

Montre-moi tes commentaires et je te dirai si tu es un bon développeur

En programmation, les commentaires sont des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas nécessaires à l'exécution du programme, nous dit Wikipédia.



De fait ils sont le plus souvent utilisés pour expliquer le code, et contrairement à ce dernier qui nécessite un langage de programmation, les commentaires se font dans une langue humaine, généralement en anglais.

C'est une opération indispensable et une partie intégrante du travail du développeur, afin qu'un autre puisse comprendre et éventuellement modifier le code initial (quitte à ce que cet autre soit l'auteur même des commentaires, reprenant son code quelques temps plus tard).

De par le fait que sa nature en autorise l'accès et le ré-appropriation, on comprendra que ces commentaires soient encore plus importants dans le cas d'un logiciel libre.

Il se trouve cependant que, par paresse, manque de temps ou mépris pour ce qui ne serait pas de la programmation stricto sensu, ils sont parfois quelque peu négligés par les développeurs.

Or l'hypothèse défendue ici est qu'il y aurait une corrélation directe en la qualité des commentaires et la qualité du code.

Affirmer ainsi péremptoirement que l'un n'irait pas sans l'autre est somme toute assez osé. Mais cela a le mérite

d'évoquer cette « face cachée » de la programmation qui fait aussi partie des compétences attendues d'un bon développeur^[1].

Si les commentaires sont moches, le code est moche

If the comments are ugly, the code is ugly

*Esther Schindler – 15 novembre 2009 – IT World
(Traduction Framalang : Goofy)*

C'est un peu dur, je sais, mais finalement je crois que c'est vrai.

Le développeur de Plone Martin Aspeli m'a signalé un billet assez judicieux dans lequel un programmeur en C faisait part de trois règles apprises à ses dépens. Celle qui a particulièrement attiré l'attention de Martin (et la mienne) est la suivante :

Les bons programmes ne comportent aucune faute d'orthographe ni de faute de grammaire. Je pense que cela vient probablement d'une attention soutenue aux moindres détails ; dans les programmes excellents tout est correct à tous les niveaux, jusqu'aux points qui terminent les phrases de commentaires.

« Allez, tu charries ou quoi ? ». Vous pourriez croire que pinailler à ce point est indigne de vous, « Et bien je m'en vais commencer à vous signaler les erreurs dans votre code lui-même ». J'ai été bien embarrassé les deux ou trois fois où ça m'est arrivé.

La programmation, que vous la pratiquiez en enthousiaste de l'open source ou parce que vous êtes au service du patron, est un exercice qui exige de l'attention aux détails. Quiconque écrit un logiciel doit être pointilleux, sinon le code ne fonctionnera pas. Je suis sûre que je n'ai pas besoin de vous

citer la série des célèbres bogues de programmation dont s'est rendu coupable un programmeur qui a utilisé une virgule et non un point, ou des plantages aléatoires provoqués par deux lignes de code placées au mauvais endroit ?

Il est possible que mon jugement soit faussé par mes fonctions d'écrivain et d'éditrice, mais je crois fermement que si vous ne pouvez pas trouver le temps d'apprendre les règles de syntaxe de la langue (y compris la différence entre « c'est » et « ces », ou encore « m'a » et « ma »), je ne crois pas que vous puissiez être beaucoup plus consciencieux pour écrire du code en respectant les bonnes pratiques. Si vos commentaires sont négligés, je m'attends à trouver du code négligé.

Mon postulat vient de ce que m'a enseigné un éditeur très brillant, il y a dix ans (Laton McCartney, si tu lis ça : merci !). Si tu peux écrire le chapeau, disait-il, tu peux écrire l'article très facilement. Si tu ne peux pas écrire le chapeau, c'est que tu ne comprends pas encore ce que tu veux dire, et tu ne devrais pas commencer à écrire (le « chapeau » est le petit paragraphe d'accroche qui suit le grand titre, qui dit en un mot de quoi il est question et invite gentiment le lecteur à en savoir plus). De filandreuses « explications » du code dans les commentaires de l'application (c'est-à-dire, ceux qui se lisent comme de bonnes excuses) indiquent que le développeur ne savait probablement pas ce qu'il faisait au juste. Ce qui signifie que son code est éligible au titre de nid à bugs.

Se plaindre de la mauvaise documentation interne est un vieux refrain, mais il existe une raison pour laquelle il est important de la faire bien. Vos commentaires sont la seule façon dont vous disposez pour communiquer avec la prochaine personne qui regardera votre logiciel (il se peut même que ce soit vous) en profondeur, et pas juste une ligne ou deux. À quoi pensiez-vous en écrivant ce code ? D'accord, du code « auto-documenté » c'est l'idéal, mais c'est un peu arrogant de prétendre que vous l'avez atteint, et ça l'est tout autant

de ma part si je prétends que mes phrases n'ont pas besoin d'être révisées (elles en ont besoin, je suis bien contente d'avoir un correcteur).

Un autre problème habituel dans la médiocrité des commentaires apparaît lorsque les développeurs mettent à jour le code sans mettre à jour les commentaires ; comme l'a expliqué un consultant, les commentaires ne sont pas testés. Mais est-ce qu'il ne s'agit pas là encore d'un manque d'attention aux détails ? À chaque fois que vous n'êtes pas totalement attentif, vous êtes prédisposé à laisser tomber un petit bout de logique.

Vous voulez quelques exemples ?

En voici un tiré d'une explication à propos d'un choix de conception. (« oui, je devine que c'était probablement intentionnel », a écrit le programmeur qui m'a montré cet exemple. « mais c'est bien là le problème : il faut que je devine ».)

«... des questions prévues pour injustifier un débat...»

Voici maintenant une authentique ligne de code avec un commentaire. Notez que le commentaire met l'accent sur la faute d'orthographe. Sans compter que le commentaire vous indique que le programmeur n'avait pas la moindre idée de ce qu'il faisait au départ. Particulièrement parce qu'il n'aurait pas dû écrire « finished » (*NdT : fini*) dans ce bout de code, mais qu'il aurait dû essayer « complete » (*NdT : terminé*).

```
if item.getState() == 'finsihed': #est-ce correct?
```

Bon, je vous accorde quelques exceptions. Si l'anglais n'est pas votre langue maternelle, il est possible que les commentaires que vous laissez dans votre code montrent que vous n'êtes pas à l'aise pour écrire en anglais. Cependant, en ce qui me concerne je n'ai pas trouvé que c'était le cas. Comme beaucoup d'Américains s'en sont rendu compte, les

capacités en anglais des « étrangers » sont généralement bien meilleures que les nôtres. Parmi les développeurs que je connais, ceux qui ont des lacunes dans leur maîtrise de la langue anglaise en sont conscients et font tout leur possible pour qu'un anglophone passe en revue leur documentation.

Je ne me lancerai pas dans des considérations sur l'indentation du code et des commentaires, c'est une guerre de religions. Mais j'ai rencontré des développeurs qui réagissaient aussi violemment devant un formatage de code « moche » que moi devant une grammaire moche.

Les environnements de développement modernes procurent la discutable possibilité aux développeurs d'être négligents sans effets pervers ; une interface glisser-déposer vous permet de créer une application vite fait bien fait (ou plutôt mal fait, j'en ai peur) avec bien moins d'horribles dommages collatéraux qu'à l'époque où les logiciels étaient généralement écrits en assembleur. Pourtant, d'une certaine façon, j'ai du mal à considérer le « je peux être négligent » comme un avantage réel.

La règle du commentaire « moche » vaut autant pour les applications propriétaires que pour les logiciels open source. Mais les développeurs de FOSS (*NdT : Free and Open Source Software, les logiciels libres et open source*) doivent faire beaucoup plus attention encore pour deux raisons. La première, c'est qu'au bureau, vous avez de grandes chances de croiser le développeur du code mal fichu et de pouvoir lui poser des questions (ou de lui en coller une, si c'était vraiment horrible). De plus, dans la communauté open source, davantage de gens regardent dans le code et ont besoin de le comprendre.

Toutefois, lorsque j'ai suggéré à des développeurs que « si les commentaires sont moches, le code est moche », beaucoup n'étaient pas d'accord. Et vous non plus peut-être. J'aimerais savoir pourquoi, et je vous invite donc à laisser un... commentaire.

Notes

[1] Crédit photo : Fraserspeirs (Creative Commons By)