisamment de fonds propres, et en haut de l'échelle les grands groupes semblent savoir jongler avec la complexité du système pour optimiser leur fiscalité ; mais il a été fait remarquer qu'au milieu, là où sont les moyennes entreprises qui font la force du système allemand, il y a une zone où les entreprises subissent d'un coup un alourdissement de charges très important, qui peut freiner leur croissance ; on a remarqué que cela pourrait expliquer pourquoi de nombreuses startups sont rapidement revendues, les investisseurs initiaux semblant peu motivés pour accompagner leur croissance ; mais on a aussi observé que la phase de croissance nécessite des personnes aux qualités différentes par rapport à la phase initiale.

**Proposition n° 10:** Étudier les moyens d'éviter les effets de seuil dans les charges pesant sur les entreprises du logiciel.

Les difficultés auxquelles l'industrie du logiciel fait face de ce fait sont donc surtout des difficultés qui tiennent à la culture économique de la France. Toute évolution de ce point de vue ne pourra donc être que progressive, même si les propositions n°5 et n°6 peuvent contribuer à créer, à l'échelle d'une génération, chez les futurs gestionnaires de fonds de capital-risque une conscience de l'intérêt que présente l'informatique et des enjeux spécifiques de l'industrie du logiciel.

Enfin, ils existent des composants logiciel libre d'utilité générale, qui auraient intérêt à être maintenus par des structures soustraites aux aléas de la compétition économique, et fournir un socle

solide non seulement de code, mais aussi de compétences, si une communauté se forme autour d'eux entre développeurs, chercheurs et utilisateurs.

L'importance économique de ces composants génériques libres a déjà été notée dans le rapport "Briques Génériques du Logiciel Embarqué", mais cette valeur ne se limite pas à ce domaine.

Une possible telle structure peut être une Fondation sur le modèle des fondations Apache, Mozilla ou Linux, dont la gouvernance et le modèle de financement nécessitent une étude économique approfondie. L'État doit prendre l'initiative de cette étude et doit ensuite définir une incitation si l'effet de levier est avéré.

**Proposition n. 11 :** Étudier la possibilité de créer une Fondation d'utilité publique pour soutenir le développement des logiciels libres d'intérêt général

### 6.3 Des modes de financement des infrastructures de recherche peu adaptés au logiciel

Dans l'effort actuel de financement de la recherche, certains instruments ont été conçus pour soutenir des investissements importants dans des infrastructures collectives en se basant sur le modèle issu des grandes installations de la physique ou de l'astrophysique. L'État alloue des crédits exceptionnels, parfois considérables pour financer la construction d'un grand équipement de recherche (un accélérateur de particules, un télescope, ...) laissant aux "opérateurs de la recherche" (CNRS notamment) le soin de recruter et d'affecter le personnel qui va l'utiliser.

Ce modèle d'intervention de l'État est inadapté pour financer les "forges à logiciel" modernes qui sont indispensables aux grands développements collaboratifs des technologies logicielles de demain. En effet, la part principale de coût de ces infrastructures est le coût des personnels. En particulier, elles ne trouvent leur pleine efficacité que si des ingénieurs de recherche en nombre suffisant y sont affectés, permettant de finaliser et professionnaliser les prototypes logiciels, de maintenir les outils et plateformes logicielles de développement et de former les utilisateurs nouveaux entrants.

L'essor du *Cloud Computing* permettrait par exemple de construire des forges ou des centres de données mutualisés qui réduiraient énormément les coûts et augmenteraient l'efficacité du développement, mais des initiatives de ce genre n'entrent pas dans la définition des infrastructures de recherche actuellement financées, les montants d'investissement étant trop faible, et la part de fonctionnement trop importante.

Les forges existantes fonctionnent plus ou moins en vase clos. La forge du CRU <sup>16</sup> accepte par exemple la création de projets venant de l'ensemble des universités, mais elle refuse les projets étudiants et fonctionne à l'intérieur du monde universitaire, sans ouverture réelle sur le monde des entreprises. La forge de l'INRIA est une des plus significatives de France, et elle rend un service appréciable aux équipes propres ou associées à cet institut.

Le besoin de créer une forge au niveau du CNRS a été exprimé formellement <sup>17</sup> et une proposition de réfléchir à une forge au niveau enseignement supérieur et recherche existe aussi <sup>18</sup>.

Ces outils existants et ces projets ne prennent pas (ou imparfaitement) en compte les besoins d'articulation entre le monde académique et celui des entreprises pour faire éclore les innovations technologiques logicielles ; ils ont aussi tendance à oublier que le développement collaboratif est un besoin qui ne se limite pas au code, mais se retrouve aussi dans l'écriture de documents techniques.

L'innovation logicielle prenant place dans des "écosystèmes" décloisonnés, ces forges doivent être accessibles à tous les acteurs d'un développement innovant, chercheurs d'organismes différents,

<sup>16</sup><http://www.cru.fr/faq/sourcesup/index>

<sup>17</sup><http://www.dgdr.cnrs.fr/drh/omes/documents/pdf/etude-metiers-bap-e.pdf>

<sup>18</sup><http://www.projet-plume.org/fr/ressource/projet-forge-esr>

PME partenaires, pôles de compétitivité, etc.

Laisser le soin aux opérateurs (CNRS, INRIA, CEA, universités) de déployer séparément de telles infrastructures sans que celles-ci ne soient mutualisées est donc contre-productif.

Il serait très préférable d'avoir une structure mutualisée offrant des « vues » spécialisées par types de développements (par ex.: mondes virtuels 3D et jeux, internet d'objets, etc.) ayant des besoins différents, mais ouvertes à tous les acteurs concernés, et qui puissent :

- héberger non seulement le développement collaboratif du code, mais aussi de la documentation technique associée, voire même l'écriture collaborative d'articles scientifiques dans le domaine ;
- prévoir explicitement l'hébergement de projets étudiants, et leur possible évolution dans le temps vers un cadre plus institutionnel ou international ;
- prévoir explicitement un lien et un échange de métadonnées avec les autres forges, notamment industrielles ou internationales ;
- permettre de suivre l'activité des contributeurs dans le temps, ce qui peut aider les étudiants à constituer facilement un *bilan objectif de leurs compétences logicielles* pour leur curriculum.

La mutualisation ne vise pas seulement à partager les coûts des investissements matériels (assez modestes si l'on utilise des serveurs virtualisés), mais surtout à mutualiser ces compétences qui à terme forment une part essentielle de la valeur de ces plateformes communes. Dans cet esprit, il nous paraît souhaitable que l'alliance ALLISTENE, en liaison avec les pôles de compétitivité, organise une réflexion sur la création et le fonctionnement d'une ou plusieurs forges nationales, fédératives et ouvertes à tous les acteurs concernés, et la mise en place, indispensable, d'un programme de recherche multidisciplinaire sur les environnements collaboratifs qui guide l'évolution dans le temps de ces forges.

**Proposition n°12 :** Confier à ALLISTENE, en liaison avec les pôles de compétitivité, une mission de préfiguration d'infrastructures de développement collaboratif, adaptées à un type de développement particulier, et dotées des moyens nécessaires en personnel pour mener leurs activités de manière pérenne.

#### **6.4 Un cadre à clarifier pour les activités "valorisation" des enseignants-chercheurs**

Les réflexions sur la modernisation des Universités et sur la carrière des enseignants-chercheurs ont souvent fait l'observation que les activités d'enseignement ne sont pas assez prises en compte et valorisées. Il est probable qu'un nouveau docteur recruté vers 2015 pourra faire évoluer sa carrière d'enseignant-chercheur avec une relative souplesse entre les fonctions de recherche, d'enseignement et d'administration universitaires.

Mais très peu est fait pour prendre en compte les activités des enseignants chercheurs qui font l'effort de créer des ponts avec les entreprises. Dans l'évaluation de son travail, les activités de valorisation de la recherche et de conseil en entreprise sont non seulement non reconnues, mais constituent même plutôt un handicap.

De plus, il a été observé que bon nombre d'enseignants-chercheurs en Informatique ne sont pas informés des multiples opportunités de collaboration industrielle qui leur sont ouvertes par les textes actuels, et notamment par le Code de la recherche, article L413-1 et s., les Décrets 91-267 du 6 mars 1991, 99-1081 du 20 décembre 1999, 2001-125 du 6 février 2001 et 2007-611 du 26 avril 2007.

**Proposition n°13:** Diffuser largement et régulièrement l'information sur les possibilités offertes par les textes actuels aux enseignants-chercheurs pour collaborer avec l'industrie.

Aussi, on remarque que les démarches administratives, fiscales et sociales lourdes traditionnellement nécessaires pour réaliser des activités de consultance ne facilitent pas la collaboration ponctuelle avec des partenaires industriels, pourtant indispensable pour permettre les premiers contacts. La création du régime d'auto-entrepreneur est une réponse partielle à ce problème.

Cependant, il existe toujours une incohérence dans les dispositions régissant le cumul d'emploi des fonctionnaires. Si la réalisation d'œuvres de l'esprit est libéralement consentie aux fonctionnaires, dont les enseignants chercheurs, ceux-ci, s'agissant de logiciel, ne peuvent facilement en tirer profit. En effet, là où un ouvrage scientifique peut être facilement exploité, par exemple via un contrat d'édition avec un éditeur, un logiciel a besoin d'être industrialisé pour être commercialement exploitable. Comme ces activités d'industrialisation ne sont pas des activités de recherche, elles ne peuvent être menées par les établissements publics. Une entreprise pourrait le faire, mais un fonctionnaire ne peut durablement être associé à un rôle de consultance ou de direction scientifique dans une entreprise privée (art. 25 I al 2 de la loi du 13 juillet 1983).

Dès lors, les enseignants chercheurs en Informatique sont condamnés à faire un choix si leur technologie présente des perspectives d'exploitation commerciale.



## 7- Résumé des Propositions d'actions

Étant donnée la difficulté de prévoir l'évolution technologique et les marchés émergents sur un secteur à l'évolution si rapide, le groupe de travail considère que les politiques publiques visant à favoriser le développement de l'industrie du logiciel en France et en Europe doivent être articulées autour de deux principes complémentaires :

- favoriser le développement et le fonctionnement de l'*écosystème de l'innovation logicielle* dans ses dimensions humaines (formation, mobilité...), organisationnelles, financières (investissements, fiscalité, ...) et techniques (forges à logiciel). Nos six premières propositions concernent les différents volets de ce principe d'action.
- mettre en œuvre une politique industrielle volontariste dans les domaines à fort potentiel pour lesquels des acteurs nationaux disposent d'atouts réels. Notre septième proposition porte spécifiquement sur ce point.

### 7.1 Améliorer l'image et l'attractivité des métiers de l'informatique

Les métiers de l'informatique ne sont pas suffisamment attractifs en France, et souffrent d'une image brouillée et d'un manque de "role models". À cela s'ajoute un contexte peu favorable en général aux études scientifiques, et un déficit important de femmes dans l'ensemble de la filière TIC. Selon Neelie Kroes, Commissaire européenne en charge de la société numérique, le secteur européen des TIC aura besoin de trouver 700 000 nouveaux professionnels d'ici à 2015 <sup>19</sup>.

Ce manque est en partie dû à la faible proportion de femmes ingénieures ou diplômées en informatique ; dans la recherche en France, la situation est très déséquilibrée : à titre d'exemple, dans la section 07 du CNRS, les femmes ne sont que 19% <sup>20</sup>. Il y a là un problème grave, qui ne peut être abordé en profondeur dans ce rapport : les propositions qui suivent, en rendant l'Informatique plus attractive en générale, pourraient attirer mécaniquement plus de femmes dans ces métiers, mais il est nécessaire de procéder à une étude approfondie de cette question.

Pour améliorer l'image des métiers de l'informatique, nous avons longuement évoqué au chapitre 4 l'importance de reconnaître la valeur de la programmation, et au chapitre 5 l'importance de l'introduction de l'informatique comme discipline à part entière dans le cursus des lycéens des sections scientifiques, ainsi que la généralisation de la formation à l'informatique pour toutes les disciplines du supérieur.

**Proposition n°2 :** Faire de l'informatique une discipline à part entière dans l'enseignement secondaire, et créer les corps d'enseignants, les concours de recrutement et les corps d'inspection correspondants.

**Proposition n°3 :** Introduire de manière systématique l'enseignement de l'informatique dans les cursus d'enseignement supérieur non informatiques.

**Proposition n°4 :** Favoriser et accompagner les initiatives de « main à la pâte » dans l'Informatique, sous la forme de concours de programmation à tous les niveaux.

Il faut aussi valoriser les succès réels de l'industrie du logiciel française, en communiquant régulièrement sur les réussites qui permettent aux jeunes de trouver des "role models" : un effort

<sup>19</sup> <http://www.lemondeinformatique.fr/actualites/lire-journee-de-la-femme-neelie-kroes-pointe-l-inegalite-des-genres-dans-l-it-33091.html>

<sup>20</sup> <http://bilansocial.dsi.cnrs.fr/pdf/bilan-social-2009-partie1.pdf>

régulier de communication dans ce sens doit être fait, associant les opérateurs publics de la recherche, les entreprises innovantes et les pôles de compétitivité.

## 7.2 Améliorer la formation "multidisciplinaire" des informaticiens

Au chapitre 5, nous avons aussi vu l'importance d'élargir la formation des informaticiens à des disciplines qui deviennent nécessaires pour l'innovation.

**Proposition n°5 :** Modifier les cursus spécialisés en informatique pour y intégrer des approches multidisciplinaires, qui tiennent compte des apports de disciplines connexes, comme l'ergonomie, l'économie, les mathématiques appliquées, la gestion de projet, les techniques de validation et de certification du logiciel, de contrôle qualité, etc.

Cela demande d'intervenir à plusieurs niveaux, par exemple :

- encourager les universités à créer des écoles doctorales dans des domaines à cheval entre deux disciplines, et relevant par conséquent de deux sections différentes de la CNU : biologie et informatique, informatique et urbanisme, informatique électronique et optique, etc. en offrant aux doctorants des formations croisées
- favoriser l'ouverture disciplinaire des informaticiens vers des compétences complémentaires : s'il est essentiel qu'un socle de connaissances basé sur l'informatique soit au centre de la formation, il est important que les personnes disposant de solides bases techniques puissent aussi disposer de compétences complémentaires, par exemple en design, économie, sciences humaines, gestion de projets et de communautés, etc.

## 7.3 Encourager les enseignants-chercheurs à participer à des projets d'innovation

Pour libérer le potentiel de collaboration des enseignants chercheurs en informatique avec le reste de l'écosystème, il faut prendre en compte les activités nouvelles qui leur sont demandées: la collaboration industrielle, le transfert technologique, le développement et maintenance de logiciels innovants de qualité doivent être considérés comme de véritables fonctions. Les schémas d'évaluation de l'AERES et de l'Université doivent considérer explicitement la production de code de qualité (dès lors que celui-ci est diffusé et utilisé au-delà du cercle des chercheurs), et les collaborations avérées avec des industriels, au même titre que des publications prestigieuses.

**Proposition n°1 :** Prendre en compte explicitement la production logicielle de qualité des enseignants-chercheurs dans l'évaluation de leur activité scientifique.

En parallèle, il faut informer les personnes sur les dispositifs existants permettant les collaborations industrielles.

**Proposition n°13 :** Diffuser largement et régulièrement l'information sur les possibilités offertes par les textes actuels aux enseignants-chercheurs pour collaborer avec l'industrie.

Il est aussi nécessaire de faire en sorte que les connaissances techniques des enseignants chercheurs en informatique ne se trouvent pas aspirées par des tâches qui relèvent soit de la gestion courante de l'infrastructure informatique de leur établissement ou de l'ingénierie des prototypes de recherche.

**Proposition n°1 bis** Des ingénieurs de développement et de recherche en nombre suffisant doivent travailler avec les chercheurs pour accélérer l'ingénierie des prototypes de recherche et maintenir les plateformes et outils de développement.

En comparaison avec d'autres disciplines, l'informatique est très mal dotée.

#### 7.4 Améliorer les mécanismes de financement de l'innovation logicielle

Beaucoup a déjà été fait en France et en Europe pour financer les projets innovants.

Dans le domaine du logiciel, plusieurs axes de progrès sont toutefois possibles ; tout d'abord, il faut reconnaître plus largement les spécificités de la valorisation du logiciel, basé principalement sur le droit d'auteur et non sur les brevets, alors que ces derniers sont mieux connus et reconnus par les acteurs de la valorisation.

**Proposition n°6 :** Toutes les fois où le brevet est utilisé comme mesure de l'innovation ou comme instrument de valorisation, seuls devraient compter les brevets réellement exploités, et ce en fonction de l'importance économique de cette exploitation.

**Proposition n°7 :** Améliorer les formations sur la valorisation dispensées aux scientifiques, aux économistes et aux hauts fonctionnaires, en traitant spécifiquement du cas du logiciel.

**Proposition n°8 :** Faire une étude comparative des régimes fiscaux du droit d'auteur portant sur le logiciel et du brevet, afin de rendre le droit d'auteur aussi attractif et aussi incitatif que le brevet.

Ensuite, il est important de libérer le potentiel des petites entreprises innovantes en Informatique, qui ont souvent des coûts initiaux très réduits, et d'accompagner leur croissance.

**Proposition n°9 :** Étudier la possibilité de moduler les contraintes sur les fonds propres imposées aux jeunes pousses pour accéder au financement public de l'innovation, éventuellement en contrepartie d'engagements permettant d'assurer que le logiciel développé et l'expertise afférente ne seront pas perdus en cas de défaillance de l'entreprise.

**Proposition n° 10:** Étudier les moyens d'éviter les effets de seuil dans les charges pesant sur les entreprises du logiciel.

Enfin, il peut être utile de fournir un cadre précis pour maintenir les composants logiciels libres d'intérêt général, en particulier quand ils sont issus de la recherche sur fonds publics.

**Proposition n. 11 :** Étudier la possibilité de créer une Fondation d'utilité publique pour les logiciels libres d'intérêt général

#### 7.5 Mettre en place une infrastructure technique mutualisée

Les acteurs de la formation, la recherche et de l'innovation en informatique développent des logiciels de natures différentes, mais partagent tous le besoin de disposer d'une infrastructure permettant un développement collaboratif efficace et moderne, avec des outils de contrôle de versions, de documentation technique, d'organisation et de suivi de projet.

**Proposition n°12 :** Confier à ALLISTENE, en liaison avec les pôles de compétitivité, une mission de préfiguration d'infrastructures de développement collaboratif, adaptées à un type de développement particulier, et dotées des moyens nécessaires en personnel pour mener leurs activités de manière pérenne.

## 7.6 Construire des indicateurs pertinents

Le SYNTEC Numérique d'une part, et l'OPIIEC d'autre part publient des données précises sur la sociodémographie des entreprises du secteur, les qualifications des personnels, les niveaux de rémunération recommandés par les accords de branche, etc. Le groupe de travail est unanime sur le besoin de compléter cette batterie d'indicateurs par des données économiques sur le secteur permettant de mieux comprendre l'évolution de l'industrie du logiciel en France. En particulier, il serait utile de disposer de :

- un bilan détaillé des importations et exportations de logiciels, qui permettent de connaître les points forts et les points faibles de l'offre logicielle en France ;
- un recensement des activités autour du logiciel qui sont fiscalement délocalisées et dont les revenus sont comptabilisés dans des pays aux fiscalités avantageuses pour la propriété intellectuelle, comme l'Irlande ou l'Espagne. Il serait paradoxal que des entreprises pratiquant une telle délocalisation fiscale reçoivent par ailleurs des subventions publiques pour leurs activités de R&D en logiciel en France;
- un bilan des créations d'entreprises autour du logiciel, et de leur évolution: quelle est leur durée de vie, que deviennent-elles (rachat, transformation en entreprises moyennes ou grandes) ;
- des éléments de mesure de l'efficacité de l'investissement public : en cas de rachat d'une jeune pousse par une entreprise, en particulier internationale, quelle est la part des subventions publiques qui ont créé des emplois et de la valeur sur le territoire, et quelle est la part qui a simplement été absorbée par des investisseurs étrangers ?
- un bilan des étudiants diplômés qui sortent d'une formation initiale en informatique : quels sont leurs profils, quels secteurs industriels les recrutent, quelle est la durée moyenne d'un premier contrat, quelle est la part de création de nouvelles activités ? Ces données sont disponibles pour les jeunes docteurs, et devraient être complétées pour les diplômés des niveaux de B+2 à B+5.

## 7.7 Investir dans les domaines scientifiques et techniques des TIC à fort potentiel

Il paraît évident, en considérant la masse critique de la France sur la scène mondiale, qu'il faut concentrer les *financements finalisés* sur quelques domaines industriels à haute valeur ajoutée *pour lesquels on dispose de compétences et d'une avance avérées et dans lesquels il n'y a pas encore d'acteur mondial dominant*, en évitant d'investir sur des entreprises qui s'épuisent à courir derrière des leaders américains ou asiatiques détenant déjà des positions dominantes. Ce critère devrait notamment être pris en compte pour les "investissements d'avenir" en cours de sélection.

Le ministère de l'Industrie vient de publier le rapport *Technologies clés 2015 85 technologies clés dans sept secteurs économiques* (section TIC)<sup>21</sup> qui présente un panorama assez large des enjeux sociétaux et technologiques des TIC dans les quelques années à venir, et peut fournir une base de travail. Si l'on se restreint aux domaines directement connexes au logiciel, on trouve dans ce rapport des éléments permettant d'étayer une avance française dans la mise en place de solutions de numérisation du patrimoine culturel, une expertise unique en systèmes embarqués

21 <http://www.industrie.gouv.fr/tc2015/index.php>

complexes (en particulier pour l'aérospatiale et le militaire, mais qui pourrait trouver rapidement d'autres débouchés), avec tout ce que cela comporte en modélisation et métamodélisation, simulation, visualisation et calcul parallèle, un environnement favorable pour le logiciel libre et les architectures orientées services.

Les membres de ce groupe de travail soulignent aussi l'impact de l'informatique ubiquitaire, du Cloud Computing, et des grandes masses de données, qui se généralisent aujourd'hui.

La combinaison de ces trois dynamiques porte des problèmes nouveaux.

À titre d'exemple, l'essor du 'tout connecté' et la multiplication des appareils mobiles a besoin d'avancées significatives dans le co-design matériel/logiciel, dans la conception de systèmes économes en énergies, la prise en compte de la complexité apportée par la mise en réseau de logiciels embarqués qui n'avaient pas forcément été conçus pour cela, en terme de sécurité et sûreté. Évidemment, les domaines de recherche couvrant la certification, le test, la preuve et la qualité de code sont essentiels dans cette perspective, mais c'est aussi le cas des nouvelles approches à la conception et programmation de systèmes complexes, ainsi que les techniques visant à exploiter les systèmes multicœur, qui vont permettre de produire plus rapidement des applications adaptées aux nouveaux besoins sociétaux.

La généralisation des infrastructures pour le cloud computing rend nécessaire la stabilisation d'une couche middleware ouverte, servant de base à des services à forte valeur ajoutée, permettant de la programmation distribuée rapide, avec des langages de haut niveau, qui peut être fournie en logiciel libre. Plus généralement, il y a une exigence transverse de sécurité et de garantie de la confidentialité des données qui est accrue par la dématérialisation des services et des données, et par la mise en réseau de systèmes divers et disparates : des avancées scientifiques sont nécessaires pour dépasser l'approche basée sur la contractualisation de la qualité de service qui ne permet pas d'obtenir des garanties fortes.

Enfin, la mise à disposition de grandes masses de données, couplée avec la disponibilité d'une grande puissance de calcul flexible venant du Cloud Computing ouvrent les portes à des applications nouvelles, qui ont besoin d'outils avancés pour la gestion intelligente de grandes masses d'informations, comme la traduction automatique de la parole, ou la classification, indexation et extraction en temps réel d'information audio et vidéo.

Afin d'orienter les programmes de recherche, il sera nécessaire de croiser ces éléments avec les grands défis sociétaux, dont l'identification dépasse largement la mission de ce groupe de travail.

On tient cependant ici à souligner l'importance de maintenir, ou augmenter, le soutien aux structures légères de collaboration scientifique au niveau national, comme les GDR, qui assurent à faible coût les échanges entre les différentes équipes.

## 8- Conclusion

Nous souhaitons en conclusion souligner que l'on ne pourra pas relever ces défis, et espérer garder une position importante dans le panorama de l'industrie du logiciel mondiale sans une politique volontariste en terme de ressources humaines : à la différence d'autres disciplines, l'informatique a peu besoin d'équipements scientifiques lourds et coûteux, mais de personnes compétentes et motivées auxquelles on donne des conditions de travail correctes. Il faut tout simplement plus d'informaticiens compétents (techniciens et ingénieurs de l'industrie, enseignants, chercheurs) que ceux dont on dispose aujourd'hui pour accompagner le changement sociétal qui est en cours.

Les capacités de formation de jeunes techniciens et ingénieurs en informatique en France sont insuffisantes pour répondre aux besoins des industries du secteur et des industries utilisatrices. Il nous paraît donc indispensable de maintenir les efforts budgétaires pour recruter des enseignants-chercheurs en informatique dans les écoles d'ingénieurs et dans les universités, et d'accroître le nombre de candidatures de qualité en augmentant l'attractivité des carrières et des chaires d'enseignants-chercheurs pour les chercheurs confirmés français et étrangers.