



Nadia Eghbal

## Sur quoi reposent nos infrastructures numériques ? Le travail invisible des faiseurs du web

OpenEdition Press

---

# Comment la gratuité des logiciels a transformé la société

---

DOI : 10.4000/books.oep.1811  
Éditeur : OpenEdition Press, Framabook  
Lieu d'édition : OpenEdition Press,  
Framabook  
Année d'édition : 2017  
Collection : Encyclopédie numérique  
ISBN électronique : 9782821894938



<http://books.openedition.org>

### Référence électronique

EGHBAL, Nadia. *Comment la gratuité des logiciels a transformé la société* In : *Sur quoi reposent nos infrastructures numériques ? Le travail invisible des faiseurs du web* [en ligne]. Marseille : OpenEdition Press, 2017 (généré le 26 octobre 2017). Disponible sur Internet : <<http://books.openedition.org/oep/1811>>. ISBN : 9782821894938. DOI : 10.4000/books.oep.1811.

---

# COMMENT LA GRATUITÉ DES LOGICIELS A TRANSFORMÉ LA SOCIÉTÉ

La première réflexion qui vient à l'esprit est : « Pourquoi ces développeurs ont-ils rendu leur logiciel gratuit ? Pourquoi ne pas le faire payer ? » Les arguments en faveur du logiciel public reposent sur sa riche histoire politique et sociale. Mais d'abord, regardons la vérité en face : notre société n'en serait pas là où elle est aujourd'hui si des développeurs n'avaient pas rendu le logiciel libre et gratuit.

## **Avec le logiciel libre, la production de logiciel est plus simple et considérablement moins chère**

Uber, un service de transport de personnes, a annoncé récemment que des développeurs avaient créé un système permettant de réserver une voiture en utilisant Slack (une application de développement collaboratif) et non l'application mobile Uber. Le projet a été bouclé en 48 heures par une équipe de la App Academy, une école de programmation.

Uber a constaté<sup>1</sup> que l'équipe avait été capable d'achever le projet rapidement, car elle « avait utilisé des bibliothèques ouvertes telles que Rails, Geocoder et Unicorn pour accélérer le développement tout en travaillant sur une base solide ».

En d'autres termes, la quantité de code que l'équipe a dû écrire par elle-même a été fortement réduite, car elle a pu utiliser des bibliothèques libres créées par d'autres.

1. Relaté dans un article de blog : Uber Developers, « Uber + Slack + Weekend. A story of open APIs », Medium.com, 10/11/2015.

Ruby Geocoder, par exemple, est une bibliothèque réalisée en 2010 et maintenue par Alex Reisner<sup>2</sup>, un développeur indépendant. Geocoder permet à une application de chercher facilement des noms de rues et des coordonnées géographiques.

Unicorn est un serveur datant de 2009, il est administré par une équipe de sept contributeurs<sup>3</sup> encadrés par Eric Wong, un développeur.

Créer un nouveau logiciel n'a jamais été aussi simple, car il existe de plus en plus de portions de code « prêtes à l'emploi » dont on peut se servir. Pour en revenir à la métaphore de l'entreprise de bâtiment, il n'est plus nécessaire pour construire un immeuble de fabriquer soi-même tout ce dont on a besoin, il est plus simple d'acheter du « préfabriqué » et d'assembler fondation, structure porteuse et murs comme des legos.

Du coup, il est inutile de savoir comment construire un logiciel en partant de zéro pour être qualifié de développeur. Le service des statistiques sur le travail des États-Unis (Bureau of Labor Statistics<sup>4</sup>) estime que l'emploi des développeurs va augmenter de 22 % entre 2012 et 2022, soit bien plus rapidement que la moyenne dans les autres professions.

## Le logiciel libre est directement responsable de la renaissance actuelle des startups

Les coûts de lancement d'une entreprise ont considérablement diminué depuis la première bulle internet de la fin des années 1990. Le capital-risqueur et ex-entrepreneur Mark Suster évoquait son expérience dans un billet de blog de 2011<sup>5</sup> :

2. Une brève description d'Alex Reisner est visible sur son site personnel [alexreisner.com](http://alexreisner.com).

3. Pour connaître les noms des contributeurs d'Unicorn, voir leur site web [unicorn.bogomips.org](http://unicorn.bogomips.org).

4. Aux États-Unis, le Bureau of Labor Statistics fournit des données sur les différents métiers : activités, niveau de formation, salaire moyen, perspective de recrutement, etc. En janvier 2017, il estimait une augmentation de 17 % du nombre de développeurs entre 2014 et 2024 (et 7 % tout emploi confondu). Voir Bureau of Labor Statistics, U.S. Department of Labor, *Occupational Outlook Handbook*, 2016-2017 Edition, « Software Developers ».

5. Voir le billet de Mark Suster, « Understanding Changes in the Software & Venture Capital Industries », *Both Sides*, 28/06/2011.

Quand j'ai monté ma première entreprise, en 1999, l'infrastructure coûtait 2,5 millions de dollars, simplement pour commencer, et il fallait y ajouter 2,5 millions de dollars de plus pour payer l'équipe chargée de coder, lancer, gérer, démarcher et vendre notre logiciel.

Nous avons à peine perçu le premier changement d'ampleur dans notre industrie. Il a été porté par l'introduction du logiciel libre et plus précisément par ce que l'on a appelé la pile LAMP. Linux (au lieu de UNIX), Apache (un logiciel de serveur web), MySQL (à la place d'Oracle) et PHP. Il y a bien sûr eu des variantes – nous préférons PostgreSQL à MySQL et beaucoup de personnes utilisaient d'autres langages de programmation que PHP.

Le logiciel libre est devenu un mouvement, un état d'esprit. Soudain, les logiciels d'infrastructure étaient presque gratuits. Nous avons payé 10 % du tarif normal pour l'achat des logiciels et le reste de l'argent est allé dans le support. Un tel effondrement de 90 % des coûts engendre de l'innovation, croyez-moi.

La disponibilité actuelle des composants logiciels libres et gratuits (associée à des services d'hébergement moins chers comme Amazon Web Services et Heroku) permet à une *startup* technologique de se lancer sans avoir besoin de millions de dollars. Les entrepreneurs peuvent tout à fait sortir un produit et trouver un marché sans dépenser un seul dollar, la levée de fonds auprès de capital-risqueurs se faisant seulement après avoir prouvé la viabilité de leur projet.

Alan Schaaf, qui a fondé Imgur, un site populaire de partage d'images faisant partie des 50 sites les plus consultés au monde, a justement déclaré que les sept dollars nécessaires à l'achat du nom de domaine représentaient la seule dépense indispensable au démarrage de son entreprise. Imgur était rentable et, avant de lever 40 millions de dollars en 2014 auprès de l'entreprise de capital-risque Andreessen Horowitz, Schaaf n'a eu recours à

aucun fonds extérieur pendant cinq ans<sup>6</sup>. Les capital-risqueurs ainsi que les autres acteurs de l'investissement ont, à leur tour, commencé à investir des montants moindres, développant ainsi de nouvelles formes de fonds d'investissement dont voici trois exemples :

- **Fonds spécialisés dans le capital d'amorçage** : sociétés de capital-risque préférant financer la première levée de fonds, plutôt que de participer à une augmentation de capital ultérieure.
- **Fonds de micro capital-risque** : une définition assez large sous laquelle on regroupe les sociétés de capital-risque disposant de moins de 50 millions de dollars d'actifs.
- **Accélérateurs de *startup*** : des sociétés qui financent de petites sommes, souvent inférieures à 50 000 dollars, et qui conseillent et parrainent également les toutes jeunes entreprises...

Aujourd'hui, avec 10 millions de dollars, on peut financer cent entreprises contre seulement une ou deux dans les années 1990.

## Le logiciel libre a simplifié l'apprentissage de la programmation, rendant la technologie accessible à tous, partout dans le monde

Si aujourd'hui vous voulez apprendre à coder chez vous, vous pouvez commencer par étudier Ruby on Rails. Rails est le nom d'un *framework* et Ruby est un langage de programmation. Quiconque dispose d'un accès internet peut installer gratuitement ces outils sur n'importe quel ordinateur. Parce qu'ils sont libres et gratuits, ils sont également très populaires, ce qui signifie qu'il existe quantité d'informations en ligne permettant de bien démarrer, du simple tutoriel au forum d'aide. Cela montre qu'apprendre comment coder est aussi accessible que d'apprendre à lire et écrire l'anglais ou le français.

6. Voir Jessica Guynn, « Imgur scores \$40 million in funding from Andreessen Horowitz », *Los Angeles Times*, 03/04/2014.

Pour comparer, l'utilisation de *frameworks* et de langages non *open source* impliquaient : de payer pour y avoir accès, d'utiliser un système d'exploitation et des logiciels spécifiques et d'accepter des contraintes de licence susceptibles d'entraver le dépôt d'un brevet pour un logiciel construit sur la base de ce *framework*. Aujourd'hui, il est difficile de trouver des exemples de *frameworks* qui ne sont pas publics. L'un des plus célèbres exemples de *framework* propriétaire est le .NET, développé et sorti en 2002. En 2014, Microsoft a annoncé la sortie d'une version publique de .NET, appelée .NET Core.

Audrey Eschright, une développeuse, a décrit comment les logiciels *open source* l'ont aidée à apprendre la programmation à la fin des années 1990<sup>7</sup>.

Je voulais apprendre à programmer, mais je n'avais pas d'argent. Pas la version « étudiante fauchée » : ma famille était pauvre mais également dans une situation chaotique... Cela peut sembler étrange aujourd'hui, mais à l'époque il y avait en fait deux options pour quelqu'un qui voulait écrire de véritables logiciels : on pouvait utiliser un ordinateur sous Windows et payer pour les coûteux outils de développement de Microsoft, ou bien on pouvait avoir accès à un système Unix et utiliser le compilateur GCC. Mon but devint donc d'avoir accès à des systèmes Unix pour pouvoir apprendre à programmer et faire des trucs sympas.

Jeff Atwood, un développeur .NET de longue date, a expliqué sa décision d'utiliser Ruby pour un nouveau projet, *Discourse*, en 2013<sup>8</sup> :

Quand on habite en Argentine, au Népal ou en Bulgarie par exemple, il est vraiment très difficile de démarrer en programmation avec les outils fournis par Microsoft. Les systèmes d'exploitation, les langages et

7. Voir Audrey Eschright, « For Love and For Money », [lifeofaudrey.com](http://lifeofaudrey.com), blog personnel.

8. Voir Jeff Atwood, « Why Ruby ? », *Coding Horror: Programming and Human Factors*, blog personnel, 22/03/2013.

les outils *open source* permettent de mettre qui que ce soit au même niveau, ils constituent le socle sur lequel travaillera, partout dans le monde, la prochaine génération de programmeurs, celle qui nous aidera justement à changer le monde.

Le nombre de *startups* a explosé et dans leur sillage sont apparues beaucoup d'initiatives pour enseigner la programmation à la population : aux enfants et aux adolescents, mais aussi aux membres de communautés défavorisées, aux femmes ou aux personnes en reconversion professionnelle. Parmi ces initiatives, on retrouve Women Who Code, Django Girls, Black Girls Code, One Month et Dev Bootcamp.

Certaines de ces organisations offrent leurs services gratuitement, tandis que d'autres les font payer. Toutes reposent dans leur enseignement sur des logiciels libres et gratuits. Par exemple, Django Girls<sup>9</sup> a appris à coder à plus de 2 000 femmes dans 49 pays. Bien que l'organisation n'ait pas développé Django elle-même, elle a le droit d'utiliser ce logiciel, que les étudiantes téléchargent et utilisent gratuitement dans leur programme d'apprentissage.

Dev Bootcamp apprend à programmer aux personnes qui veulent changer de carrière, et prépare indifféremment tout individu, du professeur d'anglais au vétéran, à devenir développeur professionnel. Le programme coûte entre 12 et 14 000 dollars. Dev Bootcamp enseigne entre autres Ruby<sup>10</sup>, JavaScript, Ruby on Rails et SQL. Les étudiants peuvent télécharger et utiliser tous ces outils gratuitement, pour lesquels Dev Bootcamp n'a pas besoin de payer. La société Dev Bootcamp a été rachetée par Kaplan en 2014 pour une somme inconnue.

Si des logiciels aussi importants n'étaient pas gratuits, beaucoup seraient dans l'incapacité de participer à la renaissance technologique actuelle. Il existe encore une multitude d'obstacles économiques et sociaux qui empêchent qu'ils soient encore plus nombreux à participer, comme le prix

9. Voir [Djangogirls.org](http://Djangogirls.org).

10. Voir la présentation de Ruby sur [ruby-lang.org](http://ruby-lang.org).

du matériel nécessaire pour avoir un ordinateur portable et une connexion internet, mais les outils de programmation eux-mêmes ne coûtent rien.