

Apprenez de vos utilisateurs (Libres conseils 21/42)

21/42 ! Tiens, nous voilà déjà à mi-chemin de la traduction d'*Open Advice*.

Deux ou trois articles petits ou grands chaque jeudi, traduits en un temps record par une bande de furieux, ceux qui sont là depuis le début et ceux qui débarquent et demandent s'ils peuvent participer, ceux qui travaillent d'arrache-clavier et ceux qui en profitent pour ~~déconner~~ échanger des propos sur le chat du pad, ceux qui choisissent un pseudo et ceux qui restent *anonymous*, ceux qui négligent tranquillement l'orthographe et les *grammar nazis* qui rectifient, ceux qui traduisent avec *Google translate* et ceux qui se battraient pour une majuscule à Libre... on rencontre tout un peuple là, et jusqu'à présent tout se passe dans l'enthousiasme et la bonne humeur, l'échange et l'entraide devant un passage un peu ardu sur lequel on s'amuse à chinoiser...

Mais cette fois-ci c'est l'auteur de l'article lui-même qui nous a fait le plaisir de nous rejoindre pour contribuer à la traduction, ce qui est tout de même assez confortable. Merci Guillaume !

Malgré nos relectures croisées, nul doute que des coquilles auront échappé à notre vigilance et que nous trouverons des lecteurs pour les signaler, ce qui nous est précieux car la révision avant l'édition du framabook en sera d'autant facilitée.

D'ici là, en avant pour la deuxième moitié : chaque jeudi à 21h, rendez-vous sur le framapad de traduction, le travail collaboratif sera ensuite publié ici même.

Traduction Framalang : [Nyx](#), [Sphinx](#), [peupleLà](#), [Kalupa](#), [Guillaume Paumier](#), [Husi10](#), [lenod](#), [Sky](#), [Julius22](#), [Alpha](#), [RavageJo](#), [KoS](#), [Sputnik](#), [goofy](#), [lamessen](#)

Apprenez de vos utilisateurs

Guillaume Paumier

Guillaume Paumier est photographe et physicien, il habite à Toulouse. Wikipédien

depuis longtemps, il travaille actuellement pour la Wikimedia Foundation, l'organisation à but non lucratif qui héberge Wikipédia. En tant que responsable de l'ergonomie multimédia, il a notamment étudié le comportement des utilisateurs afin de créer un nouveau système d'import de fichiers pour Wikimedia Commons, la médiathèque libre associée à Wikipédia.

Vous connaissez Wikipédia, l'encyclopédie libre et gratuite que tout le monde peut modifier ? Elle a été créée en 2001 et a récemment fêté son dixième anniversaire. Bien qu'elle soit l'un des dix sites les plus visités au monde, son interface reste très « 1.0 » quand on la compare aux possibilités qu'offrent les technologies web modernes. Certains peuvent trouver ça bien : Wikipédia est un « truc sérieux » et les utilisateurs n'ont pas à être distraits par des « paillettes » dans l'interface. Pourtant, si Wikipédia a eu du mal à recruter de nouveaux contributeurs ces dernières années, c'est en partie à cause de son interface que certains considèrent comme archaïque. Ceci explique peut-être pourquoi les enquêtes sur les participants à Wikipédia ont montré à maintes reprises qu'il s'agit principalement d'une population d'hommes jeunes, attirés par la technologie, la plupart ayant une formation en informatique et en ingénierie. En dehors du fait que la connaissance libre et les licences libres sont issues du terreau fertile du logiciel libre et open source, l'interface compliquée a découragé beaucoup de contributeurs éventuels.

En 2011, alors que la majorité des plates-formes de publication collaboratives en ligne (comme WordPress, Etherpad et Google Documents) offrent un éditeur graphique, même rudimentaire, Wikipédia utilise toujours, par défaut, un ancien éditeur de texte wiki qui utilise des guillemets (""") et des crochets ([[]]) pour la mise en forme. Des efforts sont en cours afin de passer à un éditeur graphique par défaut en 2012, mais ce n'est pas un défi facile à relever.

Mais laissons l'éditeur de côté un moment. L'interface de Wikipédia demeure assez compliquée. Et de nombreuses fonctionnalités utiles sont difficiles à découvrir. Savez-vous que Wikipédia possède un système de contrôle de versions intégré ? Et que vous pouvez voir toutes les anciennes versions d'une page ? Savez-vous que vous pouvez voir la liste de toutes les modifications effectuées par un contributeur ? Savez-vous que vous pouvez créer un lien vers une version donnée d'une page ? Savez-vous que vous pouvez exporter une page en PDF ? Savez-vous que vous pouvez créer un vrai livre personnalisé à partir du contenu de Wikipédia ? Et que vous pouvez le faire livrer chez vous ?

Le modèle d'implémentation

La plupart des lecteurs de Wikipédia y arrivent via des moteurs de recherche. Les statistiques montrent qu'ils passent peu de temps sur Wikipédia une fois qu'ils ont trouvé l'information qu'ils cherchaient. Un petit nombre seulement s'attarde et explore les outils que propose l'interface. Par exemple, on critique régulièrement Wikipédia sur sa qualité et sur sa fiabilité. Nombre de ces outils rarement explorés et presque cachés pourraient s'avérer bien utiles aux lecteurs pour les aider à vérifier la fiabilité de l'information, telles que les « pages de discussion » qui témoignent des discussions (passées et en cours) entre les différents contributeurs de chaque article ayant abouti à son contenu actuel.

Wikipédia et ses projets frères (tels que Wikisource et Wikimedia Commons) sont propulsés par le moteur de wiki MediaWiki — et sont soutenus par la *Wikimedia Foundation* ; rien que ces noms, dans leur confusion, sont un péché contre l'ergonomie. Pendant longtemps, le développement de MediaWiki a été conduit par des développeurs de logiciels. La communauté MediaWiki est forte de nombreux développeurs ; à vrai dire, la communauté est presque exclusivement composée de développeurs. Ce n'est que récemment que des designers ont rejoint la communauté, et ils ont été recrutés par la *Wikimedia Foundation* pour ce rôle. Il n'y a quasiment aucun designer bénévole dans la communauté. De ce fait, l'application a été construite et « maquettée » exclusivement par des développeurs. Par conséquent, la forme de l'interface a naturellement suivi de très près le « modèle d'implémentation », c'est-à-dire la manière dont le logiciel est implémenté dans le code et les structures de données. Le modèle d'implémentation ne correspond que rarement au « modèle utilisateur », c'est-à-dire à la manière dont l'utilisateur imagine que le logiciel fonctionne.

Il serait injuste de dire que les développeurs ne se soucient pas des utilisateurs. Quand on crée un logiciel, le but — outre le plaisir d'apprendre des choses, d'écrire du code et de résoudre des problèmes — c'est de le publier afin qu'il puisse être utilisé. Ceci est particulièrement vrai dans le monde du logiciel libre et open source, où la plupart des développeurs donnent bénévolement de leur temps et de leurs connaissances. On pourrait avancer que les développeurs sont, de fait, des utilisateurs de leurs propres produits, notamment dans le monde du logiciel libre et open source. Après tout, ils les ont créés ou ont rejoint leurs équipes pour une bonne raison, et c'est rarement l'argent. Par conséquent, les

développeurs de ce type de logiciels devraient être dans une position idéale pour savoir ce que veulent leurs utilisateurs.

Mais soyons honnêtes : si vous êtes en train de lire ceci, c'est que vous n'êtes pas votre utilisateur lambda.

Le point de vue du développeur

Si vous êtes développeur, il vous est particulièrement difficile de vous mettre à la place de l'utilisateur. Tout d'abord, votre connaissance du code et de l'implémentation du logiciel vous force à observer ses fonctionnalités et son interface à travers un prisme très particulier. Vous connaissez chacune des fonctionnalités de l'application que vous avez créée. Vous savez où trouver chaque menu. Si quelque chose paraît légèrement bizarre dans l'interface, il est possible que vous l'ignoriez sans le vouloir, parce que vous savez inconsciemment que c'est lié à la façon dont la fonctionnalité est implémentée.

Imaginons que vous soyez en train de créer une application qui enregistre des données sous forme de tableau (par exemple, dans une base de données). Quand vous devrez ensuite afficher ces données pour les montrer à l'utilisateur, il est très probable que vous les représentiez comme un tableau, car c'est la façon dont vous avez implémenté leur stockage. Il vous paraîtra logique d'afficher les données dans un format qui est cohérent avec le format de stockage. Vous aurez probablement le même réflexe pour tout autre type de structure de données séquentielles : vous aurez tendance à l'afficher sous forme de séquence dans l'interface, peut-être comme une liste. Et pourtant, un autre format d'affichage aurait peut-être été plus pertinent et facile d'utilisation pour les utilisateurs, par exemple sous forme d'une série de phrases, d'un graphique ou d'une autre représentation visuelle.

Un autre défi est votre niveau d'expertise. Comme vous souhaitez que votre application soit extraordinaire, vous allez probablement vous documenter sur le sujet pour la concevoir. En fin de compte, vous n'allez pas seulement connaître votre application sur le bout des doigts, vous allez également devenir un expert dans le domaine lui-même. Un grand nombre de vos utilisateurs n'auront pas ce niveau d'expertise — ou n'en auront pas besoin. Ils pourraient être perdus par le niveau de détail de certaines fonctionnalités ou ne pas être familiers avec des

termes inconnus des profanes.

Alors, que pouvez-vous faire pour arranger cela ?

Observez les utilisateurs. Vraiment

Observer les utilisateurs à l'œuvre avec votre application est une expérience réellement révélatrice.

Bon, pour observer comment les gens utilisent votre application, vous pouvez faire appel à une société de conseil en ergonomie ; cette société va alors recruter des volontaires avec des profils différents au sein d'un vivier de plusieurs milliers de testeurs, elle va mettre au point une grille d'entretien, louer une salle dédiée aux tests d'ergonomie qui comprendra un dispositif pour enregistrer ce qui se passe sur l'écran et une caméra pointée vers le testeur, et vous serez derrière une vitre sans tain, dans une salle d'observation, à vous taper la tête contre les murs et à jurer à chaque fois que l'utilisateur fait quelque chose qui, selon vous, n'a aucun sens. Si vous avez les moyens de le faire, alors n'hésitez pas, foncez. Ce que vous y apprendrez vous permettra de complètement changer votre point de vue. Si vous n'avez pas les moyens de recourir à une procédure de test professionnelle, tout n'est pas perdu ; vous allez juste devoir le faire par vous-même. Asseyez-vous derrière un utilisateur pendant qu'il vous montre comment il effectue ses tâches et les intègre à son mode de travail. Soyez un observateur silencieux : votre but est d'observer et de noter tout ce qui se passe. Beaucoup de choses vont vous surprendre. Une fois que l'utilisateur a terminé, vous pouvez relire vos notes et lui poser des questions afin de mieux comprendre comment il fonctionne.

Pour vous aider à conduire ces tests vous-même, il existe d'excellents ouvrages comme *Don't Make Me Think: A Common Sense Approach to Web Usability* [NdT: Traduit en français : *Je ne veux pas chercher: Optimisez la navigation de vos sites et menez vos internautes à bon port*], écrit par Steve Krug, *About Face 3: The Essentials of Interaction Design*, d'Alan Cooper, Robert Reimann et David Cronin, et le projet OpenUsability. Être observé peut être un peu intimidant pour les utilisateurs, mais je parie qu'ils seront nombreux à se porter volontaires pour vous aider à améliorer votre application. Les utilisateurs qui ne peuvent pas contribuer au code sont généralement heureux de trouver d'autres façons de

contribuer au logiciel libre : vous montrer comment ils utilisent le logiciel est une manière simple de le faire. Les utilisateurs sont reconnaissants du temps que vous avez donné pour développer le logiciel et veulent vous rendre la pareille.

Vous devrez garder un esprit critique et ne pas forcément accepter toutes les modifications demandées par vos utilisateurs. Écoutez attentivement leurs histoires : elles vous donneront l'occasion d'identifier des problèmes. Mais ce n'est pas parce qu'un utilisateur réclame une fonctionnalité qu'il en a absolument besoin ; peut-être que le meilleur moyen de résoudre le problème sous-jacent est de mettre en place une fonctionnalité complètement différente. Gardez du recul par rapport aux commentaires de vos utilisateurs. Mais cela, vous le saviez probablement déjà.

Et au passage, ne faites pas non plus appel à votre famille.

Je ne dis pas ça méchamment, je suis sûr que vos parents, frères et sœurs sont des gens très bien. Mais si vous créez une application comptable et que votre sœur n'a jamais tenu la moindre comptabilité, elle sera sans doute perdue. Vous perdrez plus de temps à lui expliquer ce qu'est la comptabilité en partie double qu'à tester votre logiciel. Par contre, votre mère, qui s'est achetée un appareil photo numérique l'année dernière, pourrait être un cobaye idéal si vous travaillez sur une application de gestion de photos numériques ou d'envoi sur un site de partage en ligne populaire. Pour votre application de comptabilité, il vaut mieux demander à l'un de vos collègues ou amis qui a déjà quelques notions de comptabilité.

Variez vos cobayes

Pour des raisons qui resteront éternellement mystérieuses, les gens trouveront toujours d'innombrables façons d'utiliser et de maltraiter votre application. Ils trouveront des manières de la casser que vous n'auriez même pas imaginées dans vos pires cauchemars. Certains mettront en place des processus et des méthodes de travail avec votre application qui n'ont absolument aucun sens à vos yeux. Et, de désespoir, vous vous cognerez la tête contre les murs. D'autres utiliseront votre application avec tellement d'intelligence que vous vous en sentirez idiot. Essayez de rencontrer des gens qui utilisent votre application avec des objectifs différents.

Les utilisateurs sont de drôles d'oiseaux. Mais ils sont de votre côté. Apprenez d'eux.

Si vous ne retenez rien d'autre...

... alors retenez ceci :

- Vous serez tenté de modeler l'apparence et le comportement de votre interface sur la façon dont le logiciel fonctionne en coulisses. Vos utilisateurs peuvent vous aider à éviter ce piège.
- Les utilisateurs sont des oiseaux capricieux. Ils vont casser, maltraiter et optimiser votre application à un point que vous ne pouvez pas même pas imaginer.
- Apprenez de vos utilisateurs. Améliorez votre application en fonction de ce que vous avez appris. Vous avez tout à y gagner.