

L'ordinateur et le facteur humain

Hubert Ricard

Volume 49, Number 3, 1981

Introduction à l'informatique

URI: <https://id.erudit.org/iderudit/1104136ar>

DOI: <https://doi.org/10.7202/1104136ar>

[See table of contents](#)

Publisher(s)

HEC Montréal

ISSN

0004-6027 (print)

2817-3465 (digital)

[Explore this journal](#)

Cite this document

Ricard, H. (1981). L'ordinateur et le facteur humain. *Assurances*, 49(3), 225–229.
<https://doi.org/10.7202/1104136ar>

Article abstract

Maximum results can only be obtained when the user works closely with the computer. In his article, Mr. Ricard stresses the importance of the human factor, as individuals do, in practice, demand that electronic processing equipment function at optimum capacity.

L'ordinateur et le facteur humain

par

HUBERT RICARD
de la maison Mathéma Inc.⁽¹⁾

Maximum results can only be obtained when the user works closely with the computer. In his article, Mr. Ricard stresses the importance of the human factor, as individuals do, in practice, demand that electronic processing equipment function at optimum capacity.

225



Un ordinateur champion d'échecs! Un problème d'ordinateur retarde le départ de la navette spatiale! L'ordinateur s'est encore trompé! Voilà des phrases que vous avez déjà eu l'occasion de lire ou d'entendre et qui vous auront sans doute laissé un peu perplexe. En effet comment se peut-il qu'un ordinateur qui, dans certaines circonstances, peut accomplir des prodiges, à d'autres moments déraisonnera complètement. S'agit-il d'un dédoublement de personnalité, l'ordinateur un jour génie devient-il subitement crétin le lendemain ou bien existe-t-il une réponse simple?

Erreurs d'ordinateurs

Un ordinateur est une machine qui exécute, très rapidement et sur des grandes quantités de données, mais automatiquement, les programmes qui lui ont été confiés. La seule différence entre votre calculatrice de table et un ordinateur en est une d'échelle. Les deux machines ne peuvent pas exécuter une commande qui n'a pas été donnée par un être humain et si vous essayez d'exécuter une opération qui n'a pas de sens, telle une division par zéro,

(1) La maison Mathéma Inc. fait partie du groupe Sodarcam.

vous obtiendrez un rejet de cette opération ou un résultat imprévisible.

On s'aperçoit donc que la source la plus importante d'erreur n'est pas la machine elle-même, mais plutôt les programmes qui y sont exécutés.

226 Les erreurs causées par un mauvais fonctionnement sont assez rares dans les ordinateurs modernes. En effet, les constructeurs ont ajouté à la plupart des ordinateurs des circuits électroniques, qui vérifient de temps à autre l'état des pièces d'équipement et qui peuvent même détecter l'affaiblissement de l'une d'entre elles et l'indiquer au technicien chargé de l'entretien qui la remplacera avant qu'elle ne puisse réellement faire défaut.

Cela ne laisse donc que deux sources d'erreur: le programmeur et l'utilisateur. Nous allons voir dans les prochains paragraphes que les responsabilités pour les erreurs sont assez bien partagées entre les deux groupes.

Les programmeurs

Sous ce vocable, nous regroupons tous les informaticiens qui entrent en contact avec les utilisateurs. Ce sont eux qui analysent les besoins de l'utilisateur en lui faisant décrire le cheminement des opérations; ils conçoivent le système, le proposent à l'utilisateur, le programment, le documentent, et enseignent à l'utilisateur le mode d'opération. Ils en supervisent aussi la mise en opération.

Pour le programmeur, le système idéal est celui où l'utilisateur est capable de décrire dans ses moindres détails le fonctionnement du système à développer et où les spécifications originales ne sont pas modifiées durant la période nécessaire au développement. La raison de cet immobilisme et de ce grand besoin de précision est que les programmeurs doivent définir chaque opération à exécuter, chaque validation dans ses moindres détails et les convertir dans un langage compréhensible par l'ordinateur. Par lui-même, l'ordinateur ne peut porter aucun jugement sur la sémantique d'une information, du moment que celle-ci est valide syntaxiquement, elle sera acceptée; par exemple, l'ordinateur

n'acceptera pas un nombre qui contient deux points décimaux ou dont le signe n'est pas au début ou à la fin; toutefois le programmeur devra faire les tests appropriés si un nombre doit se trouver dans un certain intervalle ou avoir une ou plusieurs valeurs précises (tel un nombre d'heures travaillées dans un système de paye qui doit être positif, être moins de 100 heures et un bon programme de validation donnerait en plus un avertissement si ce nombre était inférieur à 20 ou supérieur à 50). Une partie du travail de programmation est donc extrêmement fastidieux et les programmeurs n'aiment en général pas avoir à les recommencer. Un autre problème causé par les modifications est que certaines choses sont beaucoup plus faciles à réaliser si elles sont prévues au niveau de la conception du système; dans les pires cas il devient impossible d'effectuer certaines modifications sans abandonner une majeure partie du travail déjà fait.

227

Les programmeurs ne sont pas, eux non plus, infaillibles et peuvent commettre des erreurs. Celles-ci peuvent provenir d'une mauvaise compréhension des besoins de l'utilisateur, soit que celui-ci les ait mal exprimés ou que le programmeur les ait mal compris, elles peuvent venir d'une erreur d'analyse ou de programmation ou plus simplement d'une situation qui n'a pas été prévue ni par l'utilisateur ni par le programmeur.

Ces erreurs peuvent se produire souvent après une assez longue période de temps, être difficiles à corriger et causer des problèmes graves (pensez au problème causé par un nombre d'heures négatif qu'un programmeur aurait oublié de rejeter; panique au département du personnel quand l'employé arrive avec son chèque négatif et son représentant syndical).

Même les périodes intensives de test ne permettent pas de détecter toutes ces erreurs possibles. C'est une réalité de la vie informatique qu'elles peuvent surgir à n'importe quel moment.

Les utilisateurs

Ce groupe comprend toutes les personnes qui vont entrer ou utiliser l'information traitée par l'ordinateur. Ce sont elles qui dé-

finissent les besoins, qui les communiquent aux programmeurs et qui utilisent le système lorsqu'il est fonctionnel.

Pour l'utilisateur, la formule idéale est souple et peut être modifiée rapidement, car les structures de l'entreprise et les conditions du marché sont dynamiques et changent continuellement.

228 Il est très difficile pour un utilisateur de définir en détail toutes les opérations du système, tous les rapports et toutes les informations requises à tous les instants. En effet, comment prévoir l'impact d'un nouveau rapport sur les besoins en information d'une entreprise? Peut-être ce rapport dévoilera-t-il des situations qui ont besoin d'être analysées et corrigées, qui pourront générer des modifications aux opérations de la compagnie.

De plus, l'utilisateur connaît peu ou pas du tout l'informatique. Il lui est souvent impossible de traduire ses besoins dans un langage compréhensible au programmeur (songez que chaque type d'entreprise possède un jargon technique pour décrire ses propres opérations), ce qui pose des problèmes certains lors de la définition du dialogue.

C'est au niveau des utilisateurs que l'on rencontre souvent des objections face à l'informatisation de certaines opérations (et peut-on toujours les blamer quand ce sont eux qui ont à faire face à tous les problèmes causés par un mauvais fonctionnement du système), voire même parfois de l'hostilité. Ces réactions négatives ont deux sources: l'incompréhension quant aux objectifs et au mode d'opération ou incapacité d'un individu à s'adapter au nouveau mode de procéder. Le premier problème peut être assez facilement résolu en informant d'avantage les individus; par exemple les administrateurs d'une compagnie peuvent vendre le nouveau système informatique en montrant les avantages pour la compagnie et pour les utilisateurs, de même les programmeurs doivent expliquer le fonctionnement du système dans un langage compréhensible aux utilisateurs.

La deuxième catégorie de problèmes, heureusement rare, représente des cas insolubles. L'individu doit alors être recyclé dans d'autres fonctions où il aura très peu de contact avec l'ordinateur.

GIGO:

Cet acronyme anglo-saxon (employé pour *garbage in, garbage out*: déchets à l'entrée, déchets à la sortie) décrit le problème fondamental de tout système informatique: l'ordinateur ne produira des résultats sensés que si l'information et les programmes qui lui ont été fournis par des êtres humains sont exacts.

229

Pour arriver à développer un système informatique qui fonctionne avec un minimum de problèmes dans des délais raisonnables, il est primordial que les programmeurs et les utilisateurs travaillent en équipe et que chaque partie accepte de mettre un peu d'eau dans son vin; par exemple, les programmeurs accepteront certaines modifications nécessaires, mais en contrepartie les utilisateurs remettent à plus tard d'autres modifications utiles, mais non essentielles. La coopération entre les deux parties assurera un développement rapide et avec un minimum de problèmes. Elle permettra surtout de développer un instrument de travail qui sera utile et qui sera bien accepté par les utilisateurs.

Taxes à la consommation au Canada: Guide destiné aux hommes d'affaires, par la maison Clarkson, Gordon. Montréal.

La maison Clarkson, Gordon continue la publication de ces textes très simples destinés aux hommes d'affaires. Dans le cas présent, il s'agit des taxes à la consommation que l'auteur passe en revue. Le texte est récent puisqu'il remonte à juin 1981. Même si certaines modifications ont pu y être apportées, il sera utile à celui qui doit se dépêtrer au milieu de la jungle fiscale.