

Préface

Il fut un temps où la qualité d'un code se mesurait à ses performances par rapport aux ressources utilisées. C'était l'époque où les capacités des processeurs, la quantité de mémoire vive ou les capacités de stockage constituaient de véritables limites qui étaient prises en compte. Un temps où les ordinateurs prenaient de la place sur le bureau et y restaient longtemps (dans la mémoire du porte-monnaie aussi). Une époque où l'on reconnaissait un bon développeur à ses aptitudes à faire rentrer son code dans la petite zone allouée. Ces choses étaient précieuses. C'était il y a quelques dizaines d'années.

Et puis, alors que le changement climatique est devenu une réalité quotidienne dans la vie de millions d'humains, alors que les métaux rares et autres ressources non renouvelables sont à l'origine de conflits géopolitiques, alors que les enseignements scolaires de nos pays mettent en avant les questions du développement durable et prônent notre soi-disant « exemplarité » en matière de réduction de gaz à effet de serre, on assiste tout simplement à un engouement pour un développement logiciel où l'efficacité se mesure en temps de travail pour l'informaticien et non plus en efficacité par rapport aux ressources. Et tant pis si cette précipitation se traduit par des logiciels toujours plus gourmands, puisque les ressources sont perçues comme « virtuelles » et donc « illimitées » : quand ça « rame », il suffit de remplacer son smartphone ou sa tablette par un modèle plus puissant et le tour est joué. De toute façon, le code sera obsolète à la prochaine génération d'équipements et il faudra donc le réécrire rapidement. Les conséquences ne sont pas visibles à l'échelle de temps du développement logiciel : pourquoi se préoccuper d'autre chose ?

Et pendant ce temps, le gaspillage fait des ravages. Dans ce monde où l'information nous submerge, où les sites web rivalisent de générosité dans une ambiance générale du « tout gratuit » et du toujours plus, il n'y a plus de temps et, « à force d'oublier l'essentiel pour l'urgence, de faire de l'urgence l'essentiel, on finit par oublier l'urgence de l'essentiel » (Edgar Morin).

L'essentiel, c'est quoi au fait ? Chacun aura sa réponse. Mais peut-être pouvons-nous nous entendre sur un dénominateur commun qui serait de transmettre à nos enfants une planète vivante.

La première bonne nouvelle, c'est que les compétences existent encore, dans quelques niches bien pointues : les mondes de l'« embarqué », du microcapteur, des codes de simulation ou de modélisation pour la recherche scientifique, etc., regorgent de développeurs hautement qualifiés pour satisfaire les contraintes croisées des exigences fonctionnelles et des limites technologiques. La deuxième bonne nouvelle, c'est qu'un effort de développement sur les milliards de milliards de lignes de code à venir aurait un réel effet sur les impacts des TIC en général, notamment via une réduction de l'obsolescence.

Alors, dans cet immense chantier qui est de remettre de l'essentiel au cœur de nos pratiques, cet ouvrage, proposé par l'un des spécialistes du green IT en France, apporte des réponses, des trucs et astuces pour commencer ou intensifier, chacun à sa mesure, ce travail de fourmi pour réapprendre à ne plus gaspiller le temps et l'espace de nos précieux objets technologiques. Il appartiendra alors à chacun de ne pas utiliser les espaces ainsi libérés pour y installer de nouvelles applications, sans quoi les gains escomptés pourraient bien être réduits à néant !

Françoise Berthoud
Directrice du groupe EcoInfo
www.ecoinfo.cnrs.fr