

Honneur et déshonneur des lieux communs

par Ivan Maffezzini

Darwin [knows] full well that the most dependable data on behavior and breeding of domesticated and cultivated organisms would be obtained from active farmers and husbandmen, not from lords of their manors or authors of theoretical treatises.

(Steven J. Gould, *The Structure of Evolutionary Theory*)

Ceci est une sorte d'introduction à la chronique qui dans chaque numéro de la revue, abordera un lieu commun du génie logiciel. Pour expliquer objectifs et méthode, commençons par quelques questions et considérations générales sur les lieux communs.

Qu'est-ce qu'un lieu commun ?

Pourquoi toute personne un tant soit peu critique essaie-t-elle de les éviter comme un mal honteux ?

Couramment par « lieux communs » on entend les banalités, les idées rebattues, les clichés, les formules, etc. Une fois qu'on les définit de la sorte, ce n'est pas seulement tentant mais c'est presque un devoir que de les fustiger. Ce qui a comme conséquence fâcheuse que l'on se laisse trop facilement emporter pas la *vis polemica* et qu'on balaye du revers de la main toute nuance, tombant ainsi dans un simplisme comparable à celui des lieux communs que l'on conspue. On a beau être les défenseurs de la raison, de l'approfondissement critique et de la réflexion sans entraves dogmatiques, quand on s'insurge contre l'emploi de formulations acceptées de façon acritique par un grand nombre de gens on dilue, souvent sans s'en apercevoir, la raison dans l'acide du moralisme.

On essaiera d'éviter ces écueils, en s'appuyant sur la certitude que :

1. *Personne n'est à l'abri des lieux communs.* À un moment ou à un autre on en a tous besoin, ne fût-ce que pour faire l'économie de détours que l'on juge inutilement longs. Les lieux communs deviennent problématiques, graves, lorsqu'ils prennent une place hors du commun, ce qui arrive, malheureusement un peu trop souvent, dans notre profession. Certes on invoque pour cela des justifications solides : pas de temps pour les distinguos trop recherchés, pris comme nous le sommes par les contraintes de la production : de machines, ou d'articles, peu importe. Réutiliser des formules bien confectionnées et prêtes à porter fait gagner du temps. La réutilisation n'est-ce pas un des mots clefs de la discipline ?
2. *L'un des premiers devoirs* dans un travail technique, quand il ne s'agit pas d'une simple exécution matérielle ou d'une défense acritique d'opinions personnelles, est d'analyser les éléments que l'on a devant soi pour en étudier les origines, les fins, la composition et la structure, et cela vaut autant pour les éléments matériels que langagiers. Ce qui implique que l'on ne bombarde pas un lieu commun pour l'écraser : ce serait inutile et bête, car les lieux communs, par définition, sont partagés par beaucoup de gens et ils renaîtraient, sans doute sous une forme encore plus pernicieuse, quelque part ailleurs.
3. *En raison de son extrême jeunesse*, le génie logiciel a besoin d'échafauder des protections conceptuelles, pas nécessairement « scientifiques », pour avancer et répondre aux demandes toujours plus pressantes que son succès appelle. Cet échafaudage ne deviendra inutile que quand la structure conceptuelle se sera mieux stabilisée. Cette chronique, par exemple, se veut une minuscule contribution à la consolidation de la nouvelle structure qui est encore bien loin d'être tirée au cordeau.

L'approche, comme le souligne le titre, ne se limitera donc pas à considérer les « déshonneurs » des lieux communs, mais aussi à prendre en considération ce qu'il y a d'honorable en eux. Le fait d'avoir une histoire derrière eux implique qu'ils ont eu la force de surmonter un certain nombre d'obstacles et l'intelligence d'éviter certains pièges. Dans cette partie « honorable », on peut voir un clin d'œil à la signification première de « lieux communs » qui n'avait pas la connotation péjorative moderne.

Rappelons que les « lieux communs » naissent sous la plume d'Aristote et que pour le Stagirite, ils étaient une source où les humains pouvaient puiser les éléments communs nécessaires à toute démarche discursive car seulement quand

« Nous aurons dressé l'inventaire des opinions qui sont celles de la moyenne des gens, nous nous adresserons à eux, non point à partir de propositions qui leur seraient étrangères, mais à partir de celles qui leur sont propres, quand nous voudrions les persuader de renoncer à des affirmations qui nous paraîtront manifestement inacceptables. »

Même si cette acception de l'expression « lieux communs » est actuellement cantonnée à certaines spécialisations de la philosophie ou des sciences politiques, elle nous influence aussi en raison de la démarche de SWEBOK¹ qui se situe à l'intérieur de ce cadre aristotélicien où les « gens » deviennent les « experts ». Ce qui transforme des lieux communs généraux en lieux communs particuliers à une discipline.

Mais les lieux communs d'Aristote ne se sont pas transformés en clichés de but en blanc, ils sont passés par au moins un changement significatif aux alentours de la Renaissance où les lieux communs deviennent les sentences célèbres d'écrivains anciens qui font autorité. Et qu'il est bien vu d'apprendre par cœur. Et, du « par cœur » au « cliché », le chemin est assez direct. Rien de nouveau sous le soleil donc quand on voit que, dans notre domaine, les sentences de certains livres « classiques » sont employées comme clefs pour ouvrir toutes les portes. Rien de nouveau, mais ce n'est pas pour cela qu'il ne vaut pas la peine de bien analyser ces clefs, surtout en ce moment où, après s'être débarrassés de la formule à succès « crise du logiciel² », on essaye d'en solidifier les assises.

Après cet excursus historique, retournons à nos moutons. Malheureusement pour ceux qui espèrent atteindre un idéal quelconque d'objectivité, les lieux communs des autres sont toujours plus affadés et inutiles que les nôtres. Mais les nôtres, comme ceux des autres, dépendent de notre formation, notre travail, nos paradigmes cachés... Voilà pourquoi il nous semble utile, par respect des lecteurs, de dire que les lieux communs que l'on étudiera seront ceux qui parfois nous ont aidé et parfois entravés en trente ans d'activités d'« *active farmers* » dans la ferme informatique métissées de vingt ans d'une approche plus théorique de professeur à l'université. Il s'agira donc du mélange de l'approche d'un praticien qui essaie de faire évoluer les produits qui rechignent toujours un peu plus que prévu avec l'approche pratico-théorique d'une responsabilité de formation de jeunes ayant comme défi pédagogique majeur de ralentir leur course sans pour autant ternir leur l'enthousiasme.

On est donc prêt à s'installer dans le premier lieu commun de la série et à présenter un prototype avec l'espoir que ce ne sera pas un « prototype jetable » — pour employer le jargon habituel. Il s'agit d'un lieu commun qui, depuis les années 1960, a eu des hauts et des bas avec un apogée vers la fin des années 1990. Il concerne les méthodes de développement et on pourrait le formuler comme suit : *l'utilisateur a un rôle fondamental à jouer dans la réalisation d'un système.*

NOTE : les lieux communs que nous analyserons ne seront jamais des citations car, par définition les lieux communs, même s'ils ont été créés par X, constituent désormais une « richesse » du patrimoine de la discipline. FIN DE LA NOTE

¹ IEEE, SWEBOK, 2004 version, <http://www.swebok.org/>

² Qui fera sans doute partie des lieux communs que nous aborderons.

Pour finir nous aimerions souligner que, dans cette chronique, nous n'aurons jamais la prétention d'avoir les bonnes réponses, mais que nous espérons réussir à poser au moins une bonne question.

Centralité des utilisateurs

Ce lieu commun est sans doute un bon exemple de départ parce qu'il représente assez bien le côté cliché et formule magique de la chose. Il suffit pour cela de considérer que ce lieu commun est devenu un réflexe conditionné de tous les étudiants en génie logiciel³ qui répètent immédiatement la formule dès qu'on leur demande ce qui est important pour obtenir un bon produit ; que les organisations, qui ont une approche un tant soit peu systémique et systématique de leurs applications, en ont fait le centre de leur travail ; que la majorité des livres de génie logiciel et les articles qui traitent de méthodes, de processus et de qualité le présentent comme une vérité aperte. Mais en même temps, en raison de cet accord presque universel, ce lieu commun est un excellent point de départ pour échanger sur ce qu'est un utilisateur ou un système ; sur le partage des responsabilités entre les machines et les humains ; sur la rigidité des interfaces ; sur la fiabilité etc.

Certes ce lieu commun a aussi un grave défaut comme point de départ : il est trop général et trop lié à la vision que chacun a de la place de l'humain dans le monde des machines.

Honneur

Impossible de nier que si l'utilisateur final n'est pas d'accord avec les fonctionnalités ou avec la façon de les mettre en œuvre, il y a des risques très grands que le succès du produit soit, dans le meilleur des cas, mitigé. Si l'on veut donc courir le moins de risques possibles de refus, il est complètement inconsidéré d'attendre les réponses de l'utilisateur lors de la livraison finale. Pour maîtriser la réalisation du produit, il est donc nécessaire d'entendre les avis des utilisateurs dès la phase de la préparation du document *Principes d'opérations*⁴. Leur point de vue ne devra pas être un point de vue parmi d'autres, mais le point de vue principal si l'on ne veut pas se faire piloter par les exigences bureaucratiques ou techniques.

Considérer que souvent les utilisateurs finaux, à cause de leur formation, sont incapables ou ne veulent pas participer aux premières phases est irrecevable, car il est souvent facile de trouver un « champion » et, quand cela n'est pas possible, on peut faire une analyse de la tâche pour mieux comprendre le rapport entre les personnes et les machines. Qui n'a pas vu les dégâts causés par les développeurs qui assument le rôle que devait jouer l'utilisateur en prétextant les difficultés conceptuelles de ce dernier à comprendre le système ? Les difficultés existent en effet, mais dans l'autre sens : ce sont les développeurs qui, incapables d'abandonner leur terminologie pour entrer celle du domaine, livrent souvent un produit qui ne fait pas ce qu'il était censé faire.

Déshonneur

Même si nous croyons qu'un utilisateur n'interagit jamais avec un logiciel mais toujours avec du matériel, nous n'emploierons pas cette arme létale car la « tradition » veut que cela soit possible.

³ Les premières pages des mémoires de maîtrise et des thèses de doctorat qui sont souvent un concentré de lieux communs

⁴ En ce qui concerne la terminologie, à moins d'indication explicite contraire, nous ferons référence aux normes IEEE.

NOTE. Il s'agit ici d'une exemplification de ce que nous avons affirmé dans l'introduction : le fait que le lieu commun est partagé, même si on le juge inacceptable, est suffisant pour ne pas l'écraser par en respecter l'histoire. FIN DE LA NOTE.

Nous ne considérons non plus l'objection voulant qu'il y ait un grand nombre de produits qui n'interagissent pas avec des humains (un protocole de communication par exemple) car, dans l'état actuel de l'automatisation, même des produits qui ne s'interfaçent qu'avec des machines dans leur fonctionnement normal, devraient avoir une interface pour l'entretien et le personnel responsable de l'entretien qui, du point de vue logique, est l'utilisateur « final ».

Voici quelques points faibles que nous ne voulons pas laisser de côté :

1. Considérer *le plan sur lequel la machine et l'humain interagissent* et oublier qu'il y a (au moins en théorie) une finalité qui les transcende. Certes, cette finalité, quand elle n'est pas un simple moyen économique⁵, n'est pas forcément écrite dans des documents et encore moins inscrite dans une « volonté divine » mais elle est le résultat d'un compromis *in fieri* entre les exigences techniques, organisationnelles, économiques et sociales : les volets classiques du document *Principes d'opération* (ou de *L'étude de faisabilité*). Souvent, l'utilisateur est la personne la moins apte à avoir du recul, surtout si le nouveau produit prend la place d'un produit qu'il connaît bien et avec lequel il a travaillé longtemps. *L'analyse de la tâche*, par exemple, peut avoir des effets pervers sur les produit, car l'analyse du travail de l'utilisateur facilite la perte de vue de la finalité qui, tout bien considéré, pourrait faire... disparaître la tâche, que nous avons analysée avec tant d'acharnement.
2. *Quel utilisateur ?* Si l'on oublie le cas, peu probable même s'il est très alléchant, d'un produit pour un seul utilisateur immensément riche, en manies et en argent, on doit organiser les utilisateurs en classes d'utilisateurs. Souvent plus d'une classe avec chacune de nombreux utilisateurs. Mais, dès que l'on passe d'une instance d'utilisateurs (disons, Carole Bouquet) à des classes (actrices françaises ou actrices jeunes ou actrices de théâtre, etc.) on est aux prises avec la subjectivité des classifications et donc avec les besoins de l'« utilisateur » influencés par le mécanisme de classification — ce qui, tout en voulant mettre l'utilisateur au centre, y met en fait le « classificateur ». À cela il faut ajouter que, dans des classes nombreuses, il faut choisir le « champion » qui représente la classe ce qui, encore une fois, ne dépend pas des utilisateurs⁶. Donc tout ce que la centralité de l'utilisateur fait sortir par la porte principale (le carcan organisationnel, économique et technique) revient par la porte de service.
3. *Corruption du champion*. Ceux qui ont réalisé des projets avec des « champions » des utilisateurs connaissent les dangers énormes que cela implique. Très souvent, le « champion » s'intègre tellement bien dans l'équipe qu'il perd, peu à peu, toutes ses caractéristiques d'utilisateur « normal » pour devenir un « champion » au sens du « meilleur » et donc n'être plus d'aucune utilité pour les buts recherchés. Parfois, il se crée un antagonisme entre l'équipe de développement et le « champion » qui du coup se transforme en champion pour mettre des bâtons dans les roues. Certes il y a des solutions, mais il est souvent difficile sinon impossible de savoir de combien le « champion » s'éloigne de l'utilisateur normal qu'il devrait représenter. Inutile de dire que quand le champion revient dans sa fonction d'utilisateur, il est souvent perçu comme le « champion » qui représente le projet, ce qui mêle encore plus les cartes.

⁵ Ce qui, selon nous, nie son caractère de finalité.

⁶ Proposer un choix démocratique est souvent impossible et ce n'est pas du tout sûr que cela donnerait de bons résultats.

Ces considérations nous semblent suffisantes pour être moins assertif sur la centralité de l'utilisateur.

Au-delà

Aller au-delà de ce lieu commun en mettant quelque chose d'autre au centre, par exemple *le* système, l'organisation, les fonctionnalités ou la qualité serait sans doute un remède pire que le mal, au moins pour certains types de produits.

Aller au-delà pour pouvoir rendre notre discipline plus solide implique prendre du recul pour chercher des points d'appui qui ne soient pas de simples points de vue tactiques. Par exemple :

- Pourquoi oublions-nous si souvent que l'adaptabilité est l'une des caractéristiques les plus proprement humaines ? Ce qui ne veut pas dire que l'on doit réaliser un produit n'importe comment, mais qu'il faudrait mieux exploiter ce degré d'élasticité des utilisateurs.
- Pourquoi ne pas garder toujours à l'esprit que le but principal de l'automatisation est de libérer l'humain du travail ? Ce qui implique que si l'on analyse les tâches des utilisateurs, il faut le faire dans ce but et non dans celui de trouver la meilleure façon de faire dans le contexte actuel.
- Ne devons-nous pas, nous, ingénieurs du logiciel, proposer des logiciels toujours plus solides et flexibles sur lesquels des gens d'autres disciplines viennent « accrocher » les interfaces avec les humains ? Ce qui implique, par exemple, de développer des interpréteurs pour des langages sémantiquement proches du domaine et des tâches des utilisateurs et arrêter de vouloir trouver des langages universels.
- Ne devons-nous pas arrêter de penser qu'il existe des formules valables pour tous les domaines ? Comment ne pas garder à l'esprit que le nombre des domaines automatisables est pratiquement infini et qu'il serait naïf de penser qu'il existe des méthodes universelles !
- Ne faut-il pas privilégier l'introduction de l'automatisation là où, à cause des machines déjà installées, l'appauvrissement sémantique du domaine est déjà très grand et libérer ainsi les humains du travail répétitif ?
- Les interfaces personne-machine ne devraient-elles pas devenir toujours moins importantes pour laisser plus de place aux interfaces machine-machine bien plus à la portée de tout génie ?

Et, pour finir, nous aimerions souligner que le génie logiciel est la discipline qui concerne avant tout le « matériel » — un matériel spécial mais matériel tout de même ! — qu'est le logiciel (listages, diagrammes, textes, etc.). Remettre l'accent sur cette matérialité impliquerait se rendre compte que l'on devrait développer surtout les applications pour les ingénieurs du logiciel en mettant l'utilisateur au centre, pour qu'il puisse manipuler le matériel logiciel plus facilement.